A Deterministic Dynamic Associative Memory (DDAM) Model for Concept Space Representation

by

Stefan Valerian Pantazi
M.D., "Carol Davila" University of Medicine and Pharmacy, 1998

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the School of Health Information Science

A Deterministic Dynamic Associative Memory (DDAM) Model for Concept Space Representation

by

Stefan Valerian Pantazi
M.D., "Carol Davila" University of Medicine and Pharmacy, 1998

Supervisory committee

Dr. Jochen R. Moehr, (School of Health Information Science)
Supervisor

Dr. Francis Y. Lau, (School of Health Information Science)
Departmental Member

Dr. John H. Esling, (Department of Linguistics)
Outside Member

Dr. Kin F. Li, (Department of Electrical & Computer Engineering)
Outside Member

Supervisory committee

Dr. Jochen R. Moehr, (School of Health Information Science)
Supervisor

Dr. Francis Y. Lau, (School of Health Information Science)
Departmental Member

Dr. John H. Esling, (Department of Linguistics)
Outside Member

Dr. Kin F. Li, (Department of Electrical & Computer Engineering)
Outside Member

ABSTRACT

This dissertation aims at the general goal of solving the problem of representing and processing information on conceptual principles, in an unsupervised, human-like manner, and using existing computational methods. Given this very general context, the need for intelligent applications that meet the complexity and sensitivity requirements of Medical Informatics is postulated in what is referred to as "the axiom of medical information systems." The reformulation of the axiom that "medical information systems must be, at the same time, usable and useful" leads naturally to the identification of more immediate, achievable objectives in the form of context dependent information processing and case-based reasoning research on memory models capable of unsupervised representation and processing of information, in a similarity-based manner. Further, the unification of these objectives is proposed in the form of the general problem of managing associative concept representation spaces characterized by four fundamental properties: high dimensionality, sparseness, dynamicity and similarity based organization. The thesis of this dissertation is that the solution to this problem can be approached in the most appropriate way by memory models that specifically address each and every one of the four fundamental properties. The support for the thesis is twofold and comprises theoretical accounts which lead naturally to the definition of a memory model, the

deterministic dynamic associative memory model (DDAM) which is based on the existing mathematical structure of partial order set. The model is first introduced informally by means of examples and depictions that speak for its usability. Further the formal description of the DDAM model and learning algorithms is achieved using existing fundamental concepts of set theory and combinatorics. Finally, the DDAM model is evaluated and compared with existing approaches in a series of experiments and simulations that demonstrate usefulness comparable or superior to existing approaches.

Table of Contents

List of Tables

List of Figures

Acknowledgements

Though this section is not supposed to be about me but about the people whom I want to thank, I will use some background information in order to provide the complete picture of the people whom I am grateful to and for what. The common theme that characterizes my life is "to understand how things work" which led me naturally to trying to "alter" those things. As a child, this exercise ended up most of the times in destruction and accidents. My parents, though occasionally angry, never stopped fulfilling my requests. As a result, now things end up most of the time with improvements.

Most of my education was oriented towards becoming a medical doctor; which I did in 1998. But getting there was more of a winding road rather than a straight path. As a child I was deeply passionate with biology, zoology, chemistry and physics and all these fit well with the goal of becoming a medical doctor. However, when I was about 16 years old, something unexpected happened: in communist Romania, Z-80 based personal computers (Sinclair Spectrum compatible) have started to come within the reach of "lay people" through small time entrepreneurs who would build and sell them in their spare time. A family acquaintance, who used to come and visit us often, was the first to introduce me to these "personal computers" that could "write text and draw lines and circles on a TV screen!" Since then, I could not sleep thinking of what I could do with one of those. Finally, my parents caved in and agreed to pay their six months worth of salary for a home-built Z-80 compatible computer which we bought from someone in Bucharest with the help of a relative. In was in the winter of 1989 that my life begun to change. From then on, things I used to be deeply passionate about have been replaced by my new toy, the computer.

After pursuing for a couple of years the possibility to become a computer engineer, through the intervention of another family friend, I was convinced to switch back to

medicine. After the required examination I got accepted into the medical school, and therefore the next 6 to 10 years of my life were spoken for. Failing my very first human anatomy exam and, at the same time, failing the very first exam in my then 18 year old life (that was the one and only exam I ever failed) made me first realize "**the big problem**": the extremely unreasonable amount of knowledge medical doctors are required to acquire during their training.

During my medical school years I have constantly been using various kinds of computers, whenever I had the opportunity. A good friend had lent me one of his old AMIGA computers which I actually managed to break soon after I started to use it. I want to thank him again for forgiving me.

The interest of involving computers in my student work stemmed also from the need to have my course notes in electronic form. At the same time, all this work made me wonder how could I make use of computers in order to get rid of **the big problem** I was facing not only as a student but as a future practitioner as well. The natural course of things has suggested itself to me: the exploration of computational models that are able to process medical knowledge.

In 1996, I showed some of my artificial associative memory prototypes and since then I continued to collaborate with Prof. Tiberiu Spircu. He played an important role in my career that led to my becoming, in 1999, a junior member of the Medical Informatics department of the same medical school I was just graduating from in 1998. In 2001, immediately after getting married to my wife, Felicia, I arrived in Canada in pursuit of my PhD degree under the supervision of Prof. Jochen Moehr, who agreed to have me working on the topic I was most interested in: **the big problem**.

My life in Canada has been blessed with amazing support of my only family member, my wife Felicia, to whom I am grateful for everything she has done for me. I have also been fortunate to have had the continuous moral and financial support of my mentor, supervisor and friend, Jochen who never stopped to believe in the work I was doing, despite all the snags along the way. I also am indebted to Bogdan Verjinschi, a good

friend with whom I shared many good times, crazy ideas and interesting discussions (and many beer and wine bottles as well).

I am also grateful to the many individuals who directly and indirectly contributed to my work through discussions, feedback on research papers, meetings, collaborations, conferences, presentations as well as to all outstanding researchers whose wonderful articles, textbooks, recordings, presentations and software have found their way to my computer and filled the conceptual space of this dissertation.

Through life-changing interventions of key people such as parents, friends, colleagues, professors, supervisors, business partners, researchers, authors, it appears that my career path was indeed more of a winding road. However, through the help of the very same people, I was lucky that the trip retained a certain consistency: I never had to quit *understanding how things work* or to give up trying to solve **the big problem.** And this is probably the reason why I feel now that I am getting closer, if not to the solution, at least to a better understanding of its nature.

Stefan V. Pantazi

Sunday, February 5, 2006

Dedication

This piece of my mind is dedicated to the three most important women in my life:

my wife, my mother and my sister.

Chapter 1 INTRODUCTION

This chapter contains the identification and definition of core issues, followed by the proposal for a solution (the thesis). The purpose of this chapter is to place the reader into the appropriate state of knowledge for introducing the thesis and for delving into the more advanced arguments and theoretical background in the next chapter.

1.1 RATIONALE AND OVERVIEW

This section provides a short rationale for pursuing work on fundamental aspects of medical Informatics and a concise overview that identifies goals and objectives as well as the chosen approach for the work. It begins with a very relevant quote from Knuth's Art of Computer Programming, vol. 3.

"It is interesting to note that human brain is much better at secondary key retrieval than computers are; in fact, people find it rather easy to recognize faces or melodies from only fragmentary information, while computers have barely been able to do this at all. Therefore it is not unlikely that a completely new approach to machine design will someday be discovered that solves the problem of secondary key retrieval once and for all, making this entire section obsolete."

Donald Knuth,

in "Retrieval on Secondary Keys" in "Sorting and Searching"

Art of Computer Programming, vol. 3

Most of one's education involves a tedious accumulation of domain knowledge. Domain knowledge often comes in various ways and from disparate sources such as textbooks, discussions, electronic media and one's own experiences. If domain knowledge formed a solid, clear, coherent, circumscribed, static assembly of fundamental theories, the research would consist mostly of the mere application and small refinements of existing theories. Fortunately this is not the case. Our ever-changing, uncertain reality makes it certain that scientists have jobs and that research is going to be fun and unpredictable. At the same time, the knowledge available for one's education, accumulated in too great amounts, often precludes one from learning more fundamental concepts. "Not being able to see the forest because of the trees" is a relevant saying that comes to mind when thinking that learning can actually prevent discovery. Again, fortunately, there is hope. A reasonable amount of time (probably more than a PhD degree) and a unifying driving force to fuel the attempt to make sense of seemingly disparate theories available in large numbers of knowledge artefacts, may give one a chance to put some order into and expose some of the fundamentals. This dissertation is my humble attempt towards this end. It is the result of a ten years long research program and, perhaps more importantly, of my incessant joy of building software and hardware artefacts. These gave me the

possibility to explore both theoretical concepts and practical applications through prototyping. My attempt to identify fundamental aspects and provide coherence to my own research aligns well with the general work on the theoretical foundation of Medical Informatics, a relatively new discipline, still in its infancy in my opinion.

Specifically, this dissertation aims at the general goal of solving the problem of representing and processing information in an unsupervised, human-like fashion, on conceptual principles, using existing computational methods. This work would be logically followed by the applications to Medical Informatics. Given the extreme complexity and sensitivity of Medical Informatics applications, the relevance of this general goal to our field of research and the need for intelligent applications are demonstrated by the existence of what is referred to in this dissertation as the *axiom of medical information systems*.

To advance towards the general goal, some more immediate, achievable objectives are further identified in the form of *context-dependent information processing* and *Case Based Reasoning research* on memory models which are capable of representing and processing information in a similarity-based manner which is as unsupervised and as human-like as possible. Further, this has led to the unification of the objectives into the problem of managing *associative concept representation spaces* characterized by four fundamental properties: high dimensionality, sparseness, dynamicity and similarity based organization. The hypothesis that the solution to this problem can be approached in the most appropriate way by memory models that specifically address each and every one of the four properties is the thesis of this dissertation.

The approach to support the thesis is twofold and comprises theoretical accounts that span disciplines and fields of research such as Medical Informatics, Artificial Intelligence and Computer Science. The theoretical discussions lead naturally to the definition of a memory model, the dynamic deterministic associative memory model (DDAM), which is introduced first informally, by means of examples and depictions. This gradual introduction is subsequently followed by complete, formal descriptions that make use of existing fundamental concepts of set theory. Finally, the DDAM memory model is

evaluated in a series of experiments and simulations that prove its usefulness when compared to other existing approaches.

1.2 CONCEPTUAL ROAD MAP

This section provides the structure and a compilation of all summaries of chapters and first level headings.

The dissertation is structured in five chapters.

Chapter 1. Introduction

This chapter contains the identification and definition of core issues, followed by the proposal for a solution (the thesis). The purpose of this chapter is to place the reader into the appropriate state of knowledge for introducing the thesis and for delving into the more advanced arguments and theoretical background in the next chapter.

1.1 Rationale

This section provides a short rationale for pursuing work on fundamental aspects of medical Informatics and a concise overview that identifies goals and objectives as well as the chosen approach for the work. It begins with a very relevant quote from Knuth's Art of Computer Programming, vol. 3.

1.2 Conceptual road map (you are reading it now)

This section provides the structure and a compilation of all summaries of chapters and first level headings.

1.3 The axiom of medical information systems

In this section, the "axiom of information systems" is introduced and allows the reader to become acquainted with the fundamental problem of representation. The reformulation of this axiom leads naturally to what is proposed to be a solution to this problem: creating information systems which have the capacity to acquire, with as high a degree of autonomy as possible, useful, relatively complete, problem specific representation of complex problems that need to be solved.

1.4 What does "to represent" mean, really?

In this section, a definition of the concept of "representation" is attempted by systematically exploring its semantic space. The conclusion is that fundamentally, the process of representation is a function that maps the reality that includes the object to be represented on a representation medium. Representation functions can be arbitrarily complex but most importantly they can be classified into two extreme types: the simple, non-evolving, context-independent, non-adaptive representation functions and the complex, evolving, context-dependent, adaptive representation functions.

1.5 A meta-theory of Medical Informatics

In this section, the discussion necessarily shifts to a meta level but retains the perspective of the important concepts introduced in the previous sections (i.e., representation and context). At the meta-level, science appears twofold because it comprises the creation of theories which are compressions of one's observations as well as application of theories in understanding, predicting and solving problems. Further, the notion of "knowledge

spectrum" is introduced and defined together with four interconnected modalities for representing human knowledge.

1.6 An associative memory model for concept space representation

In this section, the entire chapter is summarized and the need for natural language processing (NLP) and information retrieval research are underlined. Some important aspects such as dynamicity (frame problem), multidimensionality (case descriptions) and the similarity-based or associative organization implied by the case based reasoning paradigm are introduced. It is further suggested that the representation problem can be recast and unified around the notion of memory models capable of representing "associative concept spaces" characterized by four specific properties: multidimensionality, sparseness, dynamicity and associative (similarity-based) organization. Finally, the thesis is proposed that in order to achieve the advanced information processing required by Health Informatics applications, one has to devise approaches that efficiently address all four specific properties of associative concept spaces.

1.7 Goals, scope, method and limitations

In this section it is suggested that Medical Informatics is a young field of research which calls for work that reveals fundamental issues. A definition of the field is proposed from this perspective. The research methodology employed in this dissertation is strongly impacted by the complexity of the research and relies heavily on the processes of modeling and simulation that can be easily placed in the meta-level framework proposed earlier as well as on a literature review approach based on full text search approaches and which has allowed a broad literature review difficult to achieve by traditional methods.

Chapter 2. Theoretical Background

This chapter contains arguments and discussions of theoretical underpinnings of issues introduced previously and of the proposed solution in the light of existing literature, approaches and solutions for the management of associative concept spaces. It also contains a broad literature review of seemingly disparate theories, models and approaches that nonetheless converge and could be unified under the general umbrella of "approaches to associative concept space representation," as well as an in-depth discussion of design principles of a memory model that is able to address each one of the four fundamental properties of associative concept spaces (high dimensionality, sparseness, dynamicity, similarity based organization).

2.1 Case-based Medical Informatics

This section opens the chapter with a case based reasoning perspective on medical informatics decision making in medicine and knowledge representation and processing. The focus is on fundamental aspects of decision-making, which connect human expertise with individual context knowledge processing. Further, a knowledge spectrum perspective on biomedical knowledge is used to demonstrate that case-based reasoning is the paradigm that can advance towards personalized healthcare and that can enable the education of patients and providers.

2.2 Knowledge representation and processing

This section is a case based reasoning perspective on knowledge representation and processing. The completeness of formal languages and their connection to the frame problem is examined in detail. Further the need for approaches to deal with ubiquitous but ambiguous natural language descriptions is advocated.

2.3 Context-dependent information processing

In this section, a series of arguments regarding how information might be represented in the human brain are presented in order to show the disconnection with the representation of information in computers. The fact that the pattern space in which human brain operates is high dimensional but immensely sparse is underlined: patterns that make sense to us are memorized and they typically represent an extremely small fraction of all possible patterns. Conceptual representations that occur in the human brain are also highly complex, dynamic and context sensitive and they are examined from the perspective of elementary concepts of algorithmic information theory using a classic example of a coin tossing experiment. The discussion leads to an extension of the minimum description length (MDL) principle and to an algorithmic definition of a pattern that sacrifices description length in order to attain a reduction in the complexity of retrieval time.

2.4 Properties of associative concept spaces

In this section, the four properties of concept spaces (high dimensionality, sparseness, dynamicity and similarity based organization) are discussed. High dimensionality and sparseness are closely interrelated and can be illustrated by a "proof by resource exhaustion" argument that sets a common sense upper bound on a class of objects that are represented in a high dimensional space. Dynamicity and similarity-based organization are discussed in the context of pattern recognition, of retrieval based on secondary keys and of the dynamic classification capacity of humans.

2.5 Memory-based processing (i.e., trading space for time)

In this section the importance of memory in information processing is underlined from the perspective of language processing. More precisely, it is proposed that, in order to attain advanced information processing capabilities, the trade-off between space complexity (i.e., memory) and time complexity (i.e., speed) must favor the latter at the expense of the former.

2.6 Principles for associative concept space representation

The principles introduced in this section parallel the four properties of the concept spaces. High dimensionality is approached by hierarchical and distributed models. Sparseness and dynamicity lead to the use of hash functions, pointers, and linked lists while avoiding the use of arrays. Organization by similarity is approached through the use of trie memory models while avoiding the use of hash functions. The synthesis of all design principles leads naturally to the Deterministic Dynamic Associative Memory (DDAM) model proposed in this dissertation.

2.7 Unsupervised concept space representation tasks

This section of the theoretical background discussion is a functional perspective that reviews unsupervised functions, tasks and processes to represent associative concept spaces.

2.8 Concept space representation models and approaches

In this section the theoretical background discussion continues with a structural perspective that reviews unsupervised models and approaches which share similarities with the DDAM model in representing associative concept spaces.

Chapter 3. The Deterministic Dynamic Associative Memory (DDAM)

In this chapter the proposed associative memory model and processing algorithms are presented formally and their functionality demonstrated by means of examples.

3.1 Fundamental definitions

This section introduces fundamental mathematical concepts that describe the DDAM model, in form of definitions and pointers to additional material available in Appendix 1.

3.2 The unconstrained substring poset

In this section the building block of the DDAM model is formally defined and followed by additional important definitions (e.g., string compositions, composition ambiguity, etc.). These definitions form the core of the theoretical model.

3.3 Constrained substring posets

In this section two variants of constrained substring posets are formally defined and their functionality demonstrated. This provides the transition from purely theoretical models such as the constrained substring poset towards more practical ones.

3.4 Implementation considerations

In this section the discussion of implementation issues begins from the perspective of the four properties of concept representation spaces. The discussion continues with an estimation of the algorithmic complexity of the DDAM model and ends with a description of the visualization techniques used to create many of the graphical representations in the dissertation.

Chapter 4. Concept Space Representation Experiments

This chapter contains the experimental results and evaluation of the DDAM model through information processing experiments and comparison with the results of existing models on similar tasks. The experiments range from artificial to natural sequence processing and consist of various pattern discovery, grammar induction and natural language processing tasks.

4.1 Introduction

This introductory section is a short description of the methodology and of the global focus of the evaluation approach. The models employed in the first three experiments are the two variants of DDAM model (the simple constrained substring poset – DDAM-1 and the constrained substring model – DDAM-2).

4.2 Experiments with artificial sequences

This section comprises experiments conducted to explore the pattern discovery performances of DDAM models on artificial sequences, with a focus on the acquisition of significant regularities with long descriptions. The artificial sequences used in experiments are built by combining lexical items from artificial lexicons, in a manner similar to the descriptions available in (Elman 1990).

4.3 Experiments with natural sequences

This section comprises experiments that use the exact same processing principles, but on sequences that are natural (e.g., DNA sequences, text, etc.) rather than artificially constructed sequences as in previous experiments.

Chapter 5. Conclusions, Outlook, and Future Work

This chapter contains the conclusions, the contribution to knowledge, a to-do list of future work and a discussion of the possibilities for application of the proposed associative memory model to richer representations of information such as images, sounds and simulations.

5.1 Summary and conclusions

This section contains a short summary the work and conclusions.

5.2 Contribution to knowledge

This section summarizes the contribution to knowledge is which is threefold: methodological, theoretical (unifying traditionally distinct theories and concepts) and practical (the proposal, implementation and evaluation of the deterministic model of dynamic associative memory). It also comprises a list of publications that contain some of the material included in this dissertation.

5.3 Outlook and future work

This section contains a list of to do items as well as some ideas which are highly speculative and far reaching.

1.3 THE AXIOM OF MEDICAL INFORMATION SYSTEMS

In this section, the "axiom of information systems" is introduced and allows the reader to become acquainted with the fundamental problem of representation. The reformulation of this axiom leads naturally to what is proposed to be a solution to this problem: creating information systems which have the capacity to acquire, with as high a degree of autonomy as possible, useful, relatively complete, problem specific representation of complex problems that need to be solved.

There is no question that the concept of information is fundamental to our field of research. But, getting to the nitty-gritty of it, one would have to answer the simple, but fundamental question: "what exactly is information?" Lacking a satisfactory answer, one could turn to defining information through the things that one can do to it, that is to be created, stored, retrieved, communicated, modified and deleted. A logical next step would be to wonder about the nature of information processors involved in information processing (creation, storage, retrieval, communication, modification and deletion). At this point one can easily figure that information processing is split between human processing and information systems. This naturally brings us to the examination of the nature of this "split."

1.3.1 The split of information processing

Information systems are commonly thought of as being composed of two important parts: the *user interface* and the *problem-solving* engine. From a usability engineering perspective, if one connects the concept of *usability* with the user interface of information systems and the concept of *usefulness* with the complexity of problem-solving, the following "usability axiom" must hold: information systems must be, at the same time, *usable* and *useful*. However, because *usable* user interfaces need to be *simple* and because *useful* information systems able to solve complex problems require *complex* problem-solving engines, the usability axiom is also a paradox: information systems must be, at the same time, both *simple* and *complex*. The paradox vanishes only if we could divide

information systems completely and address separately their user interfaces and their problem-solving engines. Only in this case would we be able to build *simple*, low complexity user interfaces to systems that solve *complex* problems. Unfortunately, given existing information technology, the concepts of usability and usefulness appear dependent on each other, as if their sum must remain constant. Highly *usable* systems are often less *useful* because they typically solve trivial problems (e.g., generic, repetitive tasks). Conversely, potentially very *useful* systems that could solve complex, specific problems (e.g., planning a trip, or devising a therapy plan for a certain patient) usually end up exhibiting *usability* problems. This is so because, given a limited problem-solving engine, the complexity of the task to be solved spills over into the complexity of the user interfaces which subsequently may become an error causing factor (Koppel, Metlay et al. 2005).

1.3.2 Artificial intelligence

Before attempting to analyze the paradox, one also needs a proper characterization of the prototypical problem that medical information technology is supposed to solve. One possibility is to regard knowledge intensive decision-making – which includes prediction of outcomes of complex realities – as the prototypical informatics problem. At the extreme lies the prediction of one bit of information, such as, for example, to give/not to give a certain drug to a certain patient at a certain point in time given a particular, usually extremely complex patient description. If one disregards the medical connotation, this problem also falls under the realm of artificial intelligence (AI). As with any AI decision problem, in order to be solved, it must be represented in the memory of a problem-solver. In the context of an information system, this implies that a description of the problem must be acquired, typically, through a user interface. But the spatio-temporal reality of a patient is complex, multidimensional and dynamic and so must be the approaches to represent it without loss of information. Because it is based on abstractions, the process of acquisition of dynamic, complex, complete representations through human-computer interfaces is extremely difficult because of at least two interconnected problems: the knowledge acquisition bottleneck (Feigenbaum 2003) and the frame problem (Dennett

1984; Pylyshyn 1987; Pantazi, Arocha et al. 2004). Relevant to any expert system design, the former problem is more related to the amount of knowledge required to overcome brittleness (i.e., the susceptibility to fail outside a narrow domain), while the latter puts the emphasis on the need for representational approaches that can cope with dynamic, rapidly changing environments. In Medical Informatics, these problems are justifications for "the enormous upkeep effort" required to manually maintain dynamic, complex medical knowledge bases that are not only vast but may rapidly become obsolete (Moehr 1994).

1.3.3 The issue of representation

The focus on the representation issue reveals the strong coupling between the user interfaces and problem-solving engines of information systems and allows us to restate the usability axiom of medical information technology from a more fundamental perspective: information systems must be able to create *useful, relatively complete* (i.e., complete with respect to a certain purpose) internal representations of complex spatio-temporal realities (e.g., description of a patient), with minimal information loss, from simple, incomplete, abstract descriptions acquired through *usable, simple* human-computer interfaces. The self contradiction (i.e., the paradox) arises from the assumption that information systems are somehow able to fill in, automatically, the information gap between *usable, simple*, abstract descriptions and the *useful, relatively complete* representations required to solve complex problems. Historically speaking, this paradox has existed from the beginnings of information science and artificial intelligence. One usual reflection of this paradox was in the form of the dichotomy between symbolic processing and machine learning paradigms. Symbolic processing is the tenet of hand coded knowledge bases and rule driven expert systems. It is no surprise that these lack usefulness and suffer from the knowledge acquisition bottleneck (Feigenbaum 2003) and frame problem (Dennett 1984; Pylyshyn 1987; Pantazi, Arocha et al. 2004). Given existing technology, this suggests that machines that learn are the only alternative with the potential to address the paradox by attaining automatically, relatively complete, useful representations while allowing for simple, usable user interfaces of information systems.

The limitations of this alternative seem to lie only in the memory and processing power required for advancing from the stage of proof-of-concept "toy applications" to that of real world problem solvers.

1.3.4 The applied perspective

In Figure 1 a hypothesized relationship between usability (i.e. simplicity and ease of use of a system or user interface) and complexity/difficulty of problems solved by an information system, is illustrated.

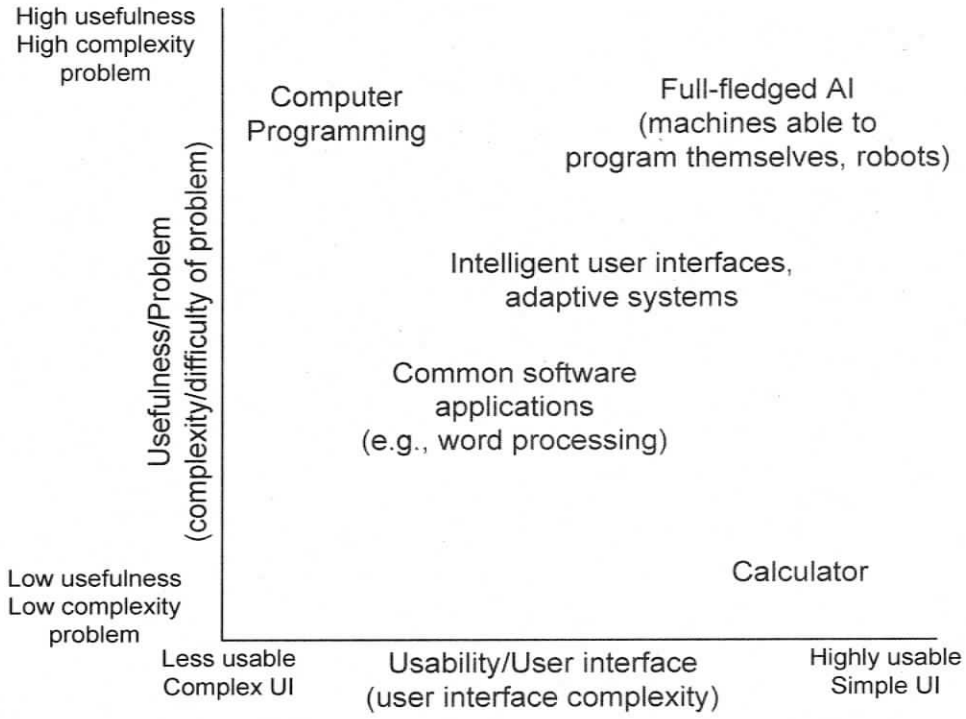


Figure 1. The relationship between usability and problem complexity and the role of artificial intelligence. Approaches that are highly usable (e.g., calculator) tend to solve less complex problems while approaches to solve highly complex problems are often considered "less usable" (e.g., computer programming)

Ideally, in healthcare we wish to "have our cake and eat it too" (Solomon, Roberts et al. 2000) - i.e., have systems that perform highly useful and complex tasks, and yet are easy and simple to use. But the expectations of systems being both highly *usable* as well as being capable of performing complex and *useful* tasks are often not met in practice. It is also unlikely that this apparent contradiction is more striking in other domains than in

health care, particularly in the area of applied artificial intelligence in healthcare. For example, early attempts at creating knowledge-based expert systems spawned development of a wide range of applications and systems, including MYCIN, INTERNIST and many other such systems (Szolovits 1982). These applications were expected to help physicians solve highly complex diagnostic tasks and yet do so in a manner that was easy to use, highly useful and that fit within the complex work practices of healthcare (Grant, Kushniruk et al. 2004). However, as well-stated by Miller and Masarie in their insightful and landmark article "The Demise of the 'Greek Oracle' Model for Medical Diagnostic Systems" it became clear that such systems that were designed to be highly useful by solving complex medical problems, would not be accepted by health care practitioners until they were also designed to actually be highly usable and simple in their operation (Miller and Masarie 1990). The field of applied artificial intelligence in medicine has been marked by issues, controversy and problems arising from the tension between usefulness and complexity on the one hand and usability and simplicity on the other. In particular, newer approaches to developing systems based on intelligent agents, adaptable user interfaces and intelligent tutoring systems (Brusilovsky and Maybury 2002), are all examples of AI systems developed to solve complex problems, while at the same time attempting to appear simple and usable to end users. This has become an ongoing, fundamental challenge both theoretically and pragmatically.

1.3.5 Conclusions

In this chapter it is going to be argued that the technical solution to addressing the paradox is centered on the thorny problem of representation. So far it was implied that complicating the user side of the equation with less *usable*, sophisticated representations is impractical, because knowledge acquisition cannot be achieved with handcrafted symbolic representations alone. Given the particular features of the prototypical informatics problem, the exact nature of the solution seems to consist of endowing information systems with the capacity to acquire, with as high a degree of autonomy as possible, *useful, relatively complete, problem-specific* representations of the complex

problems that need to be solved. To a certain extent, systems that "can learn" could be thought of being able "to program themselves." If they are considered "intelligent," solving the paradox equates to building intelligent systems. This dissertation attempts a step forward toward this end from both theoretical and practical points of view.

1.4 WHAT DOES "TO REPRESENT" MEAN, REALLY?

In this section, a definition of the concept of "representation" is attempted by systematically exploring its semantic space. The conclusion is that fundamentally, the process of representation is a function that maps the reality that includes the object to be represented on a representation medium. Representation functions can be arbitrarily complex but most importantly they can be classified into two extreme types: the simple, non-evolving, context-independent, non-adaptive representation functions and the complex, evolving, context-dependent, adaptive representation functions.

In the previous section we introduced the concept of representation but we did not define it. We will therefore continue with a conceptual exploration of the semantic space of the concept of representation. Hopefully this will allow us to understand it at a more fundamental level and lead to a more formal definition. We use one of the best-known available sources of general semantic knowledge, WordNet (Miller 1995), and explore the semantic neighbourhood of the concept in the syntactic roles of noun (i.e., "representation") and verb (i.e., "to represent").

1.4.1 The semantic space of the concept of "representation"

All twenty-five distinct senses (Table 1, Table 2) attest to the semantic richness of the concept and suggest a potentially high degree of ambiguity. However, despite various usage contexts some of which are only remotely connected or clearly outside the discourse of this dissertation (e.g., the senses related to law practice or theatrical performance), all different senses of the concept retain a certain abstract, prototypical meaning.

	Lemma(s)	Definitions and examples
1	representation, mental representation, internal representation	a presentation to the mind in the form of an idea or image
2	representation	a creation that is a visual or tangible rendering of someone or something
3	representation	the act of representing; standing in for someone or some group and speaking with authority in their behalf
4	representation, delegacy, agency	the state of serving as an official and authorized delegate or agent
5	representation	a body of legislators that serve on behalf of some constituency; "a Congressional vacancy occurred in the representation from California"
6	representation	a factual statement made by one party in order to induce another party to enter into a contract; "the sales contract contains several representations by the vendor"
7	theatrical performance, theatrical, representation, histrionics	a performance of play
8	representation	a statement of facts and reasons made in appealing or protesting; "certain representations were made concerning police brutality"
9	representation	the right of being represented by delegates who have a voice in some legislative body
10	representation	an activity that stands as an equivalent of something or results in an equivalent

Table 1. Selected senses out of the distinct ten senses of the noun "representation" in WordNet, in the decreasing order of their estimated usage

Senses of high interest, which appear directly connected to the issues addressed in this dissertation, are 1, 2 and 10 for the noun "representation" (Table 1) and 1, 2, 3, 5, 6 and 15 for the verb "to represent" (Table 2).

Sense #	Lemma(s)	Definitions and examples
1	represent, stand for, correspond	take the place of or be parallel or equivalent to; "Because of the sound changes in the course of history, an 'h' in Greek stands for an 's' in Latin"
2	typify, symbolize, symbolise, stand for, represent	express indirectly by an image, form, or model; be a symbol; "What does the Statue of Liberty symbolize?"
3	represent	be representative or typical for; "This period is represented by Beethoven"
4	represent	be a delegate or spokesperson for; represent somebody's interest or be a proxy or substitute for, as of politicians and office holders representing their constituents, or of a tenant representing other tenants in a housing dispute; "I represent the silent majority"; "This actor is a spokesperson for the National Rifle Association"
5	represent	serve as a means of expressing something; "The flower represents a young girl"
6	exemplify, represent	be characteristic of; "This compositional style is exemplified by this fugue"
7	constitute, represent, make up, comprise, be	form or compose; "This money is my only income"; "The stone wall was the backdrop for the performance"; "These constitute my entire belonging"; "The children made up the chorus"; "This sum represents my entire income for a year"; "These few men comprise his entire army"
8	defend, represent	be the defense counsel for someone in a trial; "Ms. Smith will represent the defendant"
9	represent, interpret	create an image or likeness of; "The painter represented his wife as a young girl"
10	act, play, represent	play a role or part; "Gielgud played Hamlet"; "She wants to act Lady Macbeth, but she is too young for the role"; "She played the servant to her husband's master"
11	stage, present, represent	perform (a play), especially on a stage; "we are going to stage 'Othello'"
12	represent	describe or present, usually with respect to a particular quality; "He represented this book as an example of the Russian 19th century novel"
13	represent	point out or draw attention to in protest or remonstrance; "our parents represented to us the need for more caution"
14	present, represent, lay out	bring forward and present to the mind; "We presented the arguments to him"; "We cannot represent this knowledge to our formal reason"
15	map, represent	to establish a mapping (of mathematical elements or sets)

Table 2. Selected senses out of the distinct fifteen senses of the verb "to represent" in WordNet, in the decreasing order of their estimated usage

Additional semantic information is gathered from related concepts that fall in general under the patterns "X is a representation" and "to X is to represent." The two non-exhaustive categories of lexical items are labelled "REPRESENTATION" and "REPRESENT" and listed in Table 3.

Category	Lexical items
REPRESENTATION (nouns) X is a representation	representation, image, form, model, simulation, map, stereotype, schema, perception, memory, example, adumbration, copy, replica, reproduction, carbon copy, anamorphism, impression, cutaway, display, document, drawing, ecce homo, effigy, illustration, nomogram, objectification, picture, icon, projection, rubbing, cast, duplicate, triplicate, facsimile, imitation, counterfeit, forgery, knockoff, clone, miniature, toy, photocopy, print, Xerox, transcript
REPRESENT (verbs) to X is to represent	represent, stands for, correspond, take the place of or be parallel or equivalent to, typify, symbolize, express indirectly, be representative or typical for, serve as a means of expressing, exemplify, be characteristic of, map, imitate, simulate, recreate, replicate, reproduce, duplicate, double, clone, repeat, photocopy

Table 3. Two groups of lexical items belonging to the semantic space of the concept of representation in the syntactic role of noun (i.e., representation) and verb (i.e., to represent)

1.4.2 Representation functions

From the semantic information gathered about the concept of representation one can infer that, fundamentally, the process of representation can be defined (i.e., represented) by a function (in the mathematical sense) that maps the reality of an object to be represented onto an arbitrary representation medium. Let us call this information processing function a *representation function*. Therefore, given a certain spatio-temporal reality R (e.g., a moving object), the concept of representation is a function F of R, onto another spatio-temporal region I, which is the representation, image, model or projection, that represents, stands, is projection of, or corresponds to R. Put symbolically:

$$F(R) = I$$

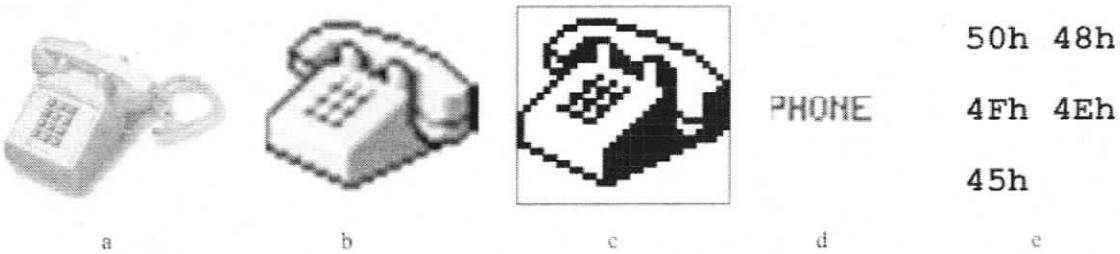


Figure 2. Possible representations of a real, actual telephone set. a, b, c are bitmaps of images, d is a bitmap of written symbols and e is the sequence of ASCII hexadecimal codes corresponding to characters "P", "H", "O", "N" and "E".

For example, if R were a real, actual telephone set, then, given a certain function F, $I = F(R)$ could conceivably be any of the possible representations in Figure 2.

Essentially, F can be thought of as an algorithm which takes inputs features from R (e.g., colors, shapes, sounds, spatio-temporal patterns) and maps them onto features (e.g., pixels of an image, words in a text, phonetic symbols of a phonetic transcription) of a medium which could be paper, clay, a blackboard, magnetic media and computer memory, in order to create an image I that is a projection of R. If F is implemented into a computing device (e.g., a digital camera, a computer), then the inputs are binary sequences generated by sensors such as imaging devices or microphones, while representation media are typically computer memory, magnetic media or paper. If F is a human information processing function, then the representation medium is our memory. One property of F is important and follows immediately. Applying F to a reality R results invariably in loss of information: for any F, there are always features in R (e.g., features at molecular or atomic levels, features in obscured sides of an object, features in a distant history of an event or that are considered non-relevant for a specific purpose, etc.) that do not make it into the representation I of R and which are therefore lost. This property is related to the limits of human perception of recording and measurement hardware and of the representation media. It could also be related to a particular, relevance-driven implementation of a representation function F which strips out from a representation I the features considered irrelevant. This is what Blois refers to as "necessarily incomplete descriptions of natural objects" or "abstractions" (Blois 1984)[1]. However, unlike the case of abstractions, applying a certain F (e.g., a lossless image compression algorithm) to an existing digital object (e.g., a digital image) could result in a complete, non-abstracted representation (e.g., a compressed data file) that loses no information.

Yet, for this discussion, the most important characteristic of F is with regard to its dependence on context.

---

[1] Page 32

1.4.3 Dependence of context

In Medical Informatics the importance of context and context-sensitive processing has already been recognized (Blois 1984; Moehr 1994). Blois shows why context is needed in order to make representations understandable by providing counter-examples of "totally out of context" messages and points out that "ordinary utterances are always made in a situation, and will thus have a context" (Blois 1984)[2]. He acknowledges the role of context in the resolution of ambiguity (Blois 1984)[3] in human communications but also underlines the implicit nature (as opposed to explicit) of context that causes its "critical role" to be underestimated (Blois 1984)[4]. In a similar vein, Moehr shows how context-deprived representations (e.g., "peptic ulcer") do not stand necessarily for an uniform reality, that their processing must depend upon the context where they were gathered and that the difficulty of processing of health data is caused precisely by its high context sensitivity (Moehr 1994)[5]. At the same time, the context dependence of medical data seems to contradict the existing biomedical research paradigms such as populational studies (e.g., randomized control trials) which purposely aim at removing the context around medical data in order to attain population-wide applicability of their findings (Pantazi, Arocha et al. 2004).

This very important context perspective allows us to distinguish two extreme types of F:

1) The simple, non-evolving, context-independent, non-adaptive representation functions,

2) The complex, evolving, context-dependent, adaptive representation functions.

---

[2] Page 22-23
[3] Page 36
[4] Page 33
[5] Page 251

1.4.3.1 Simple representation functions

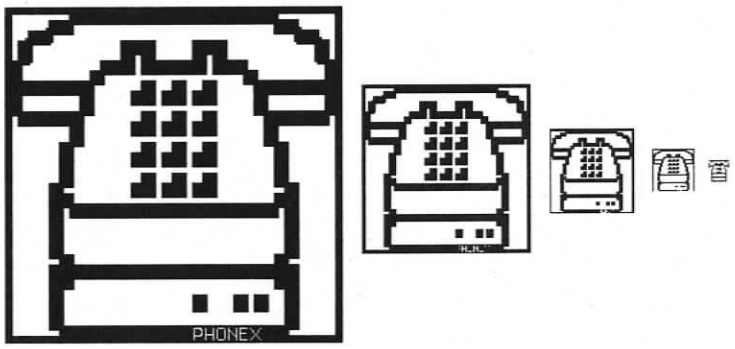


Figure 3. Example output of a simple representation function

For example, a representation function in the first category is a "shrink" algorithm that removes every second pixel from images such as those in Figure 3. Under the same category falls a zoom-out function on a world-map that generates increasingly larger scale representations. Another example could be a simple algorithm that transforms a grey image into its binary black and white representation (e.g., Figure 2 b and c). In general, representations that are affine transformations of an input (i.e., which preserve colinearity and ratios of distances (Weisstein 2005)) could be thought of as belonging in this fist category. Such functions can be applied multiple times, in various orders, even recursively, and because they are context-independent (e.g., an image to be processed could contain anything), their implementation is trivial and holds for any input.

1.4.3.2 Complex representation functions

An example for a representation function in the second category is one that accepts as input a reality R or existing representations such as images in Figure 2 (a, b, c) and outputs abstract, conceptual, symbolic representations such as Figure 2 (d, e). Such representation functions are highly context-dependent, and are often implemented as evolving, adaptive algorithms. These algorithms fall under the realms of artificial intelligence and machine learning, which are known to be challenging tasks for existing information technology.

1.4.4 Conclusions

To conclude, representation functions can be arbitrarily complex but are virtually non-evolving, context-independent functions in the case of existing information technology: the output of a simple representation function is little or not at all altered by the history of its outputs. On the other hand, context-dependent representation functions such as those that occur in the brain are dynamic (changing with time, adaptive) and highly context-dependent: new information is processed in the context of an individual's experience relevant to that new information. Such representations could be considered a higher order of processing better described as knowledge processing rather than just information processing. This leads to defining knowledge as context-dependent information and knowledge processing as context-dependent information processing.

1.5 A META-THEORY OF MEDICAL INFORMATICS

In this section, the discussion necessarily shifts to a meta level but retains the perspective of the important concepts introduced in the previous sections (i.e., representation and context). At the meta-level, science appears twofold because it comprises the creation of theories which are compressions of one's observations as well as application of theories in understanding, predicting and solving problems. Further, the notion of "knowledge spectrum" is introduced and defined together with four interconnected modalities for representing human knowledge.

When one steps back in order to take a theoretical stance, to further understanding and to propose solutions to fundamental issues in a field of research, one often attempts to propose unified views which become meta-theories of sorts. Applied to a certain area of research, a unified view is a meta-theory which, if valid, would have to account for anything carried on in that field of research. In this dissertation, the accepted definition of the concept of theory is that of "a tentative hypothesis about the natural world; a concept that is not yet verified but that if true would explain certain facts or phenomena," a definition which corresponds to the second meaning of the word "theory," available in WordNet (Miller 1995). The need for subscribing to this particular definition stems from the complexity and open endedness of biomedical research as well as from the inevitable inductive nature of theory elicitation processes resulting in hypotheses and models whose validity can never be proved to be universal truth.

1.5.1 A meta-level view of Science

Traditionally the concern of philosophers, meta-level discussions of fundamental principles will also place one's field of research in the context of other sciences. From such a general, meta-level perspective, science could be defined as "the business of eliciting theories from observations in a certain context, with the hope that those theories will help to understand, predict and solve problems." Also revolving around the "business of creating theories," R. Solomonoff's ideas (Solomonoff 1964), summarized in (Chaitin 1970), propose that a scientist's theories are compressed representations of her

observations (i.e., her experimental data). These compressed representations are used to explain, communicate and manage observations efficiently and, if valid, to help solving problems, to help understanding of issues and perhaps most importantly, to help predicting the future to the extent possible. Intuitively, the higher the compression achieved by the theory, the more "elegant" that theory and the higher its chances of acceptance. A very general meta-level perspective of the scientific endeavour makes it appear that science comprises the creation of theories (i.e., theory elicitation) as well as their subsequent use in understanding, predicting and solving problems (i.e., theory application). Therefore, science seems to be driven by two opposite forces: that of creating theories (i.e., compressing observations into theories), and that of applying those theories to practical applications (i.e., instantiating theories into action).

1.5.2 The context of scientific observations

The four-dimensional space-time continuum we live in (i.e., our universe) forms the reality (i.e., the context) of all scientific observations. The compression of the immense complexity and dynamicity of this reality in concise "theories of everything" was already demonstrated by Zuse (Zuse 1969) and recently Schmidhuber (Schmidhuber 1997). These results of theoretical computer science demonstrate the power of human theory elicitation and provide important answers to old questions of science and philosophy. However, their unfeasibility when applied to practical problems, which would be equal to building computing devices capable of running precise simulations of our reality, also widens the gap between theoretical research and practical sciences. For the time being, humanity still needs to divide science and define human knowledge as a collection of individual theories elicited from scientific observations. The immense number of theories that comprise the collective human knowledge about every possible subject, as well as its extraordinary dynamics, force us to divide it into what we commonly refer to as knowledge domains, thereby reducing the contexts of our observations to smaller space-time continuums. The attempts to process the knowledge in a domain with computers have taught us that we need to recognize the reality of the "knowledge acquisition bottleneck" (Feigenbaum 2003) and to not underestimate the importance of common-

sense knowledge (see (Blois 1984) and (Lenat and Guha 1989; Guha and Lenat 1990; Lenat 1995)). The particularities with regard to the context retention, acquisition, representation, transferability and applicability of domain knowledge, causes us to distinguish between different modalities of domain knowledge, and place them on what we refer to as the knowledge spectrum.

1.5.3 The knowledge spectrum

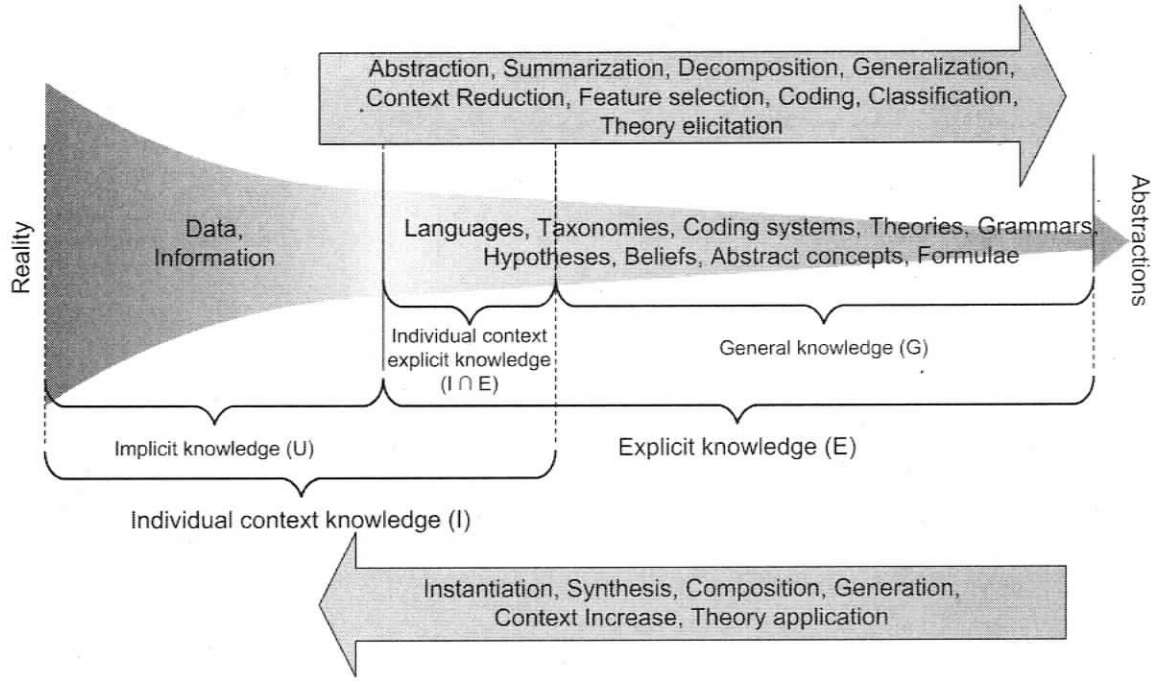


Figure 4. The knowledge spectrum

The knowledge spectrum (Figure 4) spans from a complex reality (the context of experimental data and information gathered from observations and measurements) to high-level abstractions (e.g., language concepts, coding systems, theories, grammars, taxonomies, hypotheses, beliefs, abstract concepts, formulae etc.). Therefore, it comprises increasingly lean modalities of knowledge and knowledge representations media and the relative boundaries and relationships between them. Two forces manifest on the knowledge spectrum: that of creating abstractions and that of instantiating abstractions for practical applications. The former is the theory elicitation and is synonymous to processes of context reduction, summarization, feature selection, coding, classification

and decomposition of knowledge. The latter, theory application, equates to processes of context increase, synthesis, instantiation and composition of knowledge. The engines behind the two knowledge spectrum forces are the knowledge processors, natural or artificial entities able to create abstractions from data and to instantiate abstractions in order to fit or change the reality.

Knowledge is traditionally categorized into implicit and explicit (Table 4, Table 5) and ranges from rich representations grounded in a reality, to highly abstracted, symbolic representations of that reality. The classical distinction between data, meta-data, information, knowledge and meta-knowledge is simplified by our subscription to the unified view of Algorithmic Information Theory (AIT) (Li and Vitányi 1997), which recasts all knowledge modalities and their processing into a general framework requiring a Universal Turing Machine, its programs and data represented as finite binary sequences. From this perspective a precise distinction between these modalities becomes unimportant.

1.5.4 Implicit and explicit knowledge

Implicit knowledge (*U*, from unobvious, unapparent) (Table 4) is the rich, experiential, sensorial kind of knowledge that a knowledge processor acquires when immersed into an environment (i.e., grounded in an environment), or presented with detailed representations of that environment (e.g., images, models, recordings, simulations). It is very well applicable to specific instances of problems and relies on processing mechanisms such as feature selection, pattern recognition and associative memory.

Example	The implicit knowledge used to recognize the face of a specific person.
Context dependence	Highly context dependent through retention of salient features.
Complexity	High complexity, extremely high dimensional, highly sparse representation space.
Acquisition	Detection, learning of correlations and regularities of environment (grounded in reality).
Structure	Unstructured, present implicitly in data recordings of the environment (e.g., image of a person).
Transferability	Transferable only in implicit form through the data recordings (i.e., representations) of the environment.
Applicability	Very well applicable to specific problem instances.
Processing mechanisms	Pattern recognition, feature selection, associative memory.

Table 4. Implicit knowledge (U)

Explicit knowledge (*E*) (Table 5) is the abstract, symbolic type of knowledge present explicitly in documentations of knowledge such as textbooks or guidelines. It requires a

representation language and the capability of a knowledge processor to construe the meaning of concepts of that language. It is applicable to both specific and generic problems and relies on explicit reasoning mechanisms.

Example	The explicit knowledge (e.g., textual descriptions) that would allow recognizing faces of people (including a specific person).
Context dependence	Variable dependence on context.
Complexity	Varies from high dimensional, sparse representation space to lean, more abstract, symbolic, low dimensional, compact representations (e.g., formulae).
Acquisition	Explicitation of one's implicit knowledge. Explicit acquisition of knowledge (e.g., through reading).
Structure	Varies from less structured (e.g., natural language) to very structured (e.g., formal descriptions, computer code).
Transferability	Transferable through languages (natural or formal) and communication (e.g., verbal).
Applicability	Applicable to both, specific and more generic problems.
Processing mechanisms	Reasoning.

Table 5. Explicit knowledge (E)

The distinction between implicit and explicit knowledge is useful to characterize the nature of human expertise, but becomes problematic when one wants to describe fundamental differences between theoretical and applied sciences: many applied sciences, especially knowledge intensive ones, in addition to general theories of problem solving, also make use of explicit knowledge in order to describe, with various degrees of precision, particular instances of problem solving and theory application. This represents the rationale for further dividing the knowledge spectrum into general and individual context knowledge.

1.5.5 General knowledge and the "frame problem"

General knowledge (*G*) (Table 6) is the explicit, abstract, propositional type of knowledge (e.g., guidelines), well applicable to context-independent, generic problems.

Example	Explicit general propositions, rules, algorithms, guidelines and formal theories for recognizing faces of people (e.g., a concise, formal theory of human face recognition).
Context dependence	Context independent.
Complexity	Very lean, abstract, symbolic, low dimensional, compact representations.
Acquisition	Identical to acquisition of explicit knowledge.
Structure	Very structured
Transferability	Highly transferable, explicitly as general propositions, guidelines, rules, computer code.
Applicability	Easy applicable to generic problems, difficult to apply to specific problem instances (e.g., recognition of the face of a specific person).
Processing mechanisms	Logic reasoning.

Table 6. General knowledge (G)

However, it is more difficult to use in specific contexts because of the gap between the general knowledge itself and a particular application context. This knowledge gap translates into uncertainty when a general knowledge fact is instantiated to a specific situation. For example, knowing generally that a certain drug may give allergic reactions but being uncertain whether a particular patient may or may not develop any, is an example of what we consider the uncertainty associated with general, context independent knowledge. The creation of general knowledge (i.e., abstraction, generalization, context reduction, theory elicitation) is a relevance-driven process done by "stripping away irrelevancies" (Blois 1984). This causes general knowledge to have a lower complexity and be more manageable: "generalization is saying less and less about more and more" (Blois 1984). Representations of general knowledge have been common in early artificial intelligence (AI) applications in the realm of expert system development. Such systems operated under the "closed world assumption" and were meant to make the representation of knowledge manageable, reproducible and clear. However this very assumption, which essentially postulates their context independence, also rendered the expert systems "brittle" or completely unusable when applied to real world problems (Luger 2002). Such representational approaches are known to suffer from a fundamental shortcoming, the "frame problem."

1.5.5.1 The frame problem

Daniel Dennett was the first philosopher of science who clearly articulated the "frame problem" and promoted it as one of the central problems of artificial intelligence (Dennett 1984) (also see (Pylyshyn 1987)). Janlert (Janlert 1987) identifies the frame problem with "the problem of representing change." In (Luger 2002) the frame problem is defined as "the problem of representing and reasoning about the side effects and implicit changes in a world description." Both definitions are clearly related to the need of dynamicity of representations. In order to articulate and circumvent the abstract nature of its definition, Dennett has invented a little story involving three generations of increasingly sophisticated robots. These fictitious robots are products of early artificial intelligence (AI) technology that use automated reasoning based on formal representations. These particular robots are specifically designed to solve a problem consisting of the retrieval of

their life-essential batteries from a room, under the threat of a ticking bomb set to go off soon. Although increasingly sophisticated in their reasoning, all three successive versions of the robot fail:

- The first robot fails by missing a highly relevant side effect of pulling the wagon with the batteries out of the room: the ticking bomb sitting on the same wagon was also retrieved, together with the batteries.

- The second robot did not finish its extensive, irrelevant side-effect reasoning procedures before the bomb goes off. As Dennett ironically puts it, the robot "had just finished deducing that pulling the wagon out of the room would not change the color of the room's walls and was embarking on a proof of the further implication that pulling the wagon out would cause its wheels to turn more revolutions than there were wheels on the wagon – when the bomb exploded."

- The third robot failed because it was "busily (i.e., explicitly) ignoring some thousands of implications it has determined to be irrelevant" and its batteries were therefore lost in the inevitable explosion.

The frame problem can therefore be recast as a problem of relevance (Pylyshyn 1987)[6], which is compounded by time constraints. It demonstrates that relevance judgment mechanisms based on general knowledge are time consuming and cause the failure to solve time-constrained decision problems. It is a problem *only* because in our dynamic and complex world we *do* have time constraints.

1.5.6 Individual context knowledge and case-based reasoning (CBR)

Individual context knowledge (*I*) (Table 7) (context dependent, instance specific knowledge *about* an individual), on the other hand, is a knowledge modality very well

---

[6] It is the preface of this work that contains the relevant paragraphs.

applicable to real problems, because it identifies and matches very well an application context.

Example	The implicit knowledge used to recognize and the explicit knowledge (e.g., textual description) that would allow recognizing the face of a specific person.
Context dependence	Context-dependent.
Complexity	Extremely high to high-dimensional, sparse representation space and complexity which varies from very high to lean representations (e.g., personal identifiers, small sets of highly unique features).
Acquisition	Identical to acquisition of both implicit and explicit knowledge.
Structure	Varies from unstructured to less structured.
Transferability	Transferable in both implicit and explicit form.
Applicability	Well applicable to specific problem instances, especially if context retention is high.
Processing mechanisms	Pattern recognition, feature selection, associative recall, case-based reasoning.

Table 7. Individual context knowledge (I)

The knowledge gap and uncertainty are reduced but still exist because of our dynamic, changing reality (time dimension) which may render individual context knowledge about the same patient collected in the past (e.g., value of blood pressure from a month ago), less applicable in the present or future. Because it preserves context (i.e., it is more grounded), individual context knowledge has a higher complexity than general knowledge and hence is more difficult to manage (i.e., has high memory requirements). For example, knowing the drugs and the precise description (e.g., numeric, textual, visual) of the allergic reactions that they caused in a certain person, as well as many other particular knowledge facts *about* an individual, is what individual context knowledge stands for. The uncertainty and knowledge gap related to the application of such knowledge to future instances of decision making involving that individual are reduced: individual context knowledge is supposed to fit very well an application context similar to that where it was originally captured.

1.5.6.1 Case based reasoning (CBR)

Individual context knowledge captured from a very specific context (e.g., diagnosing a particular patient with a particular disease) can be extrapolated to similar contexts. The higher the similarity between contexts, the smaller the knowledge gap and instantiation uncertainty and the higher the chances for a successful solution to a new problem. For this reason, individual context knowledge processing has become increasingly important for artificial intelligence applications and is defined as the approach to solving new problems based on the solutions of similar past problems (Kolodner 1993; Aamodt and

Plaza 1994; Watson and Marir 1994; Luger 2002). It has several flavours (e.g., exemplar-based, instance-based, memory-based, analogy-based) (Aamodt and Plaza 1994) which we will refer to inter-changeably, through the generic term of "case-based reasoning" (CBR).

There are four steps (the four "RE") that a case-based reasoner must perform (Aamodt and Plaza 1994; Watson and Marir 1994; Luger 2002):

1. RETRIEVE: the retrieval from memory of the cases which are appropriate for the problem at hand; this task involves processes of analogy-making or case pattern matching;

2. REUSE: the compositional adaptation and application of the knowledge encoded in the retrieved cases, to the problem at hand,

3. REVISE: the evaluation of solutions followed by revision if necessary, and

4. RETAIN: the addition of the current problem together with its resolution to the case base, for future use.

CBR entails that an expert system has a rich collection of past problem-solving cases stored together with their resolutions. This requires mechanisms that can deal with *high dimensional, rich representations of case data*. CBR also hinges on a proper management of the case base and on appropriate mechanisms for the matching, retrieval and adaptation of the knowledge stored in the cases relevant to a new problem. Essentially this functionality calls for *similarity-based conceptual retrieval of information* which brings CBR closer to what traditionally was the concern of information retrieval (IR) (Bichindaritz 2005). Ideally, the individual context knowledge in a case-base will progress asymptotically towards an exhaustive knowledge base, which represents the "holy grail" of knowledge engineers. From a learning systems point of view, similarly to artificial neural networks (Rumelhart, Hinton et al. 1986; Haykin 1994) and inductive inference systems (Solomonoff 2003) that learn from training examples, a CBR system

acquires new knowledge, stores it in a case base and makes use of it in new problem solving situations.

CBR approaches, devised originally as a solution to automated planning tasks (Schank and Abelson 1977), have since been used in various applications including healthcare, legal and military (e.g., battle planning) (Aamodt and Plaza 1994). This already shows a particularly good fit of a medical decision support based on CBR with its human users, the healthcare professionals.

1.5.7 The relationships between knowledge modalities

The absolute positions and shapes of boundaries between the four knowledge modalities, although admittedly not as precise as drawn on the knowledge spectrum in Figure 4, are not of importance for this discussion. However, the relative relationships between knowledge modalities are, and can be represented formally as a Venn diagram (Figure 5), which implies that:

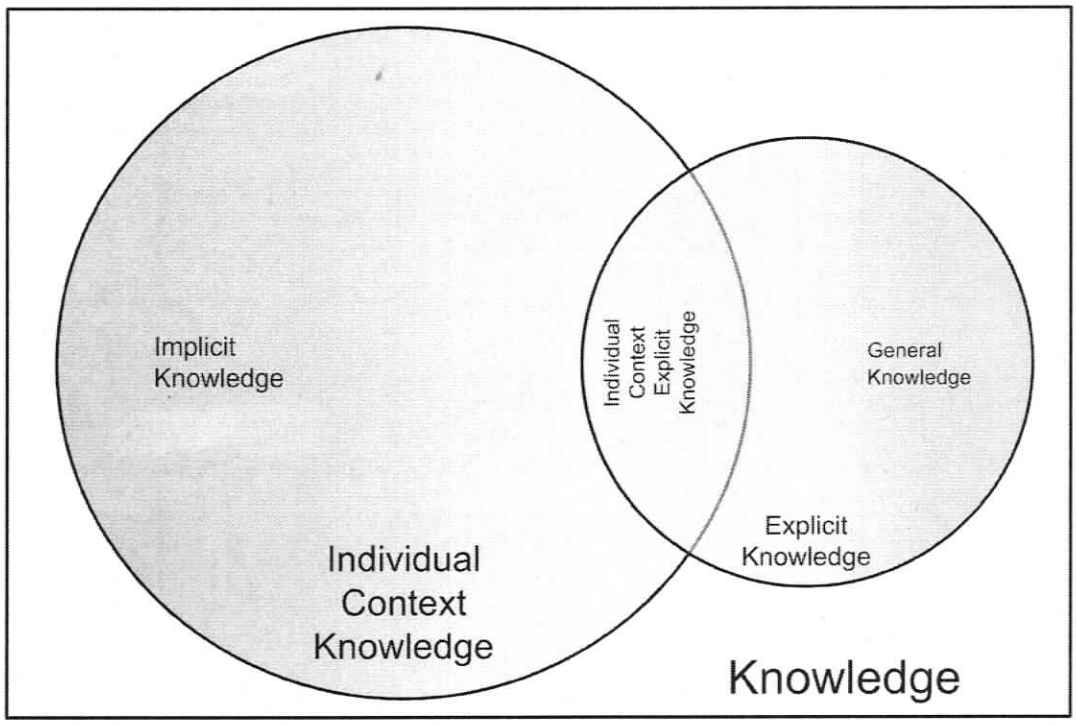


Figure 5. The relationships between the knowledge modalities

Individual context knowledge has a higher complexity than the explicit knowledge elicited from the same context (i.e., $C(I) > C(E)$). This is equivalent to stating that, for example, the picture of a person encodes more knowledge than the textual description of that person's appearance.

- Implicit knowledge is a subset of the individual context knowledge (i.e., $U \subset I$).
- General knowledge is a subset of the explicit knowledge (i.e., $G \subset E$).
- The set of individual context knowledge represented explicitly formed by the intersection of individual context knowledge with explicit knowledge is a nonempty set (i.e., $I \cap E \neq \varnothing$). This is equivalent to stating that it is possible, for example, for an explicit textual description to identify a context uniquely (e.g., the complete name and address of a person at a specified moment in time).

1.5.8 A meta-level view of Medical Informatics

Applied to the representation problem and to human and computer information processing and Medical Informatics in particular, a unified view is a meta-theory of Medical Informatics. If valid, such a meta-theory would have to account for anything carried on in Medical Informatics and could help to provide coherence to the Medical Informatics curricula of study (Moehr, Leven et al. 1982).

The meta-level overview of sciences and the definitions and properties of the knowledge spectrum and knowledge representation modalities enable us to draw some fundamental differences between theoretical sciences and applied sciences such as Medicine (Wieland 1975) and Medical Informatics. From this perspective, theoretical sciences (e.g., theoretical computer science):

- Make use of observations which are highly abstract symbolisms and create far more limited contexts of application of their theories, when compared to the complexity of the human body or of any social or biological system,

- Have as a primary purpose the creation of context independent, general knowledge comprising valid, powerful theories which explain precisely and completely the observations, and therefore,

- Include a relatively limited number of precise theories which are evaluated primarily by their power of explaining experimental observations, elegance, generality, and

- Are less concerned with the acquisition of the individual context knowledge required by practical implementations and by the application of results to real world problems.

Applied sciences such as Medicine and Medical Informatics, on the other hand:

- Gather extensively data and observations (knowledge *about* individuals) from very complex systems (Blois 1984; Shortliffe and Blois 2001) (e.g., human body), which are characterized by high individual variation and randomness;

- Have as a primary purpose not only distilling data and observations into general knowledge, but are also concerned with the implementation details and with the application of theories to individual problem solving (e.g., diagnosis and treatment of real patients),

- May lack the incentive to refine existing theories which are objectively wrong as long as practical success is achieved (Wieland 1975; Moehr 1989) ,

- Contain very few simple, "elegant" generally applicable theories (general knowledge) that can solve individual problems completely or explain and predict accurately (Blois 1984), because of the complexity of the human body and its individual variation and, therefore,

- May pursue the application of a multitude of mutually contradictory, poorly grounded, context-independent, general theories (e.g., the general theory of

medical reasoning and the concepts of "diagnosis" and "symptom") (Wieland 1975; Moehr, Leven et al. 1982),

- Abound in general theories (e.g., guidelines) which are "lossy" (i.e., ignore individual context variation) and which are evaluated statistically by their practical success relative to existing ones (e.g., cancer therapy),

- Attempt to make up for the knowledge gap between context-independent general knowledge and the reality (i.e., the context) where knowledge is applied, by employing experienced clinicians who require extensive training and recently information technology (e.g., decision support), and, in addition,

- Are compounded by time-constrained circumstances and largely unsolved ethical issues (e.g., privacy and confidentiality, genomics research).

1.5.9 Conclusions

Given the special circumstances of our applied science in the context of other sciences and the increasing recognition of the importance of knowledge processing to Medical Informatics (Musen 2002), it follows logically that Medical Informatics should complement the traditional quest for generally applicable biomedical knowledge with the advance of acquisition, storage, communication and use of context dependent, individual context knowledge. By doing so, Medical Informatics will provide a solution to the problems that arise during the use of general knowledge and, in the same time, will enable clinical research as well as advanced decision support and education of healthcare providers, patients and health informaticians.

Processing individual context medical knowledge equates to a CBR approach that employs collections of patient cases. Currently, such collections are the focus of research on Electronic Health Records (EHR). Envisioned as "womb to tomb" collections of patient-specific data, EHR could contain a wealth of data that could be used to support case-based decisions. If the EHR are to be realized in the future and used in a CBR context, the issues pertinent to the design of case-bases automatically become pertinent to

the design of EHRs, and the CBR paradigm becomes important to Medical Informatics. The proposed meta-theory of Medical Informatics seems to be centered on collections of patient-specific data (i.e., cases) organized and exploited in accord to CBR principles.

1.6 AN ASSOCIATIVE MEMORY MODEL FOR CONCEPT SPACE REPRESENTATION

In this section, the entire chapter is summarized and the need for natural language processing (NLP) and information retrieval research are underlined. Some important aspects such as dynamicity (frame problem), multidimensionality (case descriptions) and the similarity-based or associative organization implied by the case based reasoning paradigm are introduced. It is further suggested that the representation problem can be recast and unified around the notion of memory models capable of representing "associative concept spaces" characterized by four specific properties: multidimensionality, sparseness, dynamicity and associative (similarity-based) organization. Finally, the thesis is proposed that in order to achieve the advanced information processing required by Health Informatics applications, one has to devise approaches that efficiently address all four specific properties of associative concept spaces.

1.6.1 A summary of the chapter

So far, we have seen how the "axiom of information systems" (i.e., building systems that are both usable and useful at the same time) has lead naturally to the fundamental problems of informatics, i.e., the problem of representation, as well as to a proposal for its solution: creating information systems which have the capacity to acquire, with as high degree of autonomy as possible, useful, relatively complete, problem specific representation of complex problems that need to be solved. Further investigation has revealed that, fundamentally, the process of representation is a function that maps the reality that includes the object to be represented onto a representation medium and that the amount of context dependence is what fundamentally differentiates various representation functions.

Stepping up at a meta-level, we have seen that science comprises the creation of theories as well as application of theories. These two opposing forces are essentially movements on the knowledge spectrum involving four interconnected modalities of representing human knowledge and which could be essentially characterized through their context dependence as well. Context-independent, general knowledge is connected to the "frame problem," a fundamental issue of artificial intelligence that underlines the extreme importance of *representing change*, through *dynamic approaches to representation*. Context dependent, individual context knowledge is intimately connected to case-based

reasoning (CBR), i.e., the method of knowledge processing that aims at solving new problems based on the solutions to similar past problems and which requires rich, *high dimensional representations* of past problem solving instances, as well as *similarity-based organization* of such representations to facilitate their similarity-based retrieval.

Having identified the high relevance of CBR (i.e., context dependent representation and processing of knowledge) to Medical Informatics we now turn toward the practicality of its application.

1.6.2 The applied perspective

The overall knowledge processing capacity of healthcare systems is distributed between two sources: human resources (e.g., healthcare professionals) and information technology (Medical Informatics).

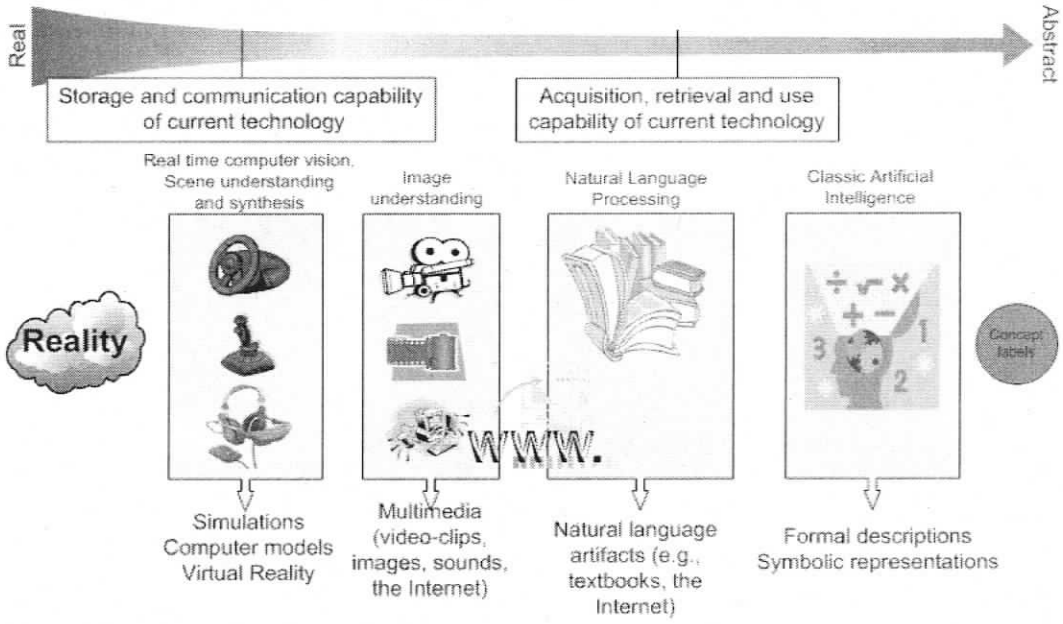


Figure 6. Knowledge representation media on the knowledge spectrum. The storage and transmission of knowledge are more advanced compared to the knowledge acquisition, retrieval and use capability of current technology

An ideal CBR approach would increase this knowledge processing capacity and verify the *usability axiom* of information systems by allowing for both *usable* and *useful,* automatic processing (acquisition, representation, storage, retrieval and use) of context

dependent, individual context knowledge present in increasingly rich knowledge media such as natural language artefacts, images, videos and complex computer simulations of reality (Figure 6). The storage and communication of such knowledge are well advanced by current information technology. However, most of the acquisition, retrieval and knowledge use are, and will continue to be the task of professionals until advanced, context dependent processing (e.g., real-time computer vision, scene understanding and synthesis, image understanding, robotics, natural language understanding) are widely applicable. Given the widespread use of natural languages as knowledge representation and communication media, it follows that *natural language processing* (NLP) and *information retrieval* research (Bichindaritz 2005) are very important components of Medical Informatics, required to advance the organization and processing of individual context knowledge in reusable case-bases. This goal to advance context-dependent processing of increasingly complex knowledge representations (e.g., natural language, sounds, images, simulations) and create intelligent machines that can hear, see, think, adapt and make decisions, emphasizes even more the importance of Artificial Intelligence (AI) to Medical Informatics. Finally, because the knowledge processing capacity of human resources tends to remain relatively constant, moving towards the ideal of individual context knowledge processing, no matter how slowly, may also have ethical implications because it proves that medical informaticians are trying to do everything they can in order to serve the interest of the individual.

1.6.3 Conclusions, thesis and significance of work

The extreme importance of human-like, context-dependent representations for Medical Informatics is fully supported by the arguments presented in this chapter. However, it has also become clear that human-like, context-dependent representations are more complex and hence more computationally expensive than context-independent ones. The aims of this dissertation consist of exploring the nature and the fundamental properties of such representations as well as proposing, implementing, demonstrating and evaluating possible solutions to achieve context-dependent representation and processing while remaining computationally feasible by current technology. The high relevance of the

"frame problem" (requiring approaches to representing change) and CBR (requiring high dimensional representations and similarity based retrieval) as well as additional theoretical and empirical work (Kanerva 1988; Pantazi, Arocha et al. 2004; Pantazi and Moehr 2004), strongly suggest that achieving human-like, context-dependent representations could be recast and unified around the notion of memory models capable of representing *associative concept spaces* characterized by the following fundamental properties:

1. *Extremely high dimensionality*, i.e., possessing thousands of dimensions,
2. *Extremely sparse structure*, i.e., containing only a tiny fraction of the theoretically possible number of entities,
3. *Dynamicity* (i.e., adaptive, changing, evolving), and
4. *Similarity-based organization.*

1.6.3.1 The thesis

The thesis of this dissertation is that human-like, similarity based, context-dependent representations required by Medical Informatics applications are most appropriate in memory models capable of managing *associative concept representation spaces* and specifically addressing each and every one of the four fundamental properties of associative concept spaces. The direct implication of the thesis is that not addressing either one of the fundamental properties of associative spaces may result in limitations that can render representations less appropriate to Medical Informatics. For example:

- a reduced dimensionality of representation leads to compact, less-sparse representations that lose their context and become too abstract to be applicable to individual problem solving,
- failing to cater for the extremely sparse nature of high dimensional representations results in representations which are wasteful spatially and lack dynamicity,

- a lack of dynamicity of representation results in fixed, unchangeable representations whose updating, evolution and adaptation are impeded,
- a lack of a similarity based organization of representations hinders similarity-based processing and retrieval, a fundamental process of recognized importance.

In addition to the theoretical accounts, this thesis is supported by empirical work on a *model of deterministic dynamic associative memory* that will demonstrate human-like, similarity based, context-dependent representation and processing of textual information in natural language processing (NLP) and information retrieval (IR) tasks.

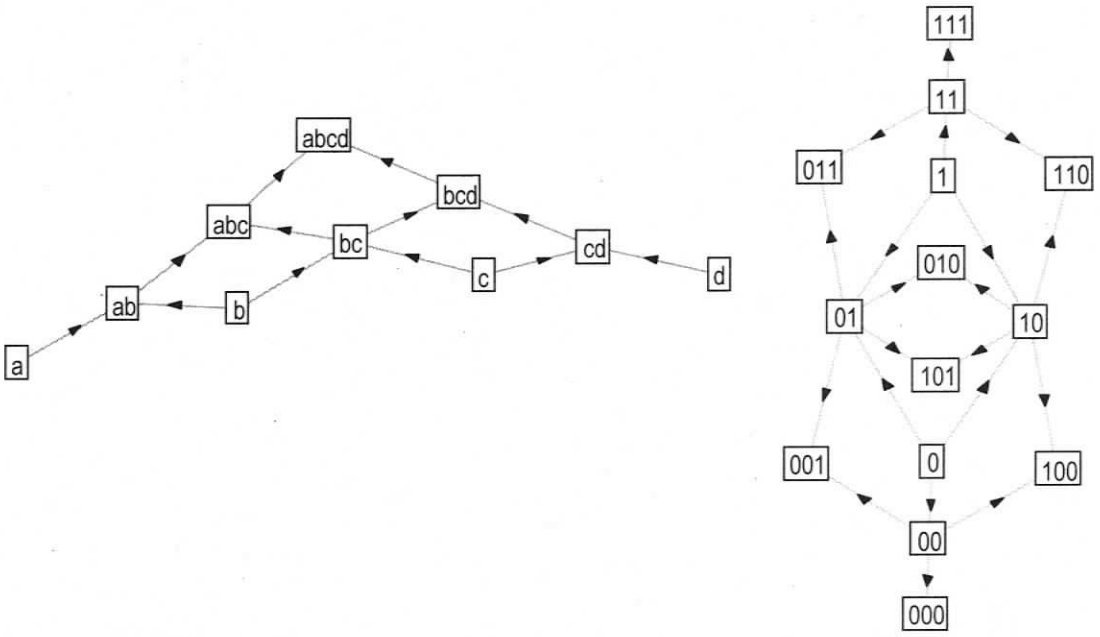


Figure 7. Examples of partial order sets (posets)

Existing computational models (Kanerva 1988; Kohonen 2001) strongly suggest that one of the most efficient ways (if not the only one) of dealing with the high dimensionality,

extreme sparseness, and dynamicity of representation is by linking entities explicitly in richly connected structures (i.e., graphs such as in Figure 7).

Specifically, the computational model that forms the focus of this dissertation is based on partial order sets or *posets*. Posets are generalizations of trees and can be depicted using Hasse diagrams such as in Figure 7. Formally, a poset is a base set together with a reflexive, antisymetric and transitive binary relation on that base set. Informally, posets could be thought of as generalizations of ordered sets in that they allow elements between which ordering relations do not exist (e.g., "c" and "d" or "abc" and "bcd" in Figure 7, left and "10" and "011" in Figure 7, right). Because posets are a more general way of representing information they could form the basis of more general information processing functions, however, at the expense of computing power.

Imposing similarity-based principles on their design where similar elements are close (e.g., "cd" and "c" or "abcd" and "abc" in Figure 7, left), effectively turns such structures into associative representations (or associative memories) that could grant access to similarity-based retrieval as well as to other high level information processing functions such as unsupervised pattern discovery and acquisition (Pantazi and Moehr 2005).

1.6.3.2 Significance of work

Context dependent information processing applications will fulfil a demonstrated desideratum for processing information on conceptual principles. For example:

- Case based reasoning (CBR), an area of research relevant to knowledge intensive problem solving such as medical decision making, could be improved by the advances in associative representations and similarity based retrieval proposed in this dissertation;

- User interfaces of complex medical information systems (e.g., diagnosis coding) could be dynamically simplified (e.g., long scroll down lists shortened) based on the context-dependent relevance (or similarity) of their content to the situation at hand; context could be assessed through recognition of patterns of sensor data,

diagnoses, symptoms, medication using pattern discovery and recognition models such as the ones proposed in this dissertation;

- Improving access control to personal and general medical information could be achieved through processing on conceptual principles the contextual knowledge about users that could be represented using principles such as those proposed in this dissertation;

- Improving the control of delivery of medical alerts could be based on contextual cues such as the recognition of workflow patterns and of the importance and of the relevance of a trigger to a certain context that could be recognized using mechanisms similar to those proposed in this dissertation;

- The robustness of alternative user input interfaces such as handwriting, gesture and speech recognition, natural language processing, image and scene recognition could be improved by context dependent processing of sensor data along the lines proposed in this dissertation;

- Indexing and retrieval of information on conceptual principles could lead to advances in automated searches for literature relevant to a certain subject matter, a possibility which has the potential to revolutionize information retrieval, which is an area of research highly relevant to this dissertation.

Finally, one major underlying assumption of this dissertation is that approaches to context-dependent information processing hold the key for a solution to the medical information technology paradox and that the difficulties to implement them arise only from their inherent complexity and from the amount of memory and processing power they require in order to overcome the stage of "toy applications" that is still characteristic of many health informatics systems.

1.7 GOALS, SCOPE, METHOD AND LIMITATIONS

In this section it is suggested that Medical Informatics is a young field of research which calls for work that reveals fundamental issues. A definition of the field is proposed from this perspective. The research methodology employed in this dissertation is strongly impacted by the complexity of the research and relies heavily on the processes of modeling and simulation that can be easily placed in the meta-level framework proposed earlier as well as on a literature review approach based on full text search approaches and which has allowed a broad literature review difficult to achieve by traditional methods.

1.7.1 Goals and scope

Medical Informatics is a relatively new discipline with a theoretical foundation still in its infancy. Therefore one of the main goals of this dissertation is to reveal and propose solutions to fundamental issues, including an appropriate definition of this field of research. As a consequence, the scope of this dissertation is broad, interdisciplinary and the research methodology is eminently exploratory and empirical in nature. So far, the identification of the extreme importance of context-dependent medical information representation and processing has led to the identification of "the management of dynamic, extremely high dimensional but extremely sparse spaces" as an important issue of Medical Informatics as well as to defining Medical Informatics itself as "context dependent medical information representation and processing" (Pantazi, Kushniruk et al. 2006). Given the broad implications and scope of this research, by removing the medical connotations, the insights gained here can be easily generalized to address general theoretical Artificial Intelligence issues such as the frame problem, case based reasoning, planning, intelligent agent design and decision support.

1.7.2 Methods and limitations

1.7.2.1 Modeling and simulation

Modeling and *simulation* in general are considered essential for intelligent behaviour in tasks such as learning and decision-making. Precisely, mental modeling and simulation processes are cognitive functions which have been recognized as an integral part of

naturalistic decision models (Klein 1999). For these reasons, modeling and simulation are also natural methods for answering research questions, especially when enhanced by the use of information technology. In addition, they fit very well into the meta-level framework described earlier and could be easily likened to the two opposite forces that manifest on the knowledge spectrum. The fact that the methodology of this research fits well into the general framework proposed here could only be a positive sign that speaks in favour of the validity of both the framework and the methodology.

The process of *modeling* could stand for creating abstract models of reality while the process of *simulation* could stand for instantiating abstract models and theories in order to achieve practical applications and to allow for the evaluation of the validity of new models and theories. Therefore modeling equates to theory creation while simulation is more akin to the theory application and while models tend to sit on the abstract side of the spectrum, the prototypes are closer to the reality side of the spectrum (Figure 8).

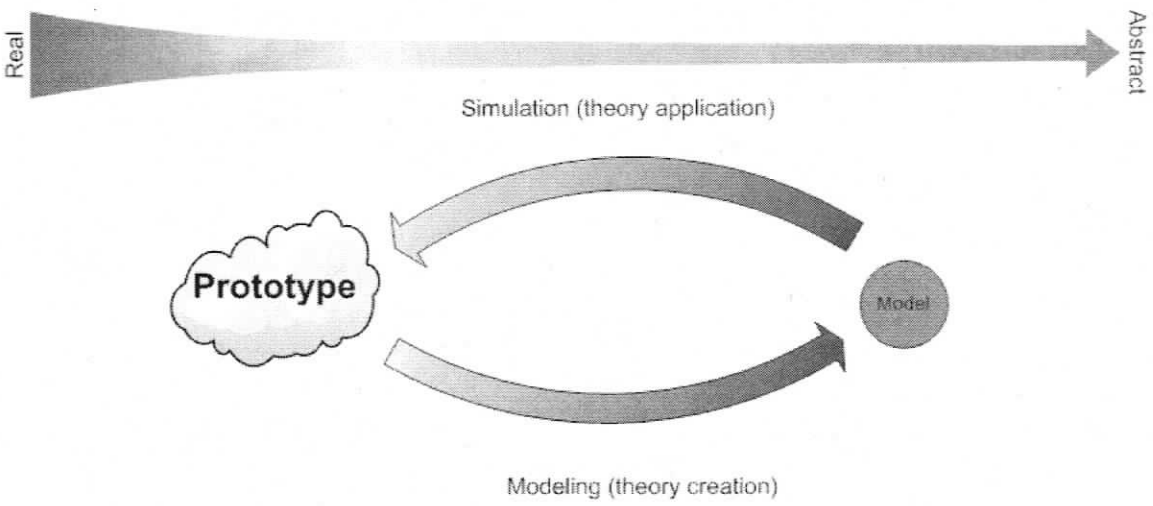


Figure 8. The processes of modeling and simulation on the knowledge spectrum.

Prototypes could be considered more complex because they also incorporate the additional knowledge about the intricacies of their implementation in a particular simulation environment (e.g., using a certain programming language, on a certain type of computer, operating system, etc.). The sometimes considerable knowledge gap (or "know how") between a model and its prototypes is often the ground for the commercial competitiveness of the various applications of a published theoretical model.

Computers are versatile information processing tools which serve as representation media for new information processing models and could simulate the behaviour of such models when applied to practical problems. This versatility can be fully exploited through a *rapid prototyping approach* and Rapid Application Development (RAD) tools. As suggested in Figure 9, computers and information technology could become a powerful and unique environment that allows for modeling, simulation and evaluation of new information processing models in ways impossible to achieve through alternative means.

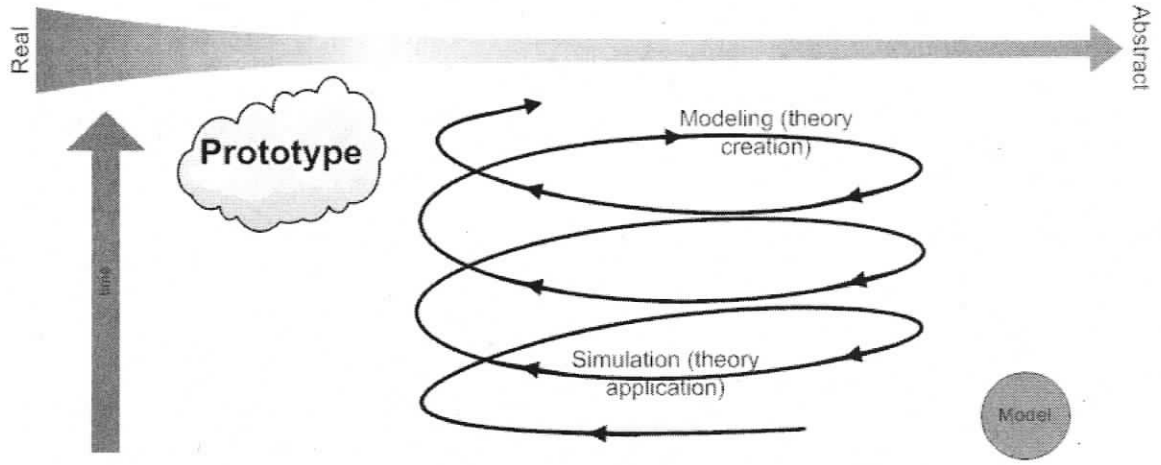


Figure 9. The rapid prototyping approach on the knowledge spectrum. The multiple modeling-simulation iterations are depicted as the loops of a spiral spanning over the time dimension.

For example, the possibility to test for thousands of variants of complex, working prototypes would be unconceivable without such an environment. The model of associative memory introduced in this dissertation, a complex connectionist system (i.e., a graph) with millions of vertices and tens of millions of edges, fits well this description and its development indeed has required modeling and simulation of thousands of prototypes.

Currently, the limitations of modeling and simulation are functions only of the computing power memory of actual computers as well as of the complexity of the model that is to be simulated. Today, associative memory models with millions of units are feasible and scaling up the results of this research to a full-fledged, real-world application, may need additional computing power. However, the limitations which arise from computational demands are relative to today technology and could be overcome by distributed

processing architectures (e.g., cluster computing, grid computing) (Moor, Norager et al. 2005).

1.7.2.2 Literature review approach

Reviewing relevant literature is essential to any research. Equally essential is the process of finding the relevant literature to review. The main research object of Information Retrieval (IR), the process of literature search and retrieval would ideally yield documents that are conceptually similar and hence relevant to a given description (i.e., representation) of one's area of research. The ability to perform conceptual comparisons between literature articles and other knowledge artefacts is therefore crucial and underlines once again the relevance of IR and NLP to this research which argues for the extreme importance of managing *associative concept representation spaces*. The advent of highly successful large-scale full text search engines (e.g., Google) also demonstrates the universality of the need to search for information given the similarity with a search query. However, the fact that most existing information retrieval technology is still limited in meeting this ideal also speaks to the need for further research on representing and retrieving information on conceptual principles such as those suggested in this dissertation.

In this dissertation, the existing search engine technology used to retrieve literature was complemented by full-text indexing and retrieval using existing desktop search technology (e.g., Google Desktop Search) on a literature database maintained by this author. The literature database, which amounts to around 10,000 text, images and multimedia files, contains documents relevant to this research which have been read, written and downloaded by this author during the last 5 years. While the fact that the accumulation of an exhaustive list of all the relevant literature to this research cannot be proved unequivocally, the validity and usefulness of such an approach could be estimated if one submits to the commonsense observation that any researcher who is keenly interested and tries to keep abreast of a research topic, will eventually accumulate, given a reasonable amount of time (in this case 5 years), most of the relevant literature written on that subject, especially if the topic in question is not overly dynamic in nature. In this

case, the computer files accumulated on that researcher's computer could be considered a good representation of, and would be able to define the conceptual space of the chosen research topic reasonably well. This also suggest that a full text indexing and retrieval approach on such a collection of documents may actually be the closest thing one could get to a complete representation of a research topic, from the particular perspective of a certain researcher such as this author. A dissertation such as this one then could be regarded a synthesis and an extension of the knowledge captured by a particular document collection about a certain research topic.

Chapter 2 THEORETICAL BACKGROUND

This chapter contains arguments and discussions of theoretical underpinnings of issues introduced previously and of the proposed solution in the light of existing literature, approaches and solutions for the management of associative concept spaces. It also contains a broad literature review of seemingly disparate theories, models and approaches that nonetheless converge and could be unified under the general umbrella of "approaches to associative concept space representation," as well as an in-depth discussion of design principles of a memory model that is able to address each one of the four fundamental properties of associative concept spaces (high dimensionality, sparseness, dynamicity, similarity based organization).

2.1 CASE-BASED MEDICAL INFORMATICS

This section opens the chapter with a case based reasoning perspective on medical informatics decision making in medicine and knowledge representation and processing. The focus is on fundamental aspects of decision-making, which connect human expertise with individual context knowledge processing. Further, a knowledge spectrum perspective on biomedical knowledge is used to demonstrate that case-based reasoning is the paradigm that can advance towards personalized healthcare and that can enable the education of patients and providers.

2.1.1 Decision Making in Medicine

Medicine is a knowledge intensive domain where time-constrained decisions based on uncertain observations are commonplace. In order to successfully cope with such situations, health professionals go through a tedious learning process in which they gain the necessary domain knowledge to evolve from novices to experts. As experts, health professionals have attained, among other things, two important, highly interrelated abilities:

- To be able to reduce the knowledge gap between knowledge facts and reality which translates into being able to reduce the uncertainty of knowledge instantiation to a particular context, and
- To be able to reduce knowledge complexity by determining efficiently what is relevant for solving a problem in a particular situation.

For example, both the presence and the absence of a past appendectomy are relevant and contribute (potentially unequally) to reducing the uncertainty of instantiation of the biomedical knowledge of an expert to a particular context of a patient with right lower abdominal pain. Fundamental to decision making, relevance judgments and uncertainty reduction seem both closely connected with the quality and quantity of knowledge available for solving a problem as well as with the nature of knowledge processing mechanisms. Studies of expert-novice differences in medicine (Patel, Arocha et al. 1994) have shown that the key difference between novices and experts is the highly organized knowledge structures of the latter, and not the explicit strategies or algorithms they use to

solve a problem. This is supported by expert system development experiences, which showed that a system's power lies in the domain knowledge rather than in the sophistication of the reasoning strategies (Luger 2002). Studies of predictive measures of students' performance indicate tests that measure the acquisition of domain knowledge to be the best predictors (Kuncel, Hezlett et al. 2001). The work on naturalistic decision-making (NDM) and the development of psychological models of "recognitional decision-making" such as the Recognition-Primed Decision (RPD) (Klein 1993; Klein 1993; Klein 1999), suggest the heavy dependence of decision makers on their previous experience of problem-solving and also on their ability to perform mental simulations.

2.1.1.1 Time constrained decision making

The discussion around the amount of problem solving experience of a decision maker becomes critical in time-constrained decision circumstances. The exhaustiveness of the knowledge base and the efficiency of retrieval mechanisms now become paramount to the decision speed. Empirical evidence that shows the existence of "systematic changes of cognitive processes" related to time stress, comes from the studies on the psychology of decision-making under time constraints (Svenson and Maule 1993). Although most of these studies attest the overall negative effect of time stress on the "effectiveness of decision-making processes" (Zakay 1993), others (Klein 1993; Klein 1999) argue that even extremely time-constrained situations could be handled successfully by human subjects, given enough expertise (i.e., enough problem solving experience).

Since humans are able to make sound relevance judgments and reduce instantiation uncertainty of knowledge most of the times, the following questions arise: What is their strategy for increasing the exhaustiveness of their knowledge base while managing the complexity of its concept space? How do they represent and organize their knowledge and how do they manage time-constrained situations? At least some of these questions have been under intense scrutiny that has resulted in important empirical work on naturalistic decision-making (Klein 1993; Zsambok and Klein 1997; Klein 1999; Salas and Klein 2001). Important insights have been gained at the individual but also at the organizational and social levels.

2.1.1.2 The knowledge spectrum perspective

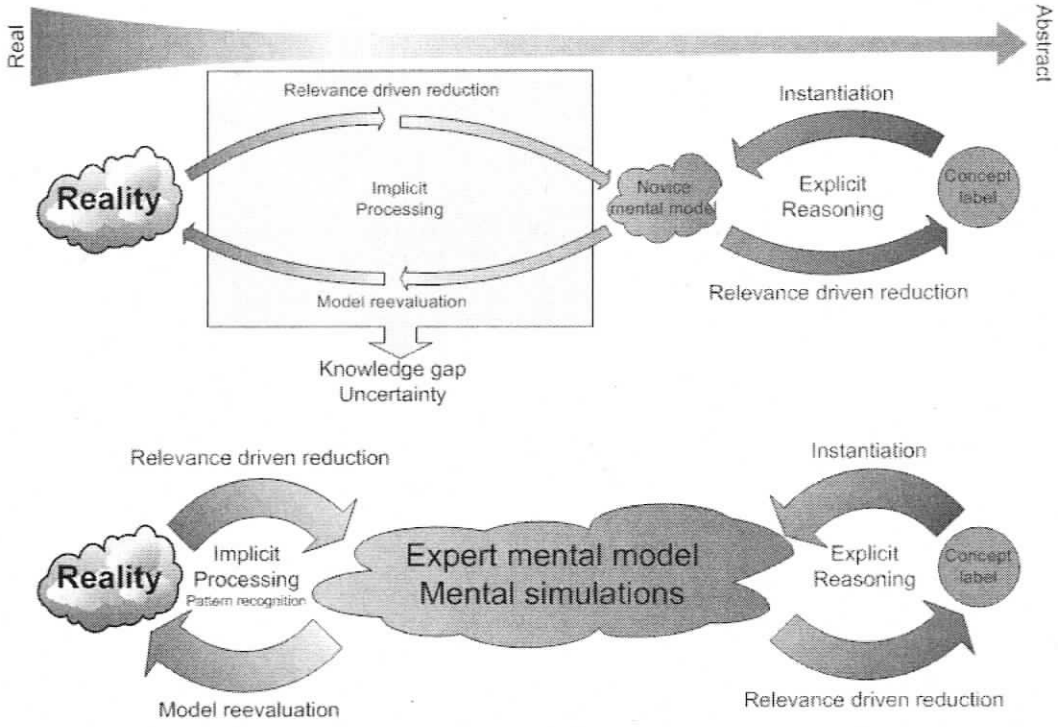


Figure 10. Knowledge representation and processing in novices and experts

From the perspective of the knowledge spectrum, it seems reasonable to associate expert decision makers with individual context knowledge and novices with the more abstract general knowledge about a subject, available in explicit knowledge artefacts (e.g., textbooks, guidelines). It is also conceivable that mental models of experts span a great length of the knowledge spectrum, causing them to efficiently perform implicit processing (feature selection, pattern recognition, associative recall) as well as just-in-time explicit reasoning (Figure 10). The ability to move freely across the knowledge spectrum causes experts to efficiently reduce data to abstractions and to create hypotheses and micro-theories through sound relevance judgments. The powerful mental simulations that experts can perform allow them to construe the appropriate meanings of concepts and to verify their hypotheses against contexts of reality.

Novices, on the other hand, have limited mental models of reality situated towards the abstract region of the spectrum. This causes them to have difficulties with construing appropriate meanings of concepts due to the wider knowledge gaps between their mental

models and reality. Novices are therefore unable to make sound relevance judgments and limited in their ability to interpret data and to create abstractions. They are also usually overwhelmed by the explicit, general knowledge present in textbooks and guidelines and unable to fully construe the meanings of concepts present in such knowledge artefacts.

2.1.1.3 Conclusions

In conclusion, in information and knowledge intensive domains such as medicine, explicit reasoning is important but individual, context-dependent knowledge acquisition (i.e., experience) and processing (i.e., CBR) are crucial for decision-making. This conclusion underlines the importance of high-dimensional representations of individual context knowledge and supports the thesis of this dissertation.

Because the nature of expertise seems largely connected with individual context knowledge processing, it follows that the evolution of novices into experts is not attainable by the provision of extensive general knowledge alone. Therefore, not only the self learning but also the collective sharing of contexts of experiences (e.g., case records, personal stories, etc.) between individuals and between generations, contribute to the way humans deal with decision problems. This conclusion supports the thesis of this dissertation by advocating the need for dynamic representations of knowledge which allow it to be continuously updated, as experience accumulates.

2.1.2 Patient-centered vs. population-centered healthcare

The major driving force of science is universally applicable knowledge (i.e., general knowledge). While creating and communicating new knowledge, scientists move across the knowledge spectrum from the data that captures the reality of their experiments and observations, towards abstract representations that allow them to communicate their theories. In biomedical research, such an example is the randomized controlled trial (RCT), currently regarded as the gold standard for knowledge creation.

2.1.2.1 Randomized controlled trials

The correct design of an RCT is crucial for the validity of the medical evidence obtained. A correct randomization process in RCTs will limit the bias and increase the chance for applicability of the evidence obtained, to a specifically selected group of patients (e.g., "women aged 40-49 without family history of breast cancer"). However, at the same time, the randomization process removes the context of individual cases and creates a knowledge gap between the RCT evidence and future application instances. As with any statistical approach, the RCT-based evidence is best applicable at the population level rather than at the individual level.

2.1.2.2 Decontextualization of medical knowledge

This depersonalization and decontextualization of medical knowledge and evidence was also noted by others (Kovac 2003; Fierz 2004) and could also be illustrated by the observation that most patients feel relieved when told that the chances of being successfully treated for a certain condition are 99%, for example. Although this is psychologically very positive, the patients should not necessarily be relieved, as they could very well happen to fall among the 1%, for whom things could go wrong and for whom, usually, the RCT-based evidence is inconclusive. An experienced physician and, from a CBR perspective, a highly efficient case-based reasoner, is most of the time able to individualize the medical decision for a particular patient for whom things are likely to go wrong and fill in the knowledge gap between the RCT evidence and the context of the medical problem at hand. This could lead to avoiding a therapeutic procedure recommended by the medical evidence. The individual context knowledge that this decision is based on is usually not provided by the RCT, but is acquired through a tedious process of training. This decision is often so complex that it cannot be easily explained as it becomes heuristic in nature and is motivated by the individual context knowledge that a decision maker possesses. Others (Patel, Kaufman et al. 2002) have also pointed out that when physicians manage their cases (e.g., diagnosis and treatment), their previous experience allows them to make informed decisions based on heuristics rather than on a sound, complete and reproducible reasoning, such as logical inference based on a predicate calculus representation of a problem. In addition, human experts often disregard

probabilistic, RCT-type evidence and consistently detach themselves from the objective models of classical decision theory (e.g. probability theory, Bayes theory) in favour of heuristics-based approaches. Although prone to occasional failures, heuristics-based decisions are much more efficient in time-constrained and uncertain situations (Klein 1999).

2.1.2.3 The knowledge spectrum perspective

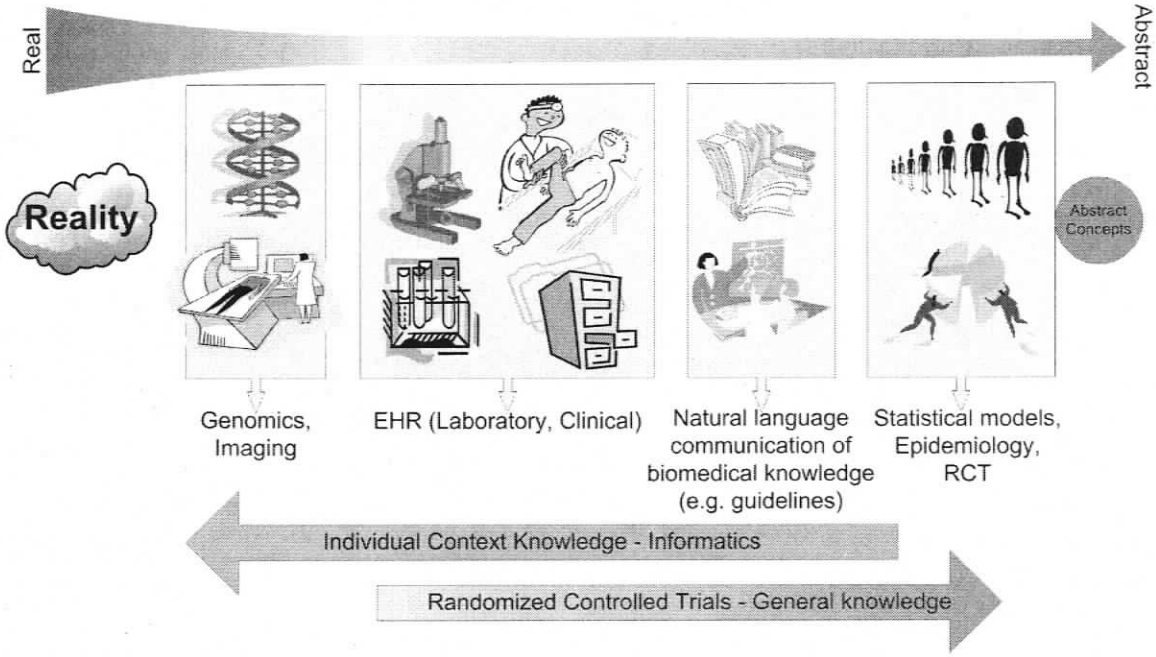


Figure 11. Biomedical knowledge on the knowledge spectrum

From the perspective of the knowledge spectrum, the driving forces of Health Informatics and RCT methodology seem to have opposite directions: while Health Informatics aims towards individual context knowledge and personalized health care, general knowledge gained through populational studies (e.g., RCTs) targets the ideal of universal applicability (Figure 11). For example, the value of a single bit of data (e.g., a Yes/No answer to a specific question such as a past appendectomy) can be very relevant in a decision-making context if it reduces the uncertainty of other knowledge. However, such individual bits of data are inevitably lost during the creation of general knowledge and can only be preserved in individual records such as Electronic Health Records (EHR).

2.1.2.4 Electronic Health Records

Medicine has always been and will always be a case oriented profession. Medical Informatics has recognized this early through the works of various researchers who pioneered the area of decision support systems (Miller 1994). The attempts to enhance early decision support systems with domain knowledge from simulated patient cases (Parker and Miller 1989) are also relevant to CBR work.

One very effective form of medical education is the retrospective analysis of case records where health professionals, both experienced or novices, learn from their own and from others' successes and failures (Greene, Hsu et al. 1996). Providing that legal and ethical implications such as provider and patient protection are dealt with appropriately, it is conceivable that the efficacy of this teaching method could be improved if case records are continuously created, enriched, accumulated and organized on similarity principles. This would be possible through a CBR approach of the EHR which, from this perspective, could serve as a comprehensive case base of managed patients that will evolve asymptotically towards an exhaustive knowledge base.

2.1.2.5 Conclusions

The recognition of the importance of CBR to medicine has led to an increase of the exploration of CBR in medical contexts, in recent years (Macura and Macura 1997; Montani, Bellazzi et al. 1998; Montani and Bellazzi 1999; Armengol, Palaudàries et al. 2000; Fritsche, Schlaefer et al. 2002; Armengol and Plaza 2003). Regardless of the problem nature, the most important technical components of a CBR expert system are the representational approaches for the case-base (i.e., the memory of past problem-solving instances) and the case matching or pattern matching procedure that retrieves relevant cases for a certain problem. While humans seem to possess a natural support for these two components, there remains significant work to be done in order to make the computer support this kind of knowledge acquisition and processing. The associative memory model introduced in this dissertation aims precisely at this goal.

Rigorously and expensively collected, general, population level knowledge is useful only in situations where individual context knowledge lacks (e.g., new drugs), providing the

decision makers have access to it and are able to apply it to specific contexts. However, general knowledge is unlikely to be used as such in many naturalistic decision-making processes because it does not support the way expert decision makers think. In addition, the knowledge gap and inherent instantiation uncertainty manifested in the application of context-free, general knowledge, does not fully enable the education of providers and patients. A more complete educational approach would also require the additional knowledge from individual contexts of successful or unsuccessful application instances such as those collected in EHR. Medical Informatics, by advancing context-dependent, individual context knowledge processing through the development of EHR, provides an alternative solution to the problems that arise from the use of general knowledge that targets universal applicability. An integral part of individual context knowledge, the use genomic data is also recognized (Kovac 2003; Fierz 2004) to be of extreme importance for a solution to the problems of general knowledge.

Finally, the support for a thesis which advocates the need for high dimensional, case based representations in Medical Informatics comes in two complementary forms. First by revealing the shortcomings of low dimensional representations of knowledge which, in many instances, are too abstract to be significant to individual problem solving, and second by advocating the need to advance case based, context-dependent, individual context knowledge processing, which translates into the need for information processing models that are able to cope with high dimensional representations.

2.2 KNOWLEDGE REPRESENTATION AND PROCESSING

This section is a case based reasoning perspective on knowledge representation and processing. The completeness of formal languages and their connection to the frame problem is examined in detail. Further the need for approaches to deal with ubiquitous but ambiguous natural language descriptions is advocated.

Languages are media for representing and transferring explicit knowledge and, roughly, can be categorised into formal languages, usually determined by strict specifications and natural languages, which allow for a certain degree of ambiguity and redundancy in representation.

2.2.1 Formal languages

2.2.1.1 The completeness of formal languages

The completeness necessary for automatic reasoning using explicit reasoning mechanisms and formal representation languages can be illustrated with the following formal definition of the concept of "a brick" in a very limited, hypothetical world, containing only simple geometric objects such as bricks and pyramids Figure 12) (adapted from (Winston 1984)).

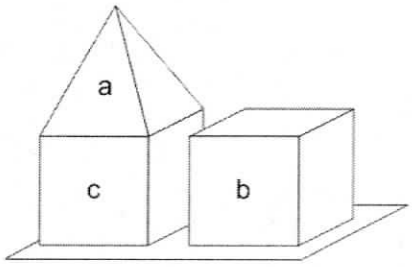


Figure 12. A blocks world example. In this particular example expressions such as: on(a,c), on(c, table), on(b,table), pyramid(a), brick(b), brick(c), ¬same-as(a,c), same-as(b,c), etc., are true

According to the definition, being a brick implies three things:

1. First, that the brick is on something that is not a pyramid;

2. Second, that there is nothing that the brick is on and that is on the brick as well; and

3. Third, that there is nothing that is not a brick and the same thing as the brick.

This definition could have the following predicate calculus representation.

$$\forall X\left(brick\left(X\right)\right)\rightarrow\begin{bmatrix}\left(\exists Y\left(on\left(X,Y\right)\wedge\neg pyramid\left(Y\right)\right)\right)\wedge\\\left(\neg\exists Y\left(on\left(X,Y\right)\wedge on\left(Y,X\right)\right)\right)\wedge\\\left(\neg\exists Y\left(\neg brick\left(Y\right)\wedge sameas\left(Y,X\right)\right)\right)\end{bmatrix} \quad (1)$$

The representation follows closely the natural language description and reads: "for all X, X being a brick implies three things:

1. There exists Y such that X is on Y and Y is not a pyramid,

2. There exists no Y such that X is on Y and Y is on X (at the same time)

3. There exists no Y such that Y is not a brick and Y is the same as X (at the same time).

The representation suggests that an intelligent agent which has no *implicit knowledge* of the hypothetical physical world and no capacity of generalization or analogy making, must be *explicitly* provided with *complete knowledge* in order to reason about "bricks" in that limited reality. The completeness of the representation manifests in the coverage of many of the facets of the definition including the rather awkward condition that "there is nothing that is not a brick and the same thing as the brick." The need for completeness is the main reason for which formal representations work only in very limited, artificial, primitive worlds and are rendered close to useless when applied to real, complex situations. As first mentioned in the previous chapter, the application of formal knowledge representations to real problems suffers from a fundamental shortcoming: the frame problem.

2.2.1.2 The frame problem

Given the capability of relatively effortless human relevance judgment, the frame problem seems a rather "artificial" creation, difficult to grasp and which usually goes unnoticed. In order to circumvent its abstract nature, Dennett uses a story-telling approach. However, the frame problem also applies to and could be illustrated from the perspective of humans, who in their first years of life, learn and can easily and efficiently reason about the side effects and the implicit changes of the complex four-dimensional spatio-temporal physical world in which they live. As this learning gradually becomes common sense knowledge, it causes us to efficiently determine the relevant implicit changes while ignoring the non-relevant ones for a given situation. For example, such trivial facts as that the clothes we are wearing are moving with us while walking or traveling are most of the times irrelevant given the context of a planned trip. However, if the trip involves some rapid movement through the air such as riding a motorbike, suddenly, wearing a sombrero becomes a relevant fact. As experts at managing our physical world, we are able, through an effortless but powerful mental simulation, to determine the relevance of such a particular fact. The recall of our personal experiences of moving fast through the air and of the dragging force of the air becomes paramount. Therefore, intelligent agents must be endowed with efficient mechanisms for determining the relevance of particular facts for a decision. In this dissertation, it is suggested that the analogue to the powerful human mental simulation is a memory-based approach, which in order to approach real-time capabilities, is based on a particular spatio-temporal complexity trade-off that increases the former in order to minimize the latter.

What seems to have made the robots vulnerable was their creators' choice for knowledge representation and reasoning: the robots did not have quick access to implicit knowledge about the relevance of particular facts (i.e., records of problem solving instances), but only to explicit facts stored in frames which had to be employed in time-consuming, immense number of explicit relevance judgments about the effects of particular actions. Although they were supposed to be experts at their task, the robots were behaving like novices. The frame problem is not a problem of the knowledge representation per se, but a problem of the choice for representation of knowledge needed to solve time-constrained

decisions. In other words, formal representations and logic reasoning work, but not in time constrained, complex, real-time situations.

2.2.1.3 The knowledge spectrum perspective

From the perspective of the knowledge spectrum, explicit, formal representations sit on the abstract side of the spectrum (Figure 13).

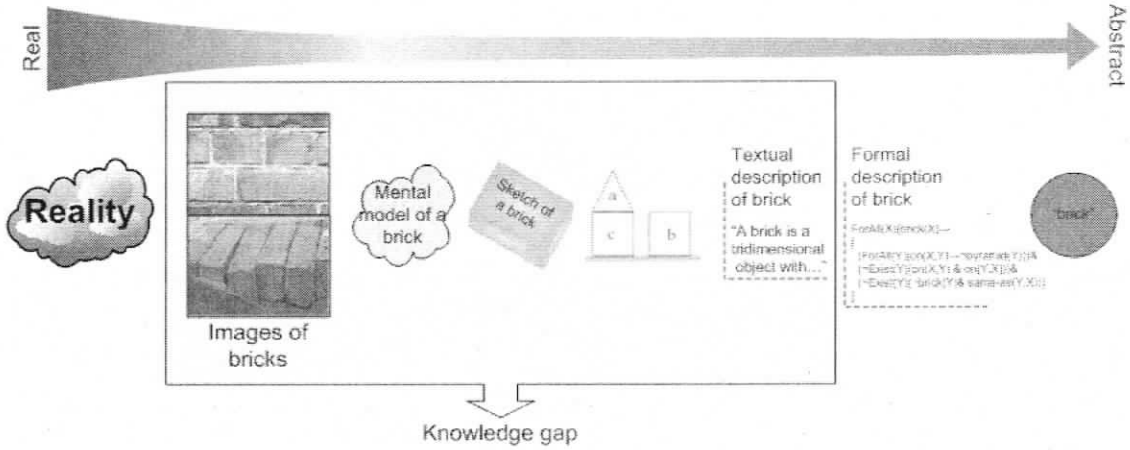


Figure 13. Representations of "brick" on the knowledge spectrum. Such representations range from rich (e.g., images, mental models) to less complex (sketches and diagrams) and to symbolic descriptions (textual, formal and conceptual).

The retrieval of explicit knowledge representation is currently the subject of the increasingly important research in information retrieval (IR). It is commonly accepted that IR is strongly coupled with the notion of intended meaning of concepts: a retrieved document is considered to be relevant to a query if the intended meanings of the authors of a document are relevant to the intended meaning of that query. It is evident that "meaning," a property that characterizes all concepts present in explicit knowledge, is intimately connected (if not identical) with the notion of context. According to this rather paradoxical view, meaning, a property that characterizes the abstract side of the knowledge spectrum, is strongly coupled with context which, by definition, is a feature of the reality side of the knowledge spectrum. Therefore, in order to construe meaning appropriately one needs to be able to efficiently move from abstractions towards richer representations of reality. This movement on the knowledge spectrum is necessary in

order to fill the knowledge gap between abstract concepts and the richer mental representations required for construing their meaning.

Explicit, formal representations attempt to capture general truth and generally applicable problem solving strategies, but become too abstract in nature. Through the abstraction process, which is essentially a reduction driven by the relevance judgments of knowledge creators, the context of a problem is lost. Losing context creates difficulties with construing meaning (which is context-dependent by definition) and widens the knowledge gap between the representation itself and the reality of a future problem-solving instance. The knowledge gap translates into the instantiation uncertainty that characterizes the application of general knowledge to specific problems (e.g., one may utter the word "brick," but what particular shape, dimension, kind, type, material, make, colour etc. do they mean exactly to use in the construction of a particular brick wall?). Making up for the knowledge gap through explicit relevance reasoning becomes time consuming and consequently takes its toll on the applicability of the representation. In sensitive applications such as medical decision-making and health research, general knowledge may potentially be harmful (e.g., prescribing a highly recommended drug to which a patient has an undocumented allergy). In addition, abstractions and general methods and theories of problem solving and decision-making (e.g., guidelines) do not fully enable the education of individuals and the learning from successes and mistakes. For example, knowing that an anonymous patient developed some allergy to an unknown drug is nearly useless compared to knowing that a specific patient, with a detailed health record which may include a genomic profile, developed a particular kind of allergic reaction, at a certain time, to a specific type or brand of a drug from a particular batch. The latter not only can help the research, but also could enable the prediction of future allergic reactions in that patient or in patients that exhibit similarities with that individual case.

2.2.1.4 Conclusions

Knowledge representation approaches must preserve to the extent possible, the context of a problem-solving instance. By efficiently recalling similar past instances of problem

solving and their contexts, intelligent agents are immediately provided with implicit knowledge about relevance, encoded in the retrieved contexts and, at the same time, with more possibilities to reduce the instantiation uncertainty of general knowledge when applied to specific problems. To enable this, informatics research must advance the representation and processing of rich, high dimensional modalities of knowledge encoded in past problem solving cases: this is the definition of CBR research. Finally, the demonstrated need for high dimensional representations organized in reusable case bases whose organization obeys similarity principles (i.e., CBR) is a strong argument for the validity of the thesis of this dissertation, in particular for the need to manage high dimensional associative concept representation spaces which possess a similarity-based organization.

2.2.2 Natural languages

Similar to formal specifications (e.g., predicate calculus), natural languages use abstractions, i.e., concepts. However, their richness and power of expression place them on the knowledge spectrum to the left side of formal specifications, but to the right side of rich descriptions consisting of images, sounds, video-clips and simulations of reality. Natural languages have power of expression but loose semantics and inherent ambiguity. However, despite their abstract nature, they remain the indispensable, main knowledge representation and transfer media between humans.

2.2.2.1 The ambiguity of natural languages

In order to illustrate the point about the ambiguity of natural languages, the reader is directed to the previous natural language definition of the concept of "a brick." Although the definition may look unequivocal, there are subtle ambiguities that make a difference in the predicate calculus representation. The first condition of an object to be "a brick" (i.e., "the brick is on something that is not a pyramid," highlighted in the definition 2 and 3) is an ambiguous natural language construction and could have slightly different formal representations:

$$\forall X\left(brick\left(X\right)\right)\rightarrow\begin{bmatrix}\left(\forall Y\left(on\left(X,Y\right)\rightarrow\neg pyramid\left(Y\right)\right)\right)\wedge\\\left(\neg\exists Y\left(on\left(X,Y\right)\wedge on\left(X,Y\right)\right)\right)\wedge\\\left(\neg\exists Y\left(\neg brick\left(Y\right)\wedge sameas\left(Y,X\right)\right)\right)\end{bmatrix} \tag{2}$$

$$\forall X\left(brick\left(X\right)\right)\rightarrow\begin{bmatrix}\left(\exists Y\left(on\left(X,Y\right)\wedge\neg pyramid\left(Y\right)\right)\right)\wedge\\\left(\neg\exists Y\left(on\left(X,Y\right)\wedge on\left(Y,X\right)\right)\right)\wedge\\\left(\neg\exists Y\left(\neg brick\left(Y\right)\wedge sameas\left(Y,X\right)\right)\right)\end{bmatrix} \tag{3}$$

In definition 2 this condition has been interpreted as: "the brick being on something IMPLIES that that something [sic] is not a pyramid" and was therefore represented as "for all Y, if X is on Y, this implies that Y is not a pyramid." In definition 3, which is identical to 1 but is repeated to the benefit of the reader, this condition was interpreted as "the brick MUST BE (or is always) on something that is not a pyramid" and that was represented as "there exists Y such as X is on Y and Y is not a pyramid." The first definition is therefore more "relaxed" as it allows the possibility that a brick sits on nothing. The second definition is more restrictive, because it requires the brick to be on something that is "not a pyramid" or otherwise X is not a brick anymore. Therefore, the first definition is more general and defines the concept of "a brick" in such a way that the definition would be true even in a world with no gravity (i.e., the brick is on nothing). In addition, the definition by equation 3 does not reject the possibility that an object sits on both another brick and a pyramid, at the same time (Figure 14).

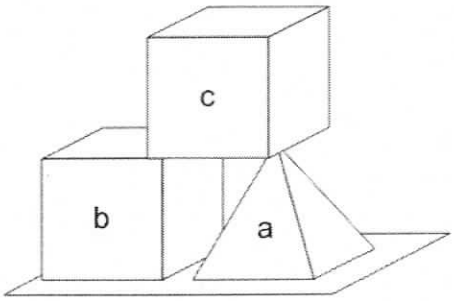


Figure 14. A blocks world example. In this particular example[7], brick(b), brick(c), pyramid(a), on(c,b), on(c,a) are true and therefore not rejected by the third definition: the condition that "c" MUST sit on something that is not a pyramid in order to be a brick is met by on(c,b).

The point is that, most often, humans receive and transmit knowledge without the deep understanding and completeness required by an exact mathematical representation of the knowledge to be transmitted. This shallowness has also been recognized by others (Sanford and Sturt 2002) who are trying to draw the attention to the fact that humans are rather superficial in their knowledge acquisition and processing and often make use of "under-specified" representations. Although, since the early days of science, scientists have fallen in love with the pure reasoning approaches, as they were reproducible, unambiguous means to express new knowledge, the problems with the use of classical predicate calculus as a knowledge representation method and of the classical logic inference as a reasoning strategy are discouraging. This is due to the requirements of complete, unequivocal representations, which prevents them from dealing with the messiness of the real world problems.

2.2.2.2 Conclusions

When possessing the necessary knowledge, humans are able to effortlessly fill the knowledge gaps between natural language representations and their richer, high dimensional representations of reality (i.e., mental models), and to easily construe the appropriate meaning of potentially ambiguous concepts. Although current technology allows for its storage, knowledge present in richer media (e.g., images, videos, simulations) is currently very difficult to process (e.g., real-time computer vision, scene understanding and synthesis, image understanding) using today's technology.

---

[7] Example indicated in a review of this work by Dr. Stefan Schulz

Because natural languages are used by people universally and allow rich representations that no other language specification can attain, natural language processing (NLP) research is a first step that Informatics should take in order to advance the organization and processing of individual context knowledge in case-bases that can be reused. This directly supports the NLP focused empirical work described in Chapter 4 of this dissertation. The insights gained by NLP research will advance knowledge processing towards richer knowledge representation media that will reduce the knowledge processing gap and consequently increase the knowledge processing capacity currently supported largely by human knowledge processors. Finally the thesis of this dissertation is fully supported by the demonstrated need for richer, high dimensional representations of knowledge.

2.3 CONTEXT-DEPENDENT INFORMATION PROCESSING

In this section, a series of arguments regarding how information might be represented in the human brain are presented in order to show the disconnection with the representation of information in computers. The fact that the pattern space in which human brain operates is high dimensional but immensely sparse is underlined: patterns that make sense to us are memorized and they typically represent an extremely small fraction of all possible patterns. Conceptual representations that occur in the human brain are also highly complex, dynamic and context sensitive and they are examined from the perspective of elementary concepts of algorithmic information theory using a classic example of a coin tossing experiment. The discussion leads to an extension of the minimum description length (MDL) principle and to an algorithmic definition of a pattern that sacrifices description length in order to attain a reduction in the complexity of retrieval time.

So far, the importance of context dependent representations and information processing for medical decision-making and information retrieval has been underlined. The following thought experiment will generalize the importance of context dependent representations and information processing to general concept spaces, lead to additional insights and provide a transition to the discussion of the four properties of the concept spaces.

2.3.1 A thought experiment

A simple thought experiment that involves the application of the same prototypical, abstract representation function to two information artefacts that are different in nature – an image and a text (e.g., a textbook) – will better illustrate the importance of context for information processing as proposed in this dissertation. In the case of an image, the representation function of choice is extremely common in information processing and performs what one would call a *zoom-out* operation which allows the visualization of various levels of abstraction in the case of images, from small scale to larger scale features. The inverse of this function would be equivalent to a *zoom-in* operation allowing one to focus on details. The zoom-in function is trivial only when such details

are available in the original object, otherwise its complexity increases in order to account for the additional information needed to make up those details.

In the case of a text, the same function is a *summarization* operation that also results in representations with various levels of abstraction (i.e., summaries) of the original text. The inverse of a summarization function aims at obtaining less abstract, richer representations from a reduced, abstract representation and would stand for what one would call *text generation* or *synthesis*. This is a function of increased complexity that often has to make use of additional information not explicitly available in the original text.

A plausible, context-independent zoom-out function implementation is one which eliminates every second pixel from a given image, thus providing various levels of abstraction of that image (e.g., Figure 3). An alternative way to look at this problem is that, for such a zoom-out operation, the identification of the pixels to be removed from the image is entirely dependent on their absolute position in the image, i.e., on their XY coordinates, but not on their colour content. This allows for an efficient context-free processing.

By the same token, a possible context-independent text summarization could be trivially implemented in a similar, context-independent way. However, eliminating every second character from a word or every second word from a text, or every second paragraph from a page or every second page from a textbook, while extremely efficient, will not result in a good summary of that text. Therefore, context-independent representation functions do not allow the appropriate visualization of various levels of abstraction of textual artefacts. This is due to the conceptual nature of textual data as well as to the fact that textual items (e.g., characters, words, phrases, sentences, paragraphs, etc) that should be part of a summary cannot be reliably identified through their absolute position in a given text. Such identification of key items can only be attained by more sophisticated context-dependent implementations of representation functions.

This simple thought experiment shows that simple, context-free representations are never appropriate for conceptual, textual information artefacts (Schank 1972) or for any other representations that make use of abstract, conceptual representations, as is the case with the user interfaces of computer systems in general (Blois 1984) (p36) and of medical information systems in particular (Moehr 1994). The amount of dependence on context seems to be the only difference between a zoom-out on an image and a text summarization. The experiment also does not dismiss a context-dependent processing of images which could form the basis of advanced image processing, image understanding and computer vision systems. At the same time, this confirms that context-dependent processing is a more general, more powerful, information processing framework situated at a higher level of the Chomsky hierarchy (Johnson-Laird 1993) than context-independent (i.e., context free) processing:

- regular grammars (finite state automata)
- context-free grammars (push-down automata)
- context-sensitive grammars (linear bounded automata)
- unrestricted transformational grammars (universal Turing machine)

2.3.1.1 Conclusions

The insights offered by the thought experiment allow a new formulation of the usability axiom introduced in previous chapter: medical information technology must attain human-like, adaptive, *context-dependent information processing* functions (i.e., knowledge processing) using existing, largely *context-independent information processing* models. This new vantage point allows us to speculate on a possible solution: human-like, adaptive, context-dependent information processing could theoretically be approximated with arbitrary precision by linking sufficiently many, simple, context-independent processing models into complex architectures. Context dependence essentially translates into high dimensionality and memory-intensiveness of representations. An important implication is that intelligent, context-dependent

information processing might be just a quantitative aspect related to the amount of memory needed in order to store the sufficiently many, general, context-independent functions whose operational whole could yield more advanced, adaptive, context-dependent processing.

These conclusions support directly the thesis of this dissertation through the demonstrated need for high dimensional, context dependent representations and thorough the important insight about the possible solution to this representation problem which, essentially, points towards a richly connected structure such as the associative memory model proposed in this dissertation.

2.3.2 Algorithmic Information Theory (AIT)

An additional indication for the importance of context-dependent information processing comes from the relatively new field of research called Algorithmic Information Theory (AIT) (Li and Vitányi 1997). AIT and the notion of Kolmogorov complexity unify the fields of computer science and information theory. It is said (Chaitin 1982) that rather than focusing on ensembles and probability distributions as in classical statistics and information theory (Shannon 1948), AIT focuses on the algorithmic properties of individual objects. Essentially AIT is a pattern processing (discovery and recognition) perspective on Information Theory in which the important notion of randomness hinges on the ability (or inability) of detecting patterns in data (Chaitin 1975). The AIT definition of algorithmic randomness also seems to converge with ideas regarding the importance of memory-based information processing, especially in information intensive domains where randomness is important (e.g., biomedical sciences) and/or where pattern spaces are sparse (e.g., natural language). While one can prove that certain data is not random by providing a highly compressed and lossless representation of that data, proving data to be random is impossible as it would be equivalent to "setting a lower bound on the complexity of that data" (Chaitin 1974). A consequence of randomness is that the attempt to represent random data (i.e., containing no regularities) in a model will require the memorization of the data as such (i.e., the representation achieves little or no

compression). In order to explain these statements better, an example adapted from (Li and Vitányi 1997) follows.

It is known that the classical probability framework generally aims at representing the reality of probabilistic experiments, the prototype of which is the typical coin tossing. In such experiments, individual tosses are generally assumed independent of each other. This is equivalent to assuming context-independence of events, in which previous tosses do not influence subsequent ones. For example an experiment involving ten consecutive tosses (T – tails, H – Heads) could results in outcomes such as HHHHHTTTTT, HTHTHTHTHT, TTTTTHHHHH, or THHTHTTHT all of which are assigned same probability $p=1/2^{10}$ under the classical statistics framework. In addition, because the ratio between the counts of heads and tails is exactly ½, probability theory also implies that the experiment indeed consisted of truly random tosses of a fair coin. However, this is exactly where our biased, experience-laden intuition departs from the objectiveness of the probability theory. Not only that our perception of the probability of such regular sequences as HHHHHTTTTT or HTHTHTHTHT does not appear to be identical to that of THHTHTTHT, but most of us are likely thinking that there is something weird about a supposedly fair coin that yields HTHTHTHTHT in 10 consecutive random tosses. The only result that seems to appear legitimate is THHTHTTHT, a particular sequence of T's and H's that appears random, or patternless. This kind of thought experiment forms the basis of the AIT framework laid down by A. Kolmogorov, G. Chaitin and R. Solomonoff (Li and Vitányi 1997).

What seems to stay at the crux of this difference between the objectiveness of the probability theory and our intuition is the fact that the number of sequences which exhibit regular structure (and hence low algorithmic complexity) is considerably smaller compared to those that appear random (later in this chapter we will actually see how much smaller). This causes the former to be extremely unlikely to occur in random experiments and causes most of us to perceive such regular structures as unlikely to be the outcome of a random experiment. This perception bias exists despite the fact that, objectively, the probabilities of regular and random sequences of same length are

identical and could very well serve as a possible explanation for such trivial matters as to why people buy 6/49 lottery tickets that seldom contain sequences such as 1, 2, 3, 4, 5, 6 for instance. This perception bias also seems to be continually reinforced by our reality which lacks discoveries of natural, lifeless objects that are perfectly regular (e.g., perfect square shape) and that are the result of random physical forces such as weather. However, more importantly, this same bias seems to suggest the fundamental mechanism that allows for high level cognitive functions in humans: experiential, context-dependent information processing which focuses only on regularity.

Because probabilities of events in classical statistics and information theory aim at objectivity and universal, context-independent applicability, they purposely disregard contextual information about the unfolding of experiments. For example, statistics focus only on the tally of heads and tails in the final state of the coin tossing experiment (i.e., after the 10 tosses). By contrast, if one chooses to account for the contextual information of an experiment, one is considered to be looking at the algorithmic properties of that experiment. Therefore, algorithms could be considered context-dependent spatio-temporal representation functions which describe events that unfold in a given reality.

The knowledge intensiveness of medical decision-making translates into the length of binary sequences and context-dependent representations are of high importance for efficient clinical reasoning. The potential to immediately recognize regular spatio-temporal patterns with low algorithmic complexity and which intuition tells are extremely unlikely to occur as the sole result of random processes, seems to be the mechanism that allows clinicians to decide whether certain sequences of events form causal relationships that represent disease or not. It is also fair to say that nearly every medical event is usually represented, if not explicitly in medical records, at least implicitly put in a spatio-temporal context by the patient and/or by the diagnostician during history taking. This equates to a more complex, algorithmic representation of medical events and aligns well with another information processing paradigm, case-based reasoning (CBR) (Pantazi, Arocha et al. 2004). This method of individual context knowledge processing that has originated in the dynamic memory models proposed by Schank (Schank and Abelson

1977; Schank 1982) and which aims at solving new problems based on the solutions to similar past problems, focuses specifically on context-dependent descriptions of problem solving cases. Therefore, by using algorithmic representations of spatio-temporal events CBR has the potential to attain more useful representations of our reality and to provide solutions to the information technology paradox.

2.3.2.1 An alternative to Minimum Description Length (MDL)

When it comes to representation of information using algorithmic approaches, one fundamental principle is undoubtedly the Minimum Description Length (MDL). In its earliest reincarnation, MDL is the principle known as *Occam's razor* which essentially states that the simplest explanation is usually the correct one. The concept of MDL was first proposed by Solomonoff (Solomonoff 1964) and formulated as such by Rissanen (Rissanen 1978) and later by Wallace and Freeman (Wallace and Freeman 1987) (a brief review is available in (Vitányi and Li 2000)). "The length of the shortest description" concept is fundamental to Algorithmic Information Theory and Kolmogorov complexity and makes perfect sense for computer representations and models where highly compressed representations of data are of importance. However, at the same time, the application of the MDL principle has the potential to result in representations that are readable only by machines and less readable by humans such as, for example, a zip compression of a text. What the idealized MDL principle seems to be missing is the simple fact that squeezing too much redundancy out of representations renders them difficult to understand by information processing devices such as the human brain, which are known to possess large amounts of long term storage and whose primary information processing goal seems to be the minimization of reading and processing time. For instance, in the context of cognitive models for language processing in general and that of text segmentation in particular, arguments against the idealized principle that the "best segmentation of the input is the one with the shortest representation" (i.e., MDL) seem to be compelling:

> *It is not clear, however, that infant minds — or even adult minds — work according to such idealized principles. For the human brain, where computation is slow and storage is plentiful, there seems no justification for a scheme which does a lot of work in order to save storage. To carry the argument a bit further, we can say that MDL and other compression*

> *schemes are an attempt to render natural language — which is "natural" to the human mind — more manageable by machines. The most efficient structure for a computer is the least redundant one, but this is not a characteristic of human language, which by its nature is highly redundant. (Batchelder 1998)*

The critique of the idealized MDL principle can be also restated as the unreasonable assumption that all computational models have a limited memory that needs to be saved to the extent possible through a compression of representations taken to the extreme. This critique is also seconded by the observation that the idealized MDL is often implemented through non-natural, biologically implausible algorithms which attain minimum description representations through computation-intensive search and evaluation of various representations from which the one with the shortest description is eventually chosen.

The alternative that could avoid this caveat of the MDL principle suggests itself and aims at a more flexible definition that could allow for biologically plausible, efficient processing of less compressed representations at the evident expense of additional memory. In this case, instead of being oriented towards the ideal of minimum description length, such a principle would advocate the more practical approach of minimizing the work to read and process the descriptions. Furthermore, calling this principle *minimum description work* (MDW) would allow an interesting analogy with the concept of work from mechanics - defined as the product between force and distance - in the sense that this permits a quantitative description of the processing work as the product between the length of a description (in bits) and the time required by an information processor to process it. Such a perspective would be able to cater for both:

- the situation when high compression is *useful* such as the case of a fast processor with limited, serial access memory, as well as for
- the situation when high compression is not *usable* such as the case of a slow processor which possesses plenty of memory which is parallel in nature and addressable by its content.

The definition of MDW is, to a certain extent, adaptable to the information-processing context because it has the potential to account for the relative strengths and weaknesses of a processor, manifested as a particular spatio-temporal complexity trade-off that could favour reducing one at the expense of the other. For example, MDW is adaptable to the case of biological information processing models, where the spatio-temporal complexity trade off seems to favour reducing time complexity at the expense space complexity.

Another side effect of this alternative view is the fact that current data compression benchmarks will need to be updated since the ideal of the absolute minimum length of description would be no longer the sole objective. Finally, the MDW perspective in information processing opens the door to a definition of the concept of "statistically rare but algorithmically significant patterns" (or regularities) where the lengths of descriptions are extremely important but their counts (i.e., statistics) may play only a secondary role.

2.3.2.2 An algorithmic definition of a pattern

The concept of pattern is associated with the concept of regularity or reoccurrence. Because of their repetition, regularities lead naturally to reductions in information content that enable intelligent agents to attain compressed representations, to recognize patterns, formulate predictions and reduce uncertainty of decisions. Conversely, if input were random, discovery of regularities, recognition of patterns and prediction would be difficult if not impossible and uncertainty of decisions would be high. But how often do regularities have to repeat in order to become significant patterns? The first insight that could lead to a possible answer to this question comes from information theory and consists of the well-known facts that the probability of any binary sequence of length n to occur in a random experiment is equal to $1/2^n$ and that the amount of information gained in the experiment is n bits (i.e., the base two logarithm of the inverse of this probability). It is now easy to see that the mere reoccurrence (i.e., at least twice) of a sufficiently long (e.g., 1024 bits) binary sequence in a random experiment is extraordinary and hence an extremely significant signal for the potential existence of other regularities in data that can explain it. The amount of information gained by the reoccurrence of such a

"monstrously rare" (Chaitin 1970)[8] event is therefore so significant (the longer the sequence the more significant) that the sequence becomes a regularity that must be remembered and recognized in the future. Because it has the potential to influence decision-making, and potentially a real-time, constrained, life-and-death decision (i.e., frame problem), the problem of representing this pattern must be approached in a way that minimizes the retrieval and processing time. Contrary to MDL principles, such an approach may have to sacrifice description length (i.e., use additional memory) in order to reduce retrieval time complexity.

2.3.2.3 Conclusions

In conclusion, this definition of significant regularities or patterns, while less useful for short descriptions, becomes very important for longer descriptions and warrants the development and evaluation of approaches that are able to discover, memorize (potentially slightly redundantly, in a MDW rather than MDL fashion) and recall efficiently regularities whose descriptions are as long as possible and hence, as significant as possible. Functionally, such approaches are associative memories able to discover and represent significant regularities in data. Structurally they are compositional representations of concept spaces where significant regularities have become features. The thesis and associative memory model proposed in this dissertation are in perfect agreement with these conclusions.

---

[8] The phrase is used actually for referring the probability for the occurrence of intelligent living creatures in a universe, suddenly and randomly, as opposed through a long, gradual evolution.

2.4 PROPERTIES OF ASSOCIATIVE CONCEPT SPACES

In this section, the four properties of concept spaces (high dimensionality, sparseness, dynamicity and similarity based organization) are discussed. High dimensionality and sparseness are closely interrelated and can be illustrated by a "proof by resource exhaustion" argument that sets a common sense upper bound on a class of objects that are represented in a high dimensional space. Dynamicity and similarity-based organization are discussed in the context of pattern recognition, of retrieval based on secondary keys and of the dynamic classification capacity of humans.

2.4.1 High dimensionality and sparseness

In order to illustrate the importance of high dimensionality and sparseness properties of concept spaces, a certain type of argument called "proof by resource exhaustion" is needed. By making use of existing, common sense bounds on existing temporal and/or spatial resources, through this line of reasoning one aims at setting upper bounds on these properties

Let us apply this principle to a simple thought experiment. Consider the 32 by 32, black and white bitmaps in Figure 15, Figure 16 and Figure 17. Each of these bitmaps, if uncompressed, requires at least 32 x 32=1024 bits of data in order to be stored. Although crucial to defining the structure of a bitmap, the additional few bits of metadata, i.e., the width and heights of the bitmaps, are ignored in this discussion.

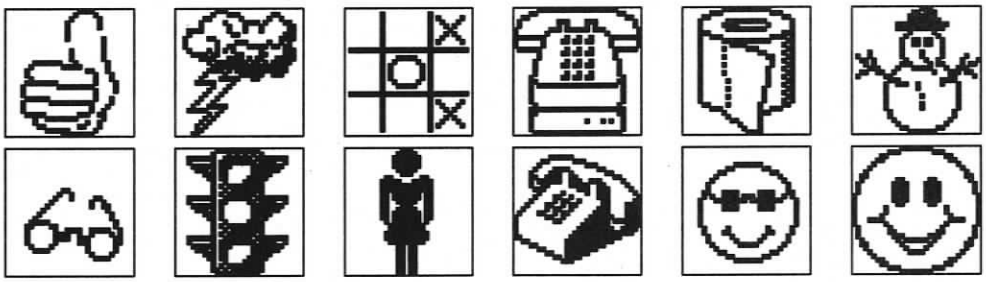


Figure 15. Bitmaps (32 by 32 pixels) that are meaningful to humans; there are very few bitmaps of this kind

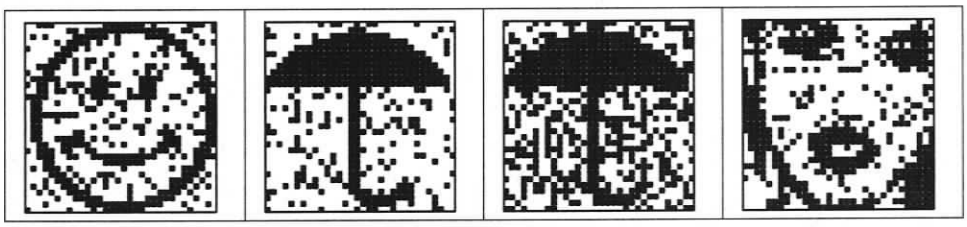


Figure 16. Bitmaps (32 by 32 pixels) that are noisy but still meaningful to humans; there are few bitmaps of this kind

Figure 17. Bitmaps (32 by 32 pixels) that are only noisy and not meaningful to humans; most of bitmaps are of this kind

2.4.1.1 Multidimensionality

Each of the bitmaps in Figure 15 possesses a certain meaning to us and could be regarded, most generally, as representations in a binary 1024-dimensional concept space. The bitmaps in Figure 16 have been created by superposing noise with meaningful images by an OR operation which affects only their white pixels. In this context, the following observations can be made:

- Computers are able to represent with extreme precision (i.e. without loss of any bit of data) all bitmaps in Figure 15, Figure 16 and Figure 17 regardless of their kind;

- Humans have high difficulties in representing precisely bitmaps such as in Figure 17 but can represent, however with various degrees of precision, the bitmaps in Figure 15 and Figure 16; the bitmaps in Figure 16 are represented with little precision (i.e., high data loss) because of the noise, which is difficult to be retained as is;

- The representation functions required in order to assess similarities between bitmaps in Figure 15, or between bitmaps in tables Figure 15 and Figure 16 are

most likely context-dependent and their implementation is not trivial on existing computers;

- It is a trivial task for any human to assess the similarities between any of the bitmaps in Figure 15 and Figure 16; however the bitmaps in Figure 17 appear random and very similar to humans, while extremely dissimilar to trivial, context-independent, distance-calculation computer algorithms.

To generalize on these observations, computers are able to represent precisely any of the 2^{1024} possible bitmaps of dimensions 32 by 32, while humans can only represent, with various degrees of precision, a fraction of those bitmaps, i.e., those that represent something meaningful to us. Estimating the number of bitmaps meaningful to us (in effect estimating an upper bound for it) will demonstrate an important property of the concept space: that of extreme sparseness.

2.4.1.2 Sparseness

Let us suppose human memory was so dynamic and adaptive that humans were able to learn (i.e., acquire and represent) a completely new, 32 by 32 black and white bitmap every millisecond of their whole life. A human life could be safely considered to span a 100 years (a century). Because a year has 52 weeks, a century will have:

$$1{,}000 \text{ msec/sec} \times 60 \text{ sec/min} \times 60 \text{ min/hour} \times 24 \text{ hours/day} \times 7 \text{ day/week} \times 52 \text{ week/year} \times 100 \text{ year/century} = 3{,}144{,}960{,}000{,}000 \text{ msec/century}$$
$$= 3.14 \times 10^{12} \text{ msec/century} \cong 2^{42} \text{ msec/century}$$

So, under these extreme assumptions, the upper bound of the number of potentially meaningful 32 by 32 black and white bitmaps is about 2^{42}. Therefore, the fraction of all 32 by 32 black and white bitmaps that could, under these extreme assumptions, ever be meaningful to humans is about $2^{42}/2^{1024}=1/2^{1024-42}=1/2^{982}$, that is, a number with more than 290 decimal zeros.

In order to put this number ($1/2^{982}$) in perspective let us consider the number of atoms on earth. If the earth were made only of hydrogen atoms (the lightest atom possible) the

following calculation will overestimate but at the same time provide a theoretical upper bound on the number of atoms on earth. The atomic mass of hydrogen is about 1, meaning that 6.02×10^{23} (Avogadro's number) atoms have a mass of about 1 gram. Therefore, a hydrogen atom weighs about $1/6.02\times10^{23}=0.166\times10^{-23}$ g/atom. The mass of the earth is estimated to be about 6×10^{27} grams (Giancoli 1980; Hewitt 1987). So the number of atoms in the earth can be no more than

$$\frac{6\times\ 10^{27}\ \text{g/earth}}{0.166\times10^{-23}\ \text{g/atom}} = 36.1\times10^{50}\ \text{atoms/earth} \cong$$
$$\cong 2^{5.18}\times\left(2^{3.32}\right)^{50} = 2^{5.18+3.32\times50} \cong 2^{171}\ \text{atoms/earth}$$

Finally, it is estimated that there are about 10^{81} (Honigwachs 2006) that is about 2^{269} atoms in the whole universe. So if we could label each atom in our universe with a distinct 32 by 32 black and white bitmap, then the number of universes needed to make use of all possible bitmaps would be the enormous number of $2^{1024}/2^{269} = 2^{755}$.

The argument has estimated an upper bound on the number of meaningful entities in a concept space of 32 by 32 black and white bitmaps and, at the same time, hinted at the infinitesimally minute ratio between this bound and the total number of items that could theoretically fill that space. Yet the argument, while extreme in the estimation of the learning capability of human brains (i.e., one bitmap every millisecond of a 100 year human life) is very conservative with regard to the dimensionality of the concept space. Many real world problems and applications involve much richer, much higher dimensional representations than rudimentary 32 by 32 black and white bitmaps. This furthers the point and demonstrates unequivocally the extreme sparseness of the concept spaces in which human information processing operates. In addition, the argument also sets a limit on the number of training examples in order to learn and develop advanced, context-dependent processing functions. But this is not to say that processing is limited to the training examples. In fact, because human associative processing is naturally trying to seek and converge towards the learned patterns, i.e., the ones that are repeated many times and develop a meaning to us even if superposed with noise such as in Figure 16,

processing can robustly generalize upon any possible input and is able to overcome much of the noise and randomness that are ubiquitous in our reality.

In general, such arguments "by resource exhaustion" are easy to construct and are often used in popular science publications. Mainstream literature and university course textbooks lack them probably due to their trivialized nature. However, the insights generated by them cannot be underestimated. A similar argument is presented in (Kanerva 1988)[9] in order to describe what Kanerva refers to as the "foremost problem" of human memory which is capable of easily operating in extremely high dimensional feature spaces unlike any known computer technology:

> *"There is no way to construct a random-access memory that has, say, 2^{1000} storage locations. Even 2^{100} locations would be too many to fit into the human brain, as 2^{100} molecules of water would more than fill it (the number of neurons in the nervous system is "only" about 2^{36}). With such a vast address space, most of the addresses cannot be represented by an address decoder and a storage location. However, there is hardly the need for 2^{1000} locations, because a human lifetime is too short to store anywhere near 2^{1000} independent entities (a century has fewer than 2^{32} seconds)."*

2.4.1.3 Conclusions

To conclude, the human brain can easily perform pattern matching in feature spaces the dimensionality of which can easily approach and even go beyond thousands of dimensions. We are able to discover and learn repeating environmental regularities (i.e., patterns) and retain them in a highly biased (or relevance based), context-dependent manner. This is in agreement with the thesis of this dissertation which states that the solution to representations problems can be approached by memory models which are able to manage associative concept representation spaces which are high dimensional and sparse.

Furthermore, at the expense of precision in representations, humans are also able to overcome randomness, to detect information inconsistencies and to master some of our complex reality by being able to predict outcomes of complex events, dynamically, in real time. Computers, on the other hand, are precise and powerful data storage and

---

[9] Page 53

communication devices, but their power is misdirected toward being able to represent *any conceivable piece of data,* regardless whether that data makes sense or not, whether it appears random or not, or whether it would ever represent any reality during the entire life of our universe estimated to be around $2^{3.58\times9} = 2^{32}$ years (about 12 billion years). This speaks again to the validity of a thesis that advocates the importance of managing associative concept spaces characterized by dynamicity and similarity based organization. It also supports memory models which aim at overcoming shortcomings of existing computational models of information processing by redirecting processing power towards the discovery, learning, representation and processing data in a context dependent manner.

2.4.2 Dynamicity and similarity based organization

The dynamicity and similarity based organization properties of associative concept spaces lead naturally to a discussion centered on a fundamental area of information processing, namely pattern recognition, a process which implies *dynamic* updating of representations in a memory which relies on a *similarity based organization* fundamental to any recognition mechanism.

2.4.2.1 Pattern recognition

Pattern recognition is an undisputed feature of human cognitive abilities and a research area in its own right. It is also the fundamental mechanism behind associative or similarity based retrieval which, in the context of computerized models, is referred to as *retrieval on secondary keys* (Knuth 1997). Though clearly useful, pattern recognition and similarity based retrieval do not seem to be as pervasive as they should in the information processing systems in current use. Hinting at the unsuitability of current computer architectures for retrieval on secondary keys tasks, Knuth (Knuth 1997)[10] acknowledges the possibility that the development of new models of computation could significantly improve this area of research and leave much of the past work on these issues obsolete. In

---

[10] Page 579

a similar vein, Kanerva offers a convincing discussion (Kanerva 1988)[11] around the "best match problem" which is essentially the problem of efficient similarity-based retrieval, first referred to as such by Minsky and Papert in their 1969 book "Perceptrons." Kanerva also concludes that "conventional computer architecture does not seem to be suited for dealing with the best match problem." At this point, it would not be difficult to conjecture that new alternatives to information processing models that aim at improving similarity-based retrieval (i.e., retrieval on secondary keys, best match) will most likely need to represent information using similarity-based, associative principles. However, as will be evident shortly, a further complication arises, as if the problem of similarity-based retrieval were not difficult enough: representational approaches must also be dynamic, meaning that they must support (though not as efficiently as the retrieval function) INSERT and DELETE functions which allow representations to be updated, dynamically, preferably in an online fashion.

2.4.2.2 The natural language perspective

Natural language, as a product of human cognition, offers compelling evidence that people are naturally inclined toward processing information using pattern recognition and associative similarity principles. This evidence consists of the widespread use of language devices such as the *simile* and the *metaphor*. These are examples of *comparison* and *analogy making* that humans perform without effort, in contrast to the difficulty of implementing them in artificial information processing systems (French 2002). Analogy making is essential to generating new knowledge and new artefact designs (Maher, Balachandran et al. 1995), as well as to problem solving and inductive reasoning (Keane 1988; Holyoak and Thagard 1995). In a case-based reasoning context, the essential tasks of case matching and retrieval also rely on pattern recognition, comparison and analogy making. In a decision making process, these mechanisms provide the immediate, implicit access to information about relevance, stored in the contexts of similar instances of problem solving. The patterns and analogies that humans are able to handle are often represented by complex spatio-temporal events with a potentially multi-sensorial impact.

---

[11] Page 49

For example, while humans have no difficulty in understanding a metaphor like "the computer swallowed the disk," an artificial information processing system that has no visual input sensors and which lacks the capability of image understanding, would probably never be able to perceive this particular analogy with the same speed, because of the extensive reasoning and amount of explicit knowledge needed to bring the swallowing process, as it occurs in living things, close to the action of inserting a disk into a computer's disk drive.

2.4.2.3 Dynamicity

In addition to operating on high dimensional, complex spatio-temporal patterns, analogy making in humans also possesses a dynamic component that could yield different relevance judgment outcomes, depending on context. A very illustrative example is given by French and Labiouse in (French and Labiouse 2002), using the concept of a "claw hammer." According to its designed purpose, the "claw hammer" is semantically close to other concepts like "nail," "hit" and "pound." However, it may be dynamically "relocated" in the semantic space (or reclassified), through effortless mental simulation and analogy-making processes, to the dynamically created class of "back-scratching devices," in the semantic neighbourhood of the "itch," "scratch" and "claw" concepts. Similarly, one could think about the concept of a wooden decoy duck, which inherits properties from at least the "wooden object", "animal duck", "toy" and "hunting gear" classes. This concept may also be dynamically relocated into the semantic neighbourhood and associated to any of the classes, depending on the context of use that may be focused on themes such as "combustibles" or "hunting" for example. In the medical domain, the contextual dependence of relevance judgments, classifications and analogies is even more important, as these are often based on uncertain information and may be dynamically re-evaluated in the light of new information about the patients or about their diseases.

2.4.2.4 Conclusions

From a similarity-based organization perspective, humans are naturally equipped with powerful pattern matching and dynamic classification capabilities which allow them to cope with complex, time-constrained relevance judgments, to easily construe meaning of

concepts and to tolerate the ambiguity of natural language. Currently, mainstream computer technology is limited when it comes to discovering and learning complex environmental regularities and predicting outcomes of complex events. To our frustration, computers are largely unable to process information contextually (e.g., distinguish between "form" and "from" in spell checking), are very sensitive to noise (e.g., misspellings), and are unable to judge the consistency of information in order to take on some of the human knowledge acquisition and processing burden.

Only relatively recently have computers come close to such functionalities with the introduction of data mining and machine learning techniques such as self organizing maps and clustering algorithms based on similarity metrics (Kohonen 2001). In machine learning approaches, the important problem of *feature discovery and selection* equates to a problem of relevance and forms the basis of useful representational approaches (e.g., rewrite systems) which, when inverted, could be successfully used for retrieval on secondary keys using inverted indexes.

The demonstrated dynamicity and similarity based organization of human representations of knowledge fully support the thesis as well as the memory model proposed in this dissertation.

2.5 MEMORY-BASED PROCESSING (I.E., TRADING SPACE FOR TIME)

In this section the importance of memory in information processing is underlined from the perspective of language processing. More precisely, it is proposed that, in order to attain advanced information processing capabilities, the trade-off between space complexity (i.e., memory) and time complexity (i.e., speed) must favor the latter at the expense of the former.

Context dependent information processing is accomplished by knowledge processors, natural or artificial entities able to create abstractions from data and to instantiate abstractions in order to fit reality. Regardless of their nature, two important features of such processors are their memory and their processing mechanisms.

2.5.1 Importance of memory

It is commonly accepted that storage and manipulation of information are necessary for complex cognitive activities in humans (Baddeley 2003). Memory is also considered crucial for both the "situation recognition" and mental modeling processes that are part of naturalistic decision models (Klein 1999). From a computational point of view, one could easily argue that without a random access memory structure there can be no effective processing. In the context of "the computational architecture of creativity," this argument is clearly outlined in (Johnson-Laird 1993). It is based on the examination of the classes of computational devices, in the ascending order of their computational power, ranging from finite-state machines to pushdown automata to linear bounded automata and Universal Turing Machines. These are paralleled by their corresponding grammars, arranged similarly in the Chomsky hierarchy, consisting of regular grammars, context-free grammars, context-sensitive grammars and of the unrestricted transformational grammars for machines with random access memory (Johnson-Laird 1993). Therefore, an important common aspect of human and computer information processing is their dependence on memory and, from a grammar point of view, their degree of context-dependence.

2.5.2 The natural language perspective

Recent natural language processing (NLP) research stresses the importance of memorization of individual natural language examples (Bosch and Daelemans 1998). The importance of memory is also emphasized in earlier (Riesbeck and Kolodner 1986) and more recent models of language processing in humans (Saffran 2000; Murdock, Smith et al. 2001). These converge on the idea that natural language processing, regardless of the processor, is memory-based. Additional evidence comes from the fact that most language constructs (e.g. words, phrases, sentences, etc.) have very low frequencies and their pattern space is high dimensional but very sparse. In fact, the very low frequency of most words in the English language (i.e., Zipf's law) is known from the 1940s since Zipf's famous book "Human Behavior and the Principle of Least Effort" (Zipf 1949) which is discussed in (Manning and Schütze 1999). The main implication of "Zipf's law" is that purely statistical approaches or language processing algorithms that do not memorize training examples will either lose important information or may need extensive data (potentially impossible to collect) in order to be able to retain important features which have extremely low frequencies (Daelemans 1998) and which may be crucial for construing the appropriate meanings of language concepts.

2.5.3 Conclusions

To conclude, the trade-off between learning effort and communication efficiency seems to be biased naturally towards memorization which implies a high space complexity rather than towards logical reasoning which implies a high temporal complexity. This is in striking contrast to the situation of the robots who did not have quick access to implicit knowledge about the relevance of particular facts (i.e., memory of problem solving instances), but only to explicit facts stored in frames which had to be employed in high temporal complexity (i.e., time-consuming), immense number of explicit relevance judgments about the effects of particular actions. By the same token, the advanced knowledge processing in humans might not be the result of very sophisticated reasoning strategies, but rather the utilization of a limited reasoning apparatus on a huge knowledge base, consisting of rich representations of one's experience. The limitations in reasoning

seem to be balanced by content-addressable memory and complex spatio-temporal pattern recognition capabilities operating on a case base (which includes common-sense knowledge) built from years of experience. From a computational complexity point of view, the processing complexity of natural language might therefore not be related to the sophistication of the algorithms, but to the memorization capabilities of the language processor. This trade-off can be likened to that of an algorithm, which, by making use of redundant data structures, is able to achieve shorter computation times at the expense of additional memory. This essentially means that the algorithm (or its designer) is "trading space for time."

In the context of a memory model one could also look at the space-time complexity trade-off from the point of view of two classes of functions that such a model must implement: the *update functions* (i.e., write) and the *retrieval functions* (i.e., read). The "space for time" complexity trade-off could manifest itself in the model proposed in this dissertation in the form of the ability to achieve efficient retrieval at the expense of less efficient updates while assuming that the retrieval needs are greater than the update needs. This functionality is based on redundant data structures (i.e., indexes) that can employed in efficient retrieval but which would have to be changed appropriately and potentially more slowly, after every memory update. Though in biological memories the two classes of functions are not as neatly separated (e.g., there is no such thing as a pure read-only retrieval from human memory) the fact that learning new information (i.e., memory update) is usually slower than quickly remembering already learned information (i.e., retrieval), seems to support the validity of such a mechanism.

In earlier sections, it has been argued that advancing towards context-dependent representations could be recast as a problem of devising efficient approaches and models for managing concept spaces characterized by *high dimensionality*, *extreme sparseness*, *dynamicity* and a *similarity-based organization*. The memory needs of an information-processing model can be easily connected with the high-dimensionality of context-dependent approaches. It has also been argued that while humans have natural support for the management of concept spaces and for the case-based reasoning paradigm, through

the memory of past experiences of problem-solving and powerful case matching mechanisms, technical solutions using current technology are challenging. The logical conclusion is that, from a case-based reasoning perspective, humans seem to be naturally endowed with the necessary memory structures for efficient case base acquisition, organization and retrieval while computers do not directly support this way of processing information and knowledge. From an evolutionary point of view, what this means, essentially, is that trading space in order to be able to attain the efficient, real-time management of concept spaces, seems to be the nature's solution to endow us with the ability to cope with our complex world. From an expert-novice perspective, this "space for time" complexity trade-off seems to be what makes the experts. This particular insight is in perfect agreement with a thesis that advocates the appropriateness of managing associative concept spaces which are high dimensional, sparse and dynamic through memory models that specifically address these properties.

2.6 PRINCIPLES FOR ASSOCIATIVE CONCEPT SPACE REPRESENTATION

The principles introduced in this section parallel the four properties of the concept spaces. High dimensionality is approached by hierarchical and distributed models. Sparseness and dynamicity lead to the use of hash functions, pointers, and linked lists while avoiding the use of arrays. Organization by similarity is approached through the use of trie memory models while avoiding the use of hash functions. The synthesis of all design principles leads naturally to the Deterministic Dynamic Associative Memory (DDAM) model proposed in this dissertation.

2.6.1 High dimensionality: hierarchical approaches

The importance of compositional and hierarchical approaches to both artificial and biological information processing is indisputable. Their universality transcends academic boundaries. In the context of the biological significance of self-organizing maps, Kohonen states it unequivocally (Kohonen 2001): "human brain is undoubtedly hierarchical." The most important reason for using hierarchical and compositional approaches most likely lies in the inherent *information compression* capability of hierarchical models. Compositional hierarchies are also often described in cognitive science, psychology and memory research literature, though often not explicitly as a method to overcome multidimensionality. One such example is the generic concept of "chunking," a form of grammar induction that aims at the "meaningful packaging of information" (Bourtchouladze 2002)[12] and which usually shares discourses with concepts such as *memory span*, *trace decay* and the *magic number seven* (i.e., the average number of chunks that our short term memory can remember) (Miller 1956).

Another area of research that advocates explicitly the extreme importance of hierarchical and compositional representations is Geographical Information Science (GIS) a relatively new field of research sharing many similarities with Medical Informatics. In particular, the fact that "hierarchies are one of the most common forms of organizing and structuring complex systems where a system is subdivided in smaller subsystems, and further

[12] See Chapter 2 of this work.

subdivision of subsystems can be recursively repeated as long as the subdivision makes sense" (Koestler 1967) resonates very well with GIS where hierarchical spatio-temporal reasoning (Timpf and Frank 1997) appears to be a common research topic. Though not without additional complications, insights from non-biomedical sciences such as GIS could be successfully extrapolated to biomedical contexts. The possibility of hierarchical spatio-temporal reasoning, though limited by the high complexity and variability of human anatomy and physiology, may be of particular interest to Medical Informatics.

2.6.1.1 Hierarchical representations in language processing

Yet the most compelling examples of compositional and hierarchical models and representations seem to be those that come from the realm of language processing. Structurally, hierarchies pervade all structural aspects of natural language discourse where artefacts are commonly organized in volumes, individual books, chapters in books, sections in chapters, subsections, paragraphs, sentences, phrases, words and punctuation, morphemes, syllables, characters, phonemes, all part of different layers of the same hierarchy.

In the specific contexts of medical natural language processing and medical terminology development, notions such as hierarchy and compositionality are indispensable. Their application range from morphosemantic decomposition of compound medical terms (Lovis, Baud et al. 1997; Baud, Rassinoux et al. 1999; Rassinoux, Ruch et al. 2000; Schulz and Hahn 2000; Hahn, Honeck et al. 2001; Schultz, Honeck et al. 2002) to the creation of compositional medical terminology schemes (e.g. GALEN (Rector, Rossi et al. 1998) and SNOMED (Spackman, Campbell et al. 1997)) and to the compositional organization of anatomical concepts into hierarchical frameworks (Cerveri, Masseroli et al. 2000). However, the complexity of the biomedical domain has lead to an important distinction between the types of hierarchical approaches which, from this perspective, can be:

- simple classifications (e.g., controlled terminology systems such as International Classification of Diseases ICD9)

- complex poly-hierarchies (e.g., the Medical Subject Headings or MESH hierarchy) and multiple inheritance systems (e.g., compositional controlled terminologies such as the International Classification of Diseases ICD10, SNOMED (Spackman, Campbell et al. 1997), GALEN (Rector, Rossi et al. 1998)),

- Information Retrieval (IR) systems based on inverted indexes.

The latter, though rarely referred to as hierarchies because of their dynamic nature, have the capacity to dynamically classify a document in multiple categories based on its content and similarity to a query (i.e., the category). This behaviour makes them functionally equivalent to a poly-hierarchy whose dynamically created categories are the result sets of a query. Though significantly more difficult to build and maintain using current technology, poly-hierarchies are a well-recognized desideratum of medical terminology systems (Cimino 1998).

Also on a functional level, hierarchical and compositional structures manifest in the syntax and semantics of human languages. Hierarchical and compositional approaches also seem to account for the infinite productivity of human languages in spite of the poverty-of- stimulus that characterises language learning. Their significant contribution to dimensionality reduction and to the generative properties of languages is illustrated by the following simplified grammar induction example.

2.6.1.2 Grammar induction as dimensionality reduction

The sequence *abcdefcdefababefcdefabcdcdefabcdefefefabcdef* is an artificially created sequence that contains certain patterns (or chunks) which cause it to exhibit both some structure (the repeating chunks themselves) as well as a degree of randomness (the apparent lack of order in the chunk order). More formally, this sequence is constructed of 44 characters drawn from the alphabet $\{a,b,c,d,e,f\}$ such that the characters form three chunks, *ab*, *cd*, *ef*, each of two characters. The fact that the chunks repeat is the basis to creating a more meaningful, potentially compressed packaging of this particular string. The specification of how to build this particular string as well as others that are *similar* to

it, is equivalent to specifying a "grammar" that generates such strings. This particular form of *grammar induction* (Manning and Schütze 1999)[13] aims at the discovery of chunks, in an unsupervised manner. This task is therefore a necessary step towards achieving, more meaningful, useful, compressed representations. Most importantly, in doing this, we change the properties of the representation space in which the sequence and other sequences similar to it are to be represented. Initially, before inducing any of the sequence structure (i.e., chunks, patterns, regularities), one can only look at it from the most general perspective. This most general perspective consists of considering the sequence as a point in a 44-dimensional representation space, together with the rest of $6^{44}-1$ possible strings of 44 characters drawn from the 6 character alphabet $\{a,b,c,d,e,f\}$. Besides our particular sequence, this 44-dimensional space includes sequences such as those that contain substrings such as *af*, *be*, *afebd*, *bacffb*, *cbfcea*, *aaaccccaa*, etc. and which, excepting the characters themselves, seem to possess no other structural similarities with our particular string. However, aside from the fact that they could be considered noise with respect to our string, these structurally dissimilar objects are also extremely many compared to those that are similar to our particular sequence. This describes well a representation space which is high dimensional but extremely sparse.

By inducing some of the structure of the sequence (i.e., the chunks *ab*, *cd*, *ef* as well as other, possibly longer regularities), one would be able to shrink the representation space at least

$$\frac{100\times 3^{22}}{6^{44}}=\frac{100\times 31,381,059,609}{1.732\times 10^{34}}=1.811^{-22}\%$$

of the original one. This is the same as projecting the rewritten sequence in the terms of its discovered regularities onto a new, lower dimensional subspace where the discovered regularities are features.

---

[13] Chapter 12.

For example, by rewriting the sequence using the machine-induced grammar in Table 8 as a composition of the 6 distinct elements {*abcdef*, *cdefab*, *ab*, *efcdefab*, *cdef*, *efef*} which are themselves recursive compositions of smaller patterns, one has achieved the hierarchical representation of the sequence and a projection onto a subspace which is only 9-dimensional.

Rule	Expansion
3 → a b	a | b
5 → c d	c | d
6 → e f	e | f
4 → 5 6	cd | ef
2 → 3 4	ab | cdef
7 → 4 3	cdef | ab
8 → 6 7	ef | cdefab
9 → 6 6	ef | ef
1 → (2 7 3 8 5 7 4 9 2)	(| abcdef | cdefab | ab | efcdefab | cd | cdefab | cdef | efef | abcdef |)

Table 8. A plausible grammar for the sequence *abcdefcdefababefcdefabcdcdefabcdefefefabcdef* which projects the original sequence from a 44 dimensional space onto a 9 dimensional feature subspace containing 6 possible features {abcdef, cdefab, ab, efcdefab, cdef, efef}

An additional reason for the usefulness of grammar induction, besides that of dimensionality reduction, will become apparent in the context of dynamic, multiple classification approaches.

2.6.1.3 Conclusions

To conclude, the fundamental property of high dimensionality of associative concept spaces can be approached naturally by hierarchical representation models which are compositional in nature. Hierarchical approaches to representation are universally accepted and transcend boundaries of academic disciplines. The structure of natural languages and of natural language artefact is also eminently hierarchical. In addition, it has also been demonstrated how the process of grammar induction, a fundamental capability of most information processors, can be regarded as a compositional approach to dimensionality reduction. The fact that one of the fundamental structural properties of the memory model proposed in this dissertation is its hierarchical nature is a good argument for its validity.

2.6.2 Sparseness and dynamicity: hash functions, linked lists, not arrays

A brief review of the concept of sparseness in a literature collection of a few thousands articles has revealed that by and large, sparseness is regarded as a problem, often referred to as the "sparse data problem" which needs to be "handled, reduced, prevented, coped with, addressed, fought, overcome, etc." The one and only exception seems to be the case of "sparse distributed codes" whose positive connotation arises from their ability to actually enhance the storage capacity of neural networks (Foldiak 1990; Rolls and Treves 1990). Similar accounts on the importance of sparseness (Barlow 1989; Field 1994; Olshausen and Field 1996) come from studies in vision and sensory coding and fall under the general term of "Minimum Entropy Encoding" which advocates the importance of sparse coding "in which each feature detector is activated as rarely as possible" (Bell and Sejnowski 1997).

The sparseness of representations allows our brains to operate in high dimensional conceptual spaces with thousands of dimensions. Sparseness is also the property that allows the proposed associative memory model to work in the way envisioned in this dissertation. However, since the model has also been realized in a physical device (i.e., a computer) that makes use of a technology not particularly suited to operate in extremely sparse spaces, design principles that address the efficiency of dealing with sparseness of representations are still needed.

2.6.2.1 Hash functions

Hash functions are mathematical objects which possess the extremely useful quality of being able to map extremely large but sparsely populated pattern spaces onto compact ranges that can fit into the address space of current computer memories (e.g., a hash table). A good overview of hash functions is available in (Knuth 1997). Subsequently, the memory locations can be probed for the existence of a pattern (or of a pointer to one) with an extreme efficiency which remains virtually independent (i.e., constant time complexity) of the number of stored patterns. At first glance, this makes hash functions an ideal choice for the implementations of content addressable memories such as search

and retrieval systems, content addressable memories, and other applications, which require references to information by content rather than by address.

2.6.2.2 Sets

In computers, sets are most commonly represented by arrays as chunks of memory locations whose address space forms a contiguous range. This makes it possible to liken the array to a real world container (e.g., a box) which also delimitates a contiguous spatial region whose dimensions are specified in advance. A container is supposed to accommodate all the instances of objects for which it was designed. Therefore, a memory array, as any real world container, can accommodate various objects, as long as their sizes do not exceed the specified dimensions. In a sense, storing in an array is a context-free operation, oblivious to the shape, composition and structure of stored objects and would be most efficient in the case of objects whose structure and dimensions fit the contained perfectly (e.g., rectangular blocks) and compactly. However, when such objects are variable in dimensions and shapes (e.g., variable length strings in an array, or various fruit in a box) containers will only be sparsely populated and will contain significant unused space. Though not always an option in real world situations, the only viable alternative seems to be to recursively disassemble bigger, complex objects, in smaller, primitive parts and store the parts compactly, while at the same time maintaining precise instructions and *lists of pointers* to the different parts in order to allow for the dynamic, appropriate reassembly of the original objects. In a sense, storing in such a way becomes a hierarchical, context-dependent operation, which is fully dependent on the shape, composition and structure of stored objects. Besides the increased efficiency in the case of objects whose dimensions are highly variable and whose structure is eminently compositional, this representation (or storage) schema has an important additional advantage: it allows the similarity-based retrieval of the stored objects. This could be achieved by taking the precise assembly instructions and *lists of pointers* to different parts for each object and invert them so that smaller subparts point to bigger subparts which eventually point to the original objects made up of those subparts. Translated into a computational context, this is similar to taking the strings in a language (i.e., the objects), discovering the grammar rules of the language (i.e., grammar induction, pattern

discovery, feature set induction), rewriting strings as a recursive composition of substrings (i.e., the subparts) and finally inverting the rewrite rules into an inverted index which can allow the search and retrieval of the original strings, by their content (i.e., by the subparts).

2.6.2.3 Linked lists

The representations referred to in this dissertation as well as the proposed associative memory model could successfully be formulated using the concepts of elementary graph theory. The standard ways to represent graphs are as collections of *adjacency lists* or as *adjacency matrices* (Cormen, Leiserson et al. 2001). Because the model amounts to the creation of extremely sparse graphs, a second "line of attack" of the sparseness problem is the use of *linked lists* (or adjacency lists) as opposed to the use of arrays (or adjacency matrices) in representations of such graphs. This is so because, in case of extremely sparse representations, using an array whose dimensions are set in advance in accord to the estimated dimensionality of the largest item stored in them will result in a lot of unused memory locations. Though the concept of "dynamic array" exists and many modern programming languages offer implementations of it, the dynamic part is related only to the dimensionality of the array and does not take in account any of the content stored in that array. As with real world containers, the problem that occurs when using arrays is inherent in their fixed dimensionality which cannot be dynamically changed without significant overhead, and which cannot accommodate objects with variable dimensions without wasting memory.

2.6.2.4 Conclusions

To conclude, in the context existing computer technology, the whole discussion around sparseness and dynamicity of representation boils down to a simple, pragmatic question: how do we represent strings of various lengths in a computer? Currently, strings are represented as fixed or variable length, null terminated, 1-dimensional arrays. If the strings have a compositional structure, such as in the case of many non-random natural sequences, the alternative to represent them dynamically suggests itself and consists of linked lists which link the components that make sequences (characters, words, phrases,

paragraphs, etc.) explicitly. Using such an approach allows representation of strings of heterogeneous, unknown lengths limited only by the available memory, while at the same time allowing for extremely efficient update (INSERT, DELETE) functions. The fact that, in a linked list representation of a sequence, an element is used to retrieve the next element, is considered by Kanerva a good model of human memory (Kanerva 1988). The fact that linked list and pointers are fundamental building blocks of the associative memory model proposed in this dissertation speaks clearly to its validity. At the same time, the lack of alternatives tempts one to conjecture that the use of pointers and linked lists might just be the only way to appropriately represent sequences that form sparse dynamic spaces.

2.6.3 Similarity based organization: tries, not hash functions

The main goal of similarity-based (associative) organization of a memory model is to enable retrieval based on similarities with a query, a process that is key to human cognition. In addition, if the retrieval were deterministic, then data must be recalled exactly. Such a model is functionally a deterministic associative memory and, at the same time, a synthetic model with little correspondence in biological models which are known to lack determinism. Computationally, similarity based retrieval is what Minsky and Papert referred to as "the best match problem" and, as Kanerva described it (Kanerva 1988), it could be thought of as the retrieval of all binary descriptions stored in an associative (i.e., where similar descriptions are close), multidimensional and highly sparse binary space, within a certain bit radius of a query (also represented as a binary description in the same space).

2.6.3.1 Hash functions revisited

Due to their efficiency, hash functions are often the first choice for the implementation of content addressable memories. However, hash functions do not organize information by similarity and preclude the possibility of similarity based retrieval within a certain bit radius, in the manner envisioned by Kanerva. Because they do not preserve the existing structure of information, hash codes are highly artificial creations and "a poor model of

human memory" (Kanerva 1988). Fortunately, yet not as efficiently as with hash functions, similar results can be achieved with an associative memory model first proposed by E. Fredkin in 1960 and called the *trie memory* because of importance in information re*trie*val. Since then, the trie model has been improved in various ways by other researchers such as R. Rivest, R. de la Briandais and D. Morrison who proposed the *generalized trie*, the *linked list trie* and the *PATRICIA (Practical Algorithm To Retrieve Information Coded in Alphanumeric) trie*, respectively.

2.6.3.2 Trie memory models

A good analysis of trie models and algorithms is available in (Knuth 1997). The main advantages of tries over hash tables, adapted from a summary in (Ellard and Ellard 2003), are:

- Hash functions are usually chosen based on known properties of sequences, besides the evident sparseness; tries eliminate the need to design a good hash function because no other assumptions are made on the sequences;
- Tries preserve the implicit ordering of sequences, given by the ordering of the alphabet on which the sequences are built;
- Tries encode content information and allow for efficient similarity-based retrieval of all sequences with a given prefix or suffix;

In addition to their associative property, tries are *n*-ary tree data structures that support efficient FIND, INSERT and DELETE operations (Ellard and Ellard 2003)[14]. Most importantly, in the *n*-ary array implementation, the time complexity of these operations does not depend on the number, but on the length of the items stored. Although efficient for dynamic applications, the *n*-ary trie data structure is wasteful in cases where the data is sparse and the typical number of children of each node tends to be small such as in the case of many natural sequences. A naïve implementation of a *dictionary trie* as a 26-ary

[14] Page 27

tree that used fixed arrays to store children node pointers was estimated in (Ellard and Ellard 2003) to require about 35 times more memory than the entire dictionary stored as a simple text file. A space efficient improvement which causes an additional overhead for child node lookups but which addresses the sparseness problem is the linked list trie or the *de la Briandais tree*. The loss in efficiency in child node lookups is impacted only by the size of the alphabet on which the items stored in the trie are constructed, but the overall time complexity increase of operations on linked list tries remains little affected by the total number of sequences stored, especially for very sparse representations.

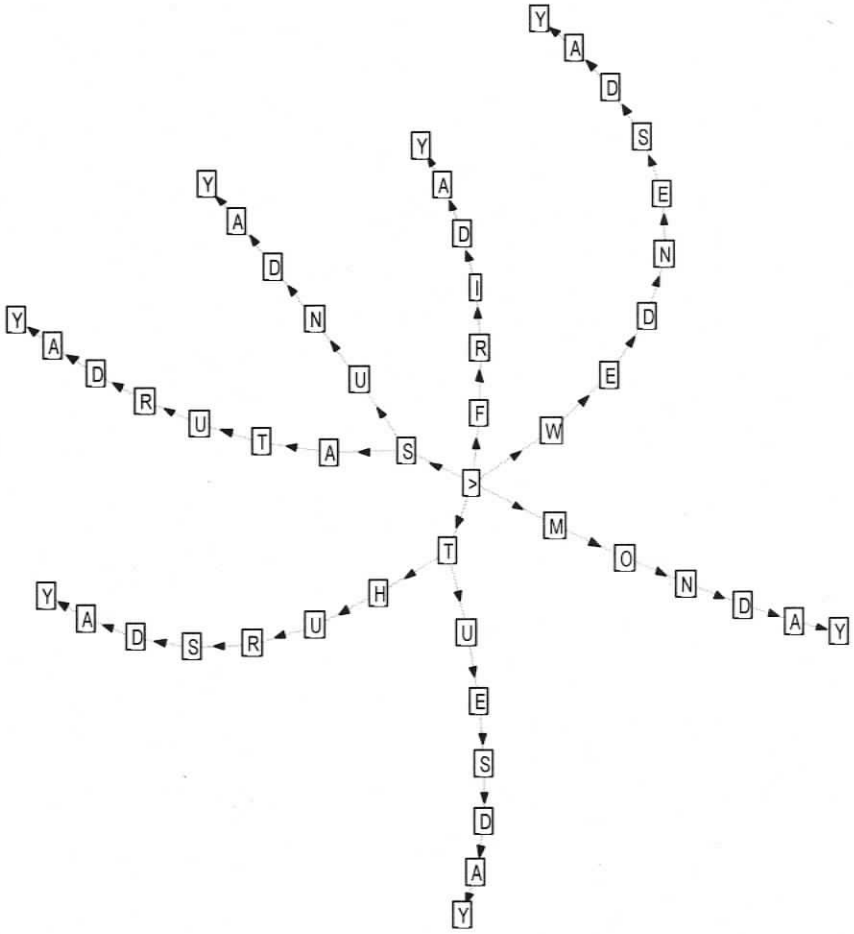


Figure 18. Suffix trie of weekday names

Given that the most common order relation in sets of strings is the left to right lexicographic order and that most writing systems are left to right, suffix tries such as the

weekdays trie in Figure 18 are the most common. This allows one to easily retrieve the items which are similar in their first characters.

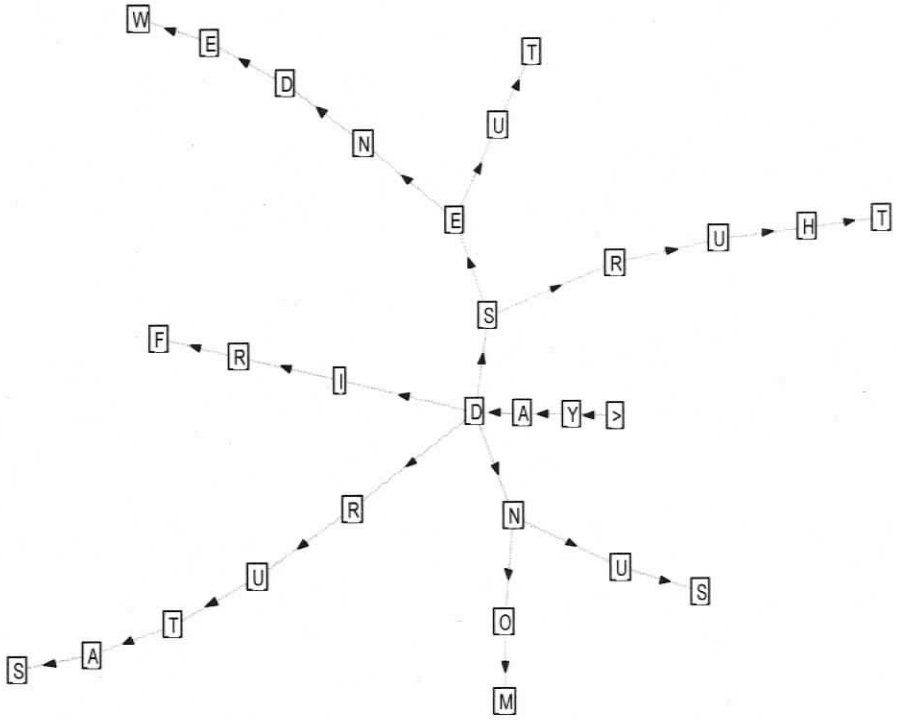


Figure 19. Prefix trie of weekday names

But the fact that suffix tries are of little use in the case of strings which are similar mostly in their endings (e.g., weekday names), suggests that, for certain applications, it may be useful to employ prefix tries such as the one in Figure 19.

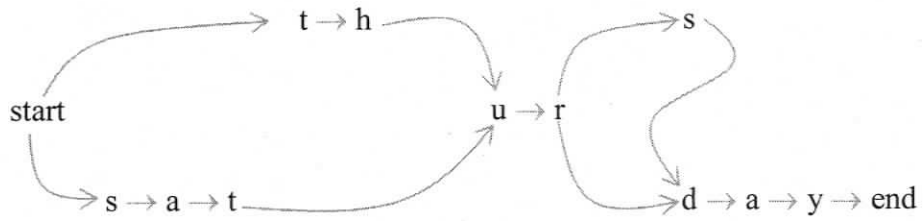


Figure 20. Diafix similarity of *thursday* and *saturday*

However, both suffix and prefix trie representations cannot reflect the fact that some sequences may also contain similar patterns in their middles (i.e., diafix similarity) such as, for example, the pattern *ur* in *thursday* and *saturday* (Figure 20). This fundamental

limitation in representational power stems from the "trie memory's being [just] a tree" (Kanerva 1988) and, in order to be overcome, would require representational approaches which are generalizations of trees, i.e. graphs such as the one in Figure 20.

However, these generalized approaches would have to be able to deal with the inherent problem of representation ambiguity which, in Figure 20, manifests itself in the representation of non-existing strings such as *thurday* or *satursday*.

2.6.3.3 Conclusions

To conclude *linked lists tries* are the data structure closest to the model proposed in this dissertation. Similar to them, the proposed model is dynamic, provides efficient access to stored sequences, is sensitive to the length of the alphabet on which the sequences are built, and it too remains virtually independent of the number of stored sequences. But due to fundamental representational limitations that reduce the capacity of similarity-based retrieval of tries, this is about as far as the similarity goes. The limitation of the representational power of trie models is overcome by the proposed associative memory model through the use of more general structures (e.g., partial order sets, directed graphs) that are able to resolve representation ambiguities through more complex, context-dependent representations.

2.6.4 Putting it all together: a deterministic dynamic associative memory model (DDAM) for associative concept space representation

The synthesis of the principles to successfully deal with the four properties of associative concepts spaces leads naturally to a memory model that is proposed in this dissertation as the deterministic dynamic associative memory (DDAM) model. From this perspective, the DDAM model is a hierarchical structure able to achieve dimensionality reduction and whose implementation makes extensive use of pointers and adjacency lists in order to overcome sparseness of data, while avoiding the use of arrays in order to attain dynamicity. The DDAM model also generalizes trie memory approaches and avoids the use of hash functions in order to attain a similarity-based organization.

2.6.4.1 The generalization of trie memory model

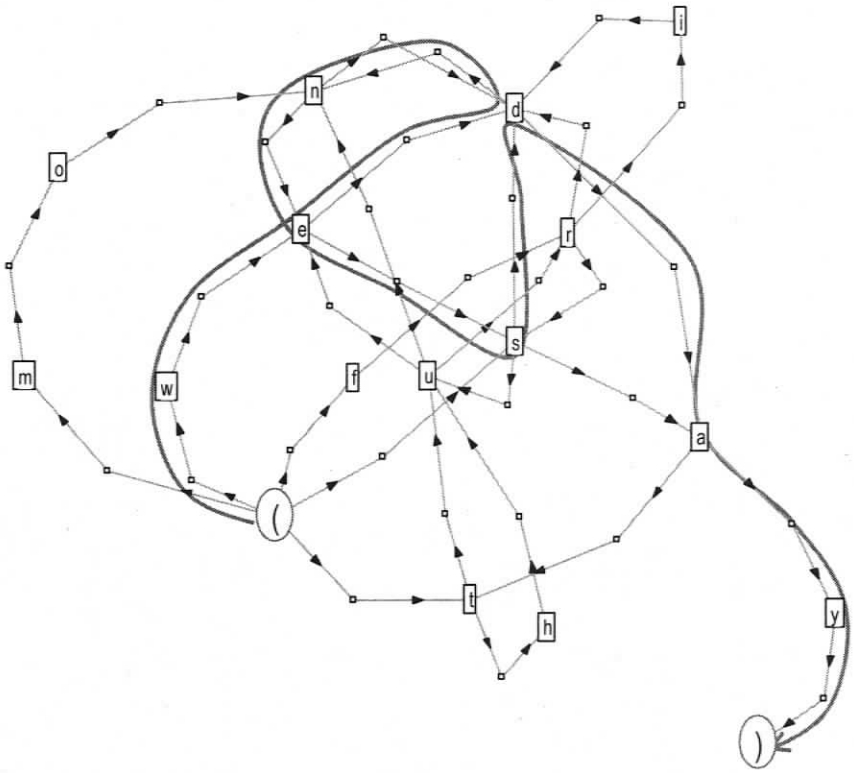


Figure 21. Trivial, highly ambiguous representation of the seven strings where single nodes stand for many instances of one character; for example, each of the nodes labelled *a* and *d* represents 8 instances (7 in *-day* and 1 in *sat-* and *wed-* respectively) of their respective characters; the 9.9 bit ambiguity representation of the string "(wednesday)" is traced by the thick path

The natural generalization of trie memory models consists of combining the prefix and suffix tries as well as the approach suggested in Figure 20, into a common data structure such as the one in Figure 21. This generalized structure ceases to be a tree and becomes a directed graph that will necessarily contain some additional nodes, intercalated with those associated with characters. As will be shown later in this chapter, these additional nodes are necessary in order to hold information about the transition probabilities of a node to another that follows it. The structure also contains two additional nodes, named start and end nodes, labelled in the figure by open and closed round parentheses, respectively. On closer inspection, this representation appears to be a particular case of *sequence alignment* in which all sequences are trivially aligned at character level. Though the approach eliminates the fundamental representational limitation of tries and allows, at the

same time, for retrieval based on suffix, prefix and diafix similarities, it also creates an additional problem due to its merging of multiple instances of a character (e.g., all *a*'s) into one, unique representational node. Depending on the numbers of instances, this causes representations to be highly ambiguous. The ambiguity manifests itself in the representation of sequences outside of the original set, such as, for example, "*(wesay)*", "*(frsay)*", "*(monesay)*", "*(sueday)*", "*(surday)*" as well as many others (Figure 21). A quantitative way to describe this ambiguity is by calculating it in bits, as the sum of the base two logarithm of the out-degree (number of links pointing outward) of all nodes (including the intercalated nodes) that make that representation. For example, the representation of the sequence *(wednesday)* in Figure 21 has an ambiguity of 9.9 bits (intercalated nodes denoted by λ, and $\log_2(0)=0$).

$$\log_2^{start}(5)+\log_2^{\lambda}(1)+\log_2^{w}(1)+\log_2^{\lambda}(1)+\log_2^{e}(2)+\log_2^{\lambda}(1)+$$
$$\log_2^{d}(2)+\log_2^{\lambda}(1)+\log_2^{n}(2)+\log_2^{\lambda}(1)+\log_2^{e}(2)+\log_2^{\lambda}(1)+$$
$$\log_2^{s}(3)+\log_2^{\lambda}(1)+\log_2^{d}(2)+\log_2^{\lambda}(1)+\log_2^{a}(2)+\log_2^{\lambda}(1)+$$
$$\log_2^{y}(1)+\log_2^{\lambda}(1)+\log_2^{end}(0)=$$
$$2.32+0+0+0+1+0+1+0+1+0+1+0+1.58+0+1+0+1+0+$$
$$0+0+0=3.9+6=9.9\ bit$$

Due to their simplicity, such trivial representations can only capture sequence similarities at character level. The only case when such representations are being able to represent longer, more significant patterns, is in the case when ***all*** sequences share the ***exact same similarity***, such as the suffix *–ay)* in the example. In addition, representations are least ambiguous only when each sequence is built on its very own specific subset of symbols from the alphabet. Nonetheless, trivial representations such as the one in Figure 21 are the starting point toward representations that possess the capability to represent sequences with least ambiguity, in a deterministic manner.

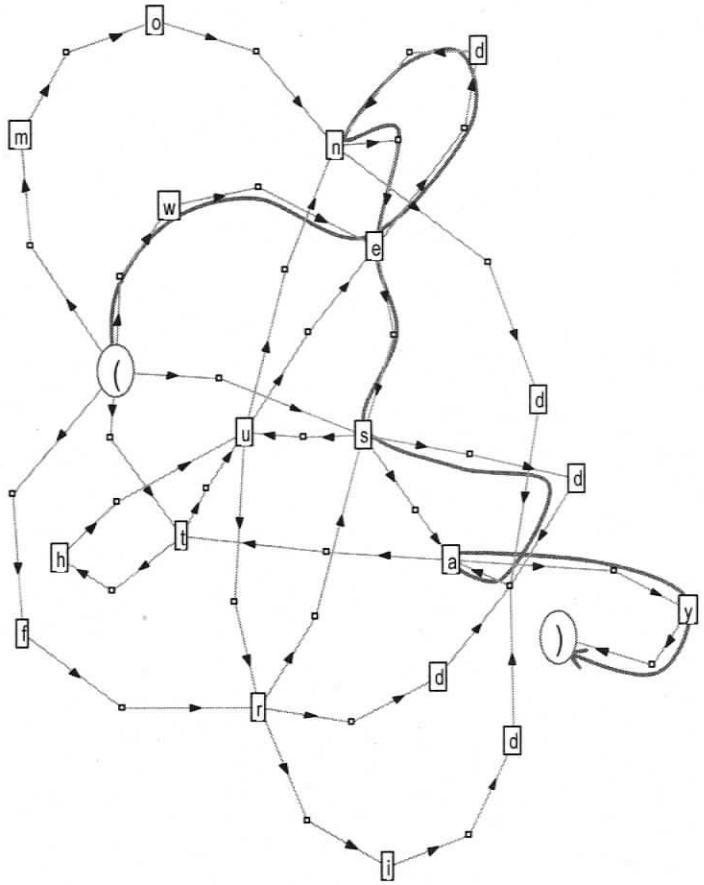


Figure 22. Less ambiguous representation of the seven strings, where single nodes still represent many instances of a character; however, there are now 5 nodes to represent the 8 instances (7 in *-day* and 1 *wed-*) of letter *d;* the 9.34 bit ambiguity representation of the string "(wednesday)" is traced by the thick path

The next step towards less ambiguous representations is shown in Figure 22. While sequences other than weekday names are still represented, the fact that now there are multiple instances of nodes for representing the letter *d*, restricts the number of sequences outside the original set. This also causes the representation ambiguity of the sequence *(wednesday)* to be reduced to 9.34 bits and prevents sequences such as *(sueday)* to be represented, while leaving sequences such as *(suednday)* or *(frsday)* still possible.

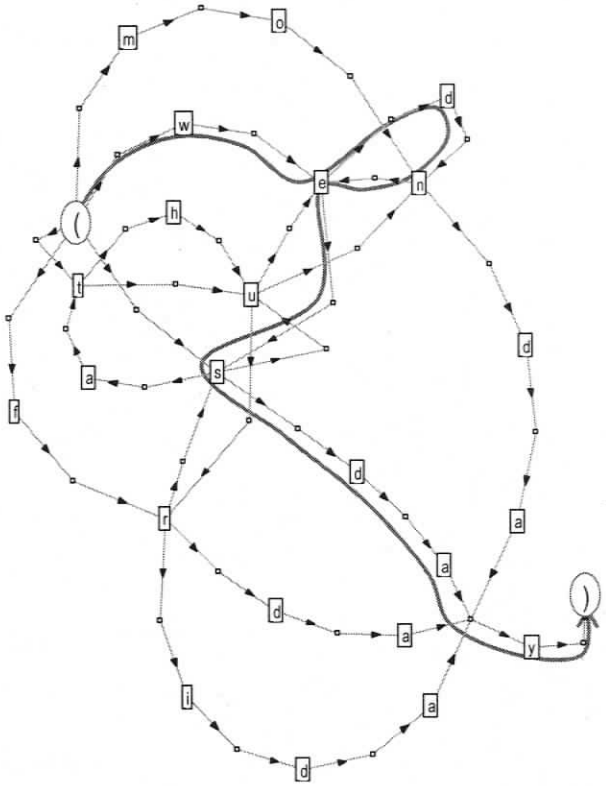


Figure 23. Less ambiguous representation of the seven strings; though there still are single nodes to represent many instances of characters (e.g., s in *sat-*, *sun-*, *thursday*), there are now 5 nodes to represent the 8 instances (7 in *-day* and 1 *sat-*) of letter *a*; the 8.34 bit ambiguity representation of the string "(wednesday)" is traced by the thick path

The next step towards less ambiguous representations shown in Figure 23 uses additional nodes to represent instances of the character *a*, and reduce the ambiguity of representation of *(wednesday)* to 8.34 bits.

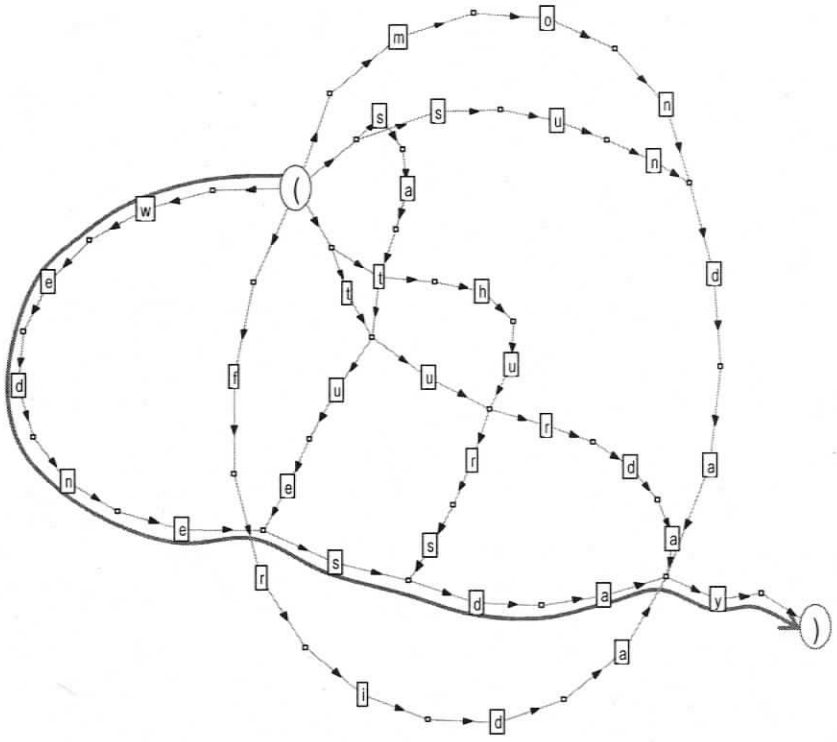


Figure 24. Less ambiguous representation of the seven strings; there are still ambiguous representations of strings such as (saturday); the 2.32 bit ambiguity representation of the string "(wednesday)" is traced by the thick path

By increasing the number of nodes to represent ambiguous characters such as *s* and *e*, the representation of *(wednesday)* in Figure 24 has attained the least ambiguity possible of 2.32 bits, or $\log_2(5)$, given by the 5 different choices of characters (i.e., *m*, *t*, *w*, *f*, and *s*) that can be used to start each one of the seven weekday names. However, several sequences outside the original set are still possible (e.g., *turday*, *satuesday*) but their number is drastically restricted.

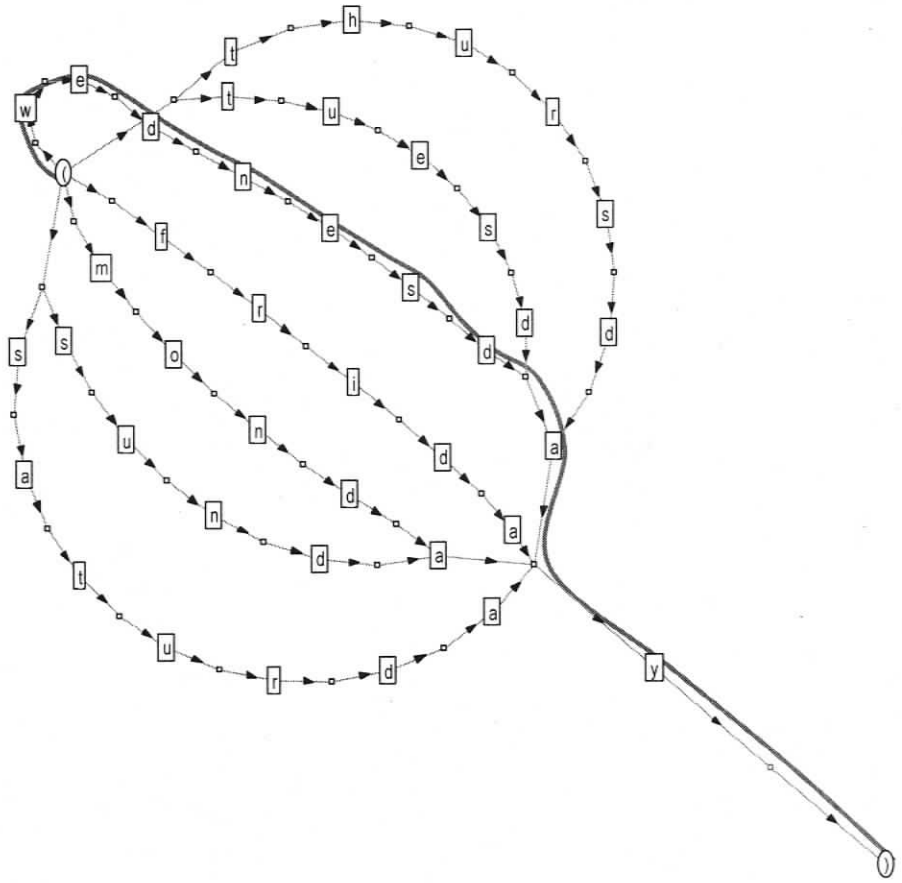


Figure 25. Least ambiguous representation of the seven strings; there are only 7 possible paths from the start symbol "(" to the end symbol ")"

In Figure 25, the least ambiguous representation of the weekday names is achieved: there are only 7 possible paths in the directed graph that represents them and hence the number of outside sequences has been reduced to zero. The representation of three of the seven sequences (i.e., the ones that start with a specific, non-ambiguous character, *monday*, *wednesday* and *friday*) has attained the ambiguity of 2.32 bits while the rest are represented with an ambiguity of 2.32+1.00=3.32 bits, due to the additional ambiguity of 1.00 bit incurred by the common start letters *t* for *tuesday* and *thursday*) and *s* for *saturday* and *sunday*. Therefore, the total ambiguity of the representation of weekday names is at least $\log_2(5) = 2.32$ bits and at most $\log_2(10) = 3.32$ bits, meaning that any sequence can be completely determined by its prefix after answering one or two questions: the first with 5 equiprobable choices (i.e., between *m*, *w*, *f*, *t*, *s*) and, if

necessary, an additional one with 2 equiprobable choices (i.e., between *tu* and *th*, or *sa* and *su*).

An additional interesting feature of this representational approach is the possibility to consider the reverse path of each sequence. The number of possible reverse paths is also 7 and the representation ambiguity is $\log_2(4)=2$ bits for *friday* and *saturday*, $\log_2(4)+\log_2(2)=3$ bits for *monday*, *sunday* and *thursday* and $\log_2(4)+\log_2(2)+\log_2(2)=4$ bits for *tuesday* and *wednesday*. Therefore, the total ambiguity of the reversed representation of weekday names is at least $\log_2(4)=2$ bits and at most $\log_2(16)=4$ bits, meaning that any sequence can be completely determined by its suffix after answering between one and three questions: the first with 4 equiprobable outcomes and, if necessary, a second one with 2 equiprobable outcomes, followed, if necessary, by a third one with 2 equiprobable choices.

Finally, though not explicitly shown in the representation in Figure 25, there exists the possibility that some sequences are perfectly determined only by the 1.0 bit of information contained in the choice between two diafixes, such as the decision case of the equiprobable choice between characters *o* and *i* for the sequences *monday* and *friday*, respectively.

Generalizing, it appears that the complete determination of a representation of a weekday name is contingent to a decision between the characters "(" (i.e., the start character), *m*, *t*, *w*, *f*, *s*, *r*, *o*, *i*, *e*, *h*, *u*, *n*, *d*, *a*, *y*, and ")"(i.e., the stop character) which incurs an ambiguity of $\log_2(17)=4.09$ bits to which one has to add a variable amount of uncertainty caused by the ambiguity of each individual character in the representation and which varies between 0.0 bits in case of characters specific to a weekday (e.g., *f*, *w*, *h*, *o*, *i*) and $\log_2(8)=3$ bits in the case of highly ambiguous characters such as *d*. As it will become clearer in the following, at the expense of space complexity, the DDAM model is able to minimize the variable ambiguity in representations by adaptively capturing a variable amount the context around the ambiguous representational nodes.

2.6.4.2 Model simplification

So far, the representations in Figure 21 throughout Figure 25 have been simplified in order to facilitate understanding. The actual representational approach involves significantly more complex structures which have been partially obscured in order to yield an uncluttered, aesthetic display.

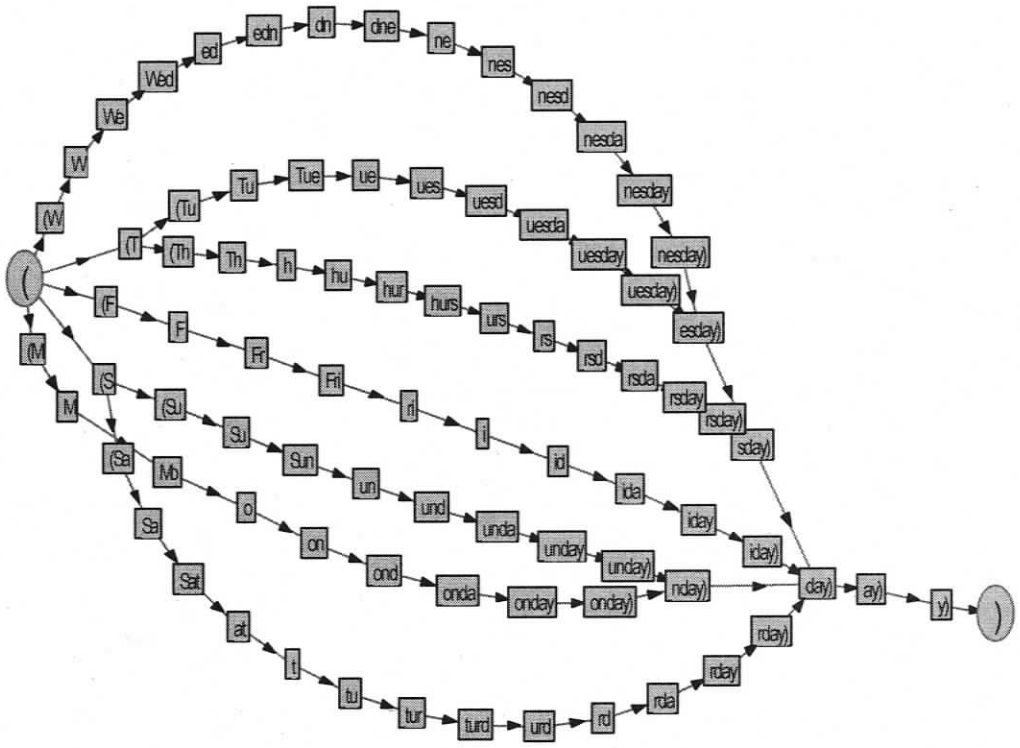


Figure 26. The actual, least ambiguous, context-dependent representation of the seven strings in the DDAM model in which nodes actually correspond to substrings, i.e., characters and their prefix and suffix contexts

Though still partially obscured, the structure in Figure 26 is isomorphic to that in Figure 25 and meant to suggests that, actually, the representational nodes are not corresponding directly to letter symbols but to more complex patterns which capture the necessary amount of context required to minimize ambiguity. This essentially renders the representations context-dependent. For example, the 8 instances of the character *d* in the seven weekday names correspond in reality to the 8 patterns *rda*, *onday*, *unday*, *iday*, *rsday*, *uesday*, *nesday* and *edn* in Figure 26. By capturing enough prefix and suffix

context around each of the 8 *d* character instances, these more complex patterns have all become unique and have render all representations of weekday names non-ambiguous. The representation algorithms proposed in this dissertation are aimed at obtaining the same kind of representations from arbitrary unstructured set of sequences.

2.6.4.3 Alignment of representations

The representations in Figure 21 throughout Figure 26 have maintained the exact same length equal to $2n+3$ (including the start and end characters), where n is the length of the original sequence. For example, all four representations of the string *(wednesday),* starting with the trivial representation in Figure 21 and ending with the least ambiguous representation in Figure 25 and Figure 26, contain the exact same number of nodes, equal to $2\times 9+3=21$ nodes, where 9 is the length of the string *wednesday*. This important property allows one to perfectly align representations, regardless of their ambiguity, and to derive one representation from another through highly efficient algorithms. As formally described in the next chapter in the context of the "flip theorem", the estimated time complexity of such algorithms is $O\left(n^2\right)$ in the worst case, i.e., quadratic in the length n of a sequence.

2.6.4.4 Conclusions

The DDAM model is a generalization of the trie structure, a directed graph that is able to achieve representations of sequential data with variable ambiguity levels that range from highly ambiguous representations (i.e., trivial) to minimum ambiguity representations. So far, the depictions of the DDAM model have been purposely simplified in order to facilitate understanding. One of the most important features of the DDAM model is that all representations of a sequence in DDAM are perfectly aligned and can be transformed from one into another by a string composition algorithm. The worst case temporal and spatial complexity of the string composition algorithm are quadratic and will be demonstrated by a theorem that is to be introduced formally in the following chapter. In addition, distinct sequences are aligned at their similar subsequences and, as will also be demonstrated in the next chapter, this alignment forms the basis of the similarity retrieval capabilities of the DDAM model.

The most important limitations of the model arise from the memory (i.e., space) requirements which, in the worst case, are quadratic in the length of sequences, as well as from the fact that the more distant two patterns in a sequence, the more difficult for this model to explicitly capture and handle the statistical dependencies between them. Though overcoming this limitation would be possible through additional innovations, these will most likely increase the spatial and temporal complexity of the model and hence are not being pursued at this time.

2.7 UNSUPERVISED CONCEPT SPACE REPRESENTATION TASKS

This section of the theoretical background discussion is a functional perspective that reviews unsupervised functions, tasks and processes to represent associative concept spaces.

As in the case of biological systems where function and structure are highly interrelated, the presentation of processes and models in separate sections is somewhat artificial. However, such separation of concerns is often useful to increasing the clarity of a presentation. In this section unsupervised concept space representation processes comprise information processing functions, tasks and problems that require little or no human intervention in order to be accomplished.

2.7.1 The duality of unsupervised information processing

In (Solomonoff 2003) Solomonoff describes a dual nature of problems in science which, according to him, are:

1. Function inversions, i.e., the typical combinatorial problems of computational complexity theory (e.g., traveling salesman, theorem proving, etc.),

2. Time limited optimizations – (e.g., surface and image reconstruction, automobile design, etc.)

This separation bears a striking similarity with the distinction between the simple, non-evolving, context-independent, non-adaptive representation functions and the complex, evolving, context-dependent adaptive representation functions, and resonates very well with the duality of general and individual knowledge and with the distinction between theoretical and practical sciences, discussed in Chapter 1.

The problems of the first kind are the typical NP complete (i.e., non-deterministic polynomial time) problems. Because they often have compact solution and representation

spaces, their exact solutions are possible for small dimensions (e.g., traveling salesman problem for 10 cities) but become intractable as soon as one moves at the very next order of dimensionality (e.g., a traveling salesman problem for 100 cities).

The problems of the second kind, on the other hand, are typically very high dimensional but their solution space is much sparser, as is the case of automobile design or medical decision making where very few viable solutions may exist for a given problem whose representation dimensionality can easily reach thousands of dimensions. Their dimensionality makes them susceptible to knowledge acquisition bottlenecks and their time-constraints to the frame problem. At the same time, problems of the second kind are solvable only by approximate means involving various forms of inductive inference and pattern recognition and their solutions can never be proven optimal. As argued in this dissertation, one such example of inductive inference at which humans excel thanks to their capacity to learn from experience, to discover and recognize patterns and to apply solutions efficiently, is case based reasoning (CBR). Since the focus in this dissertation is on problems of the second kind, all information processing tasks discussed here are necessarily forms of inductive inference and inductive reasoning based on pattern discovery and recognition mechanisms, and include:

- 2.7.2 Grammar induction
- 2.7.3 Text segmentation:
    - 2.7.3.1 Morphosemantic decomposition
    - 2.7.3.2 Word segmentation
- 2.7.4 Sequence alignment
- 2.7.5 Information compression
- 2.7.6 Unsupervised classification:
    - 2.7.6.1 Cluster analysis and visualization

- 2.7.6.2 Information retrieval

2.7.2 Grammar induction

The specification, instructions or the set or rules of how to construct something from its components is a grammar. From the knowledge spectrum perspective, a grammar is a theory or a model and applying it in order to construct something equates to a *theory application* that moves one from the abstractions side towards the reality side of the spectrum. However, real world problems often pose the reverse of this problem, namely, *theory elicitation* or *grammar induction*, which consists of making up the specifications, instructions, rules, grammar, given a large enough set of objects. In the case of computer strings, inducing a grammar is the basis to creating a more meaningful, potentially compressed packaging that changes some of the properties (e.g., dimensionality, sparseness) of the representation space in which the strings are represented.

Though grammars can describe finite sets of objects, grammar induction is often regarded as "the identification of an infinite structure [i.e., a language] with a finite structural description [i.e., the grammar] on the basis of a finite number of examples" (Adriaans and Zaanen 2004) a definition which reiterates the idea that grammars are abstractions (i.e., theories). For natural languages, grammar induction is also an extremely difficult problem, especially if one expects results to resemble the syntactic analysis derived by a linguist (Manning and Schütze 1999). Even if one does not aim specifically at linguistically correct structures, there are theoretical proofs which prevent results that satisfy optimality criteria (Adriaans and Zaanen 2004). For example, the problem of deriving the *smallest grammar* – with obvious applications in data compression – is known to be NP complete and the possibility to identify a context free grammar has already been proved formally (Gold 1967) to be unattainable only from limited positive examples of an infinite language (i.e., identification in the limit). As a consequence, grammar induction approaches generally aim at approximate results, are guided by purpose (e.g., syntactic parsing, chunking, semantic disambiguation, etc.) and typically try to make use of any available apriori knowledge in order to improve results (i.e., supervised approaches).

Despite its difficulties, the problem of grammar induction is an intensively studied topic in various contexts but predominantly in language acquisition domains (Adriaans and Zaanen 2004), hierarchical chunking (Nevill-Manning 1996; Wolff 2004), syntactic parsing (Wolff 1988; Edelman, Solan et al. 2005) grammatical inference (Hutchens 1994), unsupervised language acquisition (Solan, Horn et al. 2004).

In this dissertation, in order to keep the models and their processing capabilities as generally applicable as possible, the focus is on unsupervised approaches that remain as domain independent and knowledge free as possible.

2.7.2.1 Syntactic systematicity

Coined by Fodor and Pylyshyn (Fodor and Pylyshyn 1988), the term "systematicity," essentially, seems to refer to the cognitive ability of associating representations that share similarities. Though *syntactic systematicity* was subsequently defined more precisely (Hadley 1994) and researched (Hadley, Rotaru-Varga et al. 2001) in the *supervised learning* paradigm, the issue is still very relevant to this discussion. Of particular interest is the definition of *weak syntactic systematicity* as the capacity to generalize the use of a lexical item to the same syntactic position but in novel sentences.

Grammar induction often results in rules that share similarities. If matched appropriately, similar rules can be subsumed by one rule that contains an *equivalence set* in its right hand side. In the case of syntactic analysis of natural languages, such equivalence sets may correspond to known *parts of speech* such as nouns, verbs, etc. The elements of these equivalence sets are lexical items which share syntactic (and potentially semantic) similarities, i.e., they all have the property that, if replaced by other items taken from the same equivalence set, this operation does not affect the well-formed-ness of that sentence – though such changes often affect meaning – for a given grammar.

Grammar rule	Example
NP1 → The {doctor | nurse}	The nurse
NP2 → {patient | boy | female | girl | male | man | woman}	man
NP3 → the {15 | 20 | 25 | 30 | 35 | 40} year-old NP2	the 30 year old NP2
P → {sedated | observed | interviewed | calmed down | examined | talked to}	examined
S → NP1 P NP3	The nurse examined the 30 year-old man

Table 9. Examples of a context free grammar; rules that share similarities are written as equivalence sets

For example, given the context free grammar in Table 9 and the example sentence, it can be easily seen that by replacing the noun *man* and numeral *30* and the verb *examined* with any of the elements in the their equivalence sets, preserves the well-formed-ness of the sentence. The context free nature of such grammars stems from the unrestricted use of the rewrite rules that could be employed at any time, regardless of what has been generated so far (i.e., the prior context). Though this causes the impossibility to prevent the generation of syntactically correct but semantically inappropriate statements such as "*The doctor sedated the 40 year-old boy*" this is the basic mechanism for context free language generation.

Because one of the aims in this dissertation consists of the unsupervised creation of equivalence sets from text, the original definition of *weak syntactic systematicity* could be adapted as "the capability of a language processing model to assign a lexical item into an appropriate equivalence set, in an unsupervised manner." As it will become evident from the results of experiments carried out in Chapter 4, the DDAM as well as other models capable of grammar induction, are able to satisfy, to various extents and in slightly different ways, this criterion of weak syntactic systematicity.

2.7.2.2 Context-dependent grammar induction – a pleonasm?

The difficulty of function inversions and optimization algorithms such as grammar induction seems to arise from the need that the algorithms step at a higher order of complexity that permits the inductive process. Therefore, inducing a context free grammar seems to require a kind of information processing which is eminently context-dependent. For example, given a set of well-formed sentences for the grammar in Table 9, one mechanism to derive equivalence sets (e.g., NP2) is to *match* the immediate contexts around lexical items in every sentence and place in the same set, the items that

share the same contexts. Therefore, grammar induction models may have to necessarily be based on context-dependent information processing approaches. If this is so, then "context dependent grammar induction" is indeed a pleonasm and this leads to an important question: how much context could an algorithm afford to capture while leaving the induction tractable for languages of reasonable sizes. Given that an exhaustive search for a good grammar that uses global objective functions is problematic, one reasonable answer is that approaches based on information which is local to a representation could be the ones that scale up better. The design of the DDAM model follows this observation.

2.7.3 Text segmentation

For the purpose of this dissertation, text segmentation is considered a form of grammar induction that aims at the decomposition of a text into a series of compositional building blocks that may be morphemes, words or phrases. Therefore, this processing includes what in literature is referred to as *morphosemantic decomposition* and *word segmentation*. This definition also includes the situations where the text may be artificially created from nonsense syllables and words as well as the case where separators (e.g., blanks, commas, periods, brackets, etc.) – which are normally used in order to separate lexical items in many languages – are all removed in order to eliminate the importance of separators in unsupervised lexical acquisition evaluation tasks.

2.7.3.1 Morphosemantic decomposition

Morphosemantic decomposition is a form of grammar induction which aims at the decomposition of complex lexical items - in particular of compound words from technical, professional discourses - into their semantic compositional building blocks (i.e., morphemes).

Arguably, learning the morphology of general natural language is a more difficult problem than learning the morphology of a technical language used in professional discourses. Unlike the former, which may be considered a somewhat artificial task due to the fact that most language acquisition seems to start at word level (Goldsmith 2001; Creutz and Lagus 2002), morphological analysis is of extreme importance for

professional discourses where compound terms abound and for which semantic processing, information retrieval and automated reasoning are very important tasks. As a consequence, morphosemantic decomposition of biomedical terms has been an intensely studied problem in recent years, though exclusively in the supervised paradigm (Lovis, Baud et al. 1997; Baud, Rassinoux et al. 1999; Rassinoux, Ruch et al. 2000; Schulz and Hahn 2000; Baud, Lovis et al. 2001; Hahn, Honeck et al. 2001; Schultz, Honeck et al. 2002).

2.7.3.2 Word segmentation

Word segmentation is a form of grammar induction in which unsegmented text (i.e., text without any separators or punctuation) is segmented in words. Though unusual for languages such as English, word segmentation is of high interest and forms the object of computational linguistics approaches for the Asian languages whose writing systems do not include the use of separators (e.g., Chinese) (Marcken 1996; Teahan 1998; Kit 2000; Brent and Tao 2001; Schone 2001).

For English language, much of the interest in word segmentation has stemmed from research on hierarchical chunking (Wolff 2004) and from the study of language acquisition (Olivier 1968; Wolff 1977; Brent and Cartwright 1996; Marcken 1996; Brent 1999; Venkataraman 2001; Batchelder 2002) that aims at explaining how children acquire language words, given the little or total lack of feedback information they receive with regard to the word boundaries. Other word segmentation tasks have been reported in the context of optical character recognition (OCR) systems (Teahan 1998) which may result in streams of text where word boundaries are occasionally suppressed.

Unsegmented text has variable levels of ambiguity and difficulty of parsing, which occasionally may pose difficulties even to human language processors, especially for uncommon words. However, when facing this somewhat unusual task, humans usually employ multiple strategies and make use of any useful piece of information that helps them in the process, including high-level semantic knowledge about the concepts present in a text. Performing such a task, especially on a "difficult text," may provide insights into the levels of processing and levels of information needed for text understanding. A

human may need to read the text multiple times and make extensive use of his/her knowledge because of the increased ambiguity. The use of artificially generated text that comprises nonsense words only exacerbates these processes.

Artificial data comprising nonsensical lexical items could be thought of as being able to bring human processors closer to a more primitive information-processing model by removing some of our powerful semantic processing capabilities. This may provide additional insights into processing mechanisms and could help with the development of pattern discovery and recognition algorithms suitable for lexical acquisition and associative information retrieval, such as the DDAM model. The German psychologist Hermann Ebbinghaus was the first to empirically investigate associationist memory mechanisms in this manner. In his attempts to objectively measure the association power of human memory, Ebbinghaus has made use of nonsense syllables (e.g., NUH, VEG, KUR, etc.) as they "have the property of removing certain cross associations that manifest and are variable from person to person" (Bourtchouladze 2002)[15]. The fact that statistical properties of artificial texts are easy to control and the experimental results of such tasks are relatively easy to quantify, compare and discuss, makes artificial text useful for the evaluation of segmentation models and algorithms (Wolff 1975), (Elman 1990).

In addition to data-driven associative and pattern recognition capabilities which can be currently emulated to some extent by the DDAM model, reaching the word segmentation proficiency of human parsers for natural languages may require extensive semantic knowledge. Such integration is currently difficult and begins to resemble advanced approaches often described in literature as hybrid information-processing models that are beyond the scope of this work.

---

[15] Chapter 1 of this reference contains the discussion about the work of H. Ebbinghaus.

2.7.4 Sequence alignment

Perhaps surprisingly, problems that arise from the quest for intelligent, human-like information representation and processing models converge with those in the research on genomics and proteomics. The commonality between the two apparently distinct fields of research resides in the need to efficiently address and solve the task of *sequence alignment*. Some researchers have already perceived this convergence and regard it as a unifying principle (Wolff 2004) while others explicitly employ sequence alignment techniques (Van Zaanen 2002; Solan, Horn et al. 2004; Edelman, Solan et al. 2005) and representations of aligned sequences (e.g., sausage graphs, word lattices, lexical chains, etc.) (Barzilay and Lee 2002; Barzilay and Lee 2003).

A glimpse of this convergence has already been offered to the reader in this chapter through the depictions of the DDAM model in Figure 21 to Figure 26 and by the explicit remarks with regard to the DDAM string representations that are able to capture diafix similarities (e.g., the pattern *ur* in *thursday* and *saturday*, in Figure 20). It has also been suggested by the DDAM representations of the same string that can be perfectly "aligned" due to the fact that they maintain the same length, regardless of the ambiguity of their representations.

Therefore, the representational approach of DDAM could be regarded as a sequence alignment where sequences align at their similar regions. It also suggests a possible future extension of this research to the bioinformatics research where genomic and proteomics sequence alignment are standard tools (Notredame 2002). In this context, possible applications include DNA and protein motif extraction, phylogenetic analysis, protein classification and protein structure prediction.

To conclude, sequence alignment is just an alternative way to look at the problem of managing high dimensional, sparse, dynamic spaces, which forms the object of DDAM model introduced in this dissertation, and opens the perspective of applying the DDAM model to bioinformatics data contained in DNA and protein sequences.

2.7.5 Information compression

The usefulness of information compression is not a matter of debate. However, in the context of algorithmic information theory and minimum description length (MDL) it has been argued earlier that the degree to which information is compressed requires careful consideration. This is so because of the important relation between the compression and the algorithmic complexity of the information processing models. A compression method that squeezes as much redundancy as possible out of representations may not be adequate for slow information processors which possess adequate amounts of memory and whose modus operandi may be governed by minimum description work (MDW) rather than MDL principles. In this context, it has also been shown that hierarchical, compositional representations agree with this alternative criterion in their inherent ability to compress information to some degree, while still allowing for efficient retrieval, in the MDW sense.

To conclude, any information-processing model capable of unsupervised induction of structure from raw data has an obvious connection with the task of data compression and, if necessary, could be modified to specifically address it. This conclusion is clearly supported by the many published models which have been studied in both induction of hierarchical structure and information compression contexts (Wolff 1982; Nevill-Manning 1996; Nevill-Manning and Witten 1997; Cleary and Teahan 1998; Teahan 1998) even though most of them have been explicitly designed on MDL principles.

The DDAM model is able to attain the inherent compression caused by a design that specifically avoids the duplication of representations. Converting the model to perform data compression in the MDL sense would permit comparison with existing models in compression benchmarks. Though perfectly feasible, this line of research is not explicitly pursued in this dissertation.

2.7.6 Unsupervised classification

As in the case of information compression, the task of classification is a natural fit in the context of models capable of unsupervised induction of hierarchical structure of

information. For the purpose of this dissertation, structurally and functionally, unsupervised classifications tasks can be classified [sic] in:

- Simple
    - Hierarchical cluster analysis (joining tree)
    - Visualization techniques (self organizing maps)
- Multiple
    - Static multiple classifications (multiple hierarchies)
    - Dynamic multiple classification (information retrieval)

Simple classifications involve the creation of hierarchical representations and 2-dimensional visualizations that reveal underlying structure of data. Unsupervised multiple classification implies the induction and exploitation of multiple hierarchy structures, where an item can be classified multiple times, under multiple categories. In this dissertation, the focus is on such hierarchies but which are also dynamic in nature. On closer inspection, the objectives of inducing and using such structures are equivalent with those of information retrieval where a document is indexed based on its content and subsequently dynamically classified as belonging to any category that is determined by a query to which that particular document was determined to be relevant.

2.7.6.1 Cluster analysis and visualization

Multidimensionality of data and our limitations in visualizing more than three spatial dimensions require that multidimensional data sets be projected onto lower dimensional subspaces. Though this process causes invariably a loss of some features in the original data, if the projection is done judiciously the projection could be useful.

The universal nature of cluster analysis is easy to surmise. Cluster analysis is helpful to organize observed data into meaningful structures and to develop useful, data-driven taxonomies and classifications (Kaski 1997). As a result, cluster analysis is a method with

uses in many fields of research including unsupervised natural language processing (e.g., for a review and an application of hierarchical clustering for unsupervised lexicon generation see (Hodge and Austin 2002)).

Concretely, cluster analysis is a method of exploratory data analysis that aims at partitioning a set of data items into groups (i.e., classification), based on a similarity (or dissimilarity) measure or distance. The resulting groups or categories are called clusters and their number may be preassigned or determined automatically by the algorithms. Because it requires a pre-specified number of clusters, the former case is not an entirely unsupervised procedure. This is the case of k-means clustering procedure which classifies objects by moving them into different clusters with the goal of minimizing the intra-cluster distances while maximizing the inter-cluster distances. The latter case, a completely unsupervised procedure, is that of joining algorithms which aggregate (or amalgamate) increasingly larger clusters of increasingly dissimilar patterns (Figure 27).

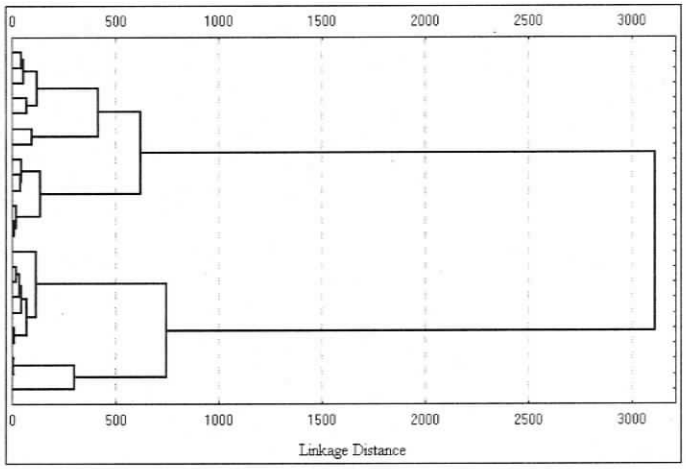


Figure 27. Joining tree (dendrogram) example; there are two very well defined clusters, corresponding to the two branches of the tree spanning over a wide linkage distance range (from approx. 750 to 3000)

Clustering is also a relative notion. The intra-cluster and inter-cluster distances may vary from data set to data set and because of this, a variable clustering threshold has to be determined for every given problem. In Figure 27, this is equivalent to visually inspecting the amalgamation trees followed by determining a threshold value and the corresponding

number of clusters. If this is achieved automatically the procedure remains unsupervised while if not, then the procedure is partially supervised.

The visual representation of a multidimensional data set can also be achieved using self-organizing maps (SOM, feature map, Kohonen map) (Kohonen 2001). A biologically plausible model of artificial neural network able to provide a convenient 2-dimensional visual representation of high dimensional input data, SOM uses an unsupervised learning algorithm based on a distance calculation causing each input pattern to be associated with a zone on the resulting map (Figure 28).

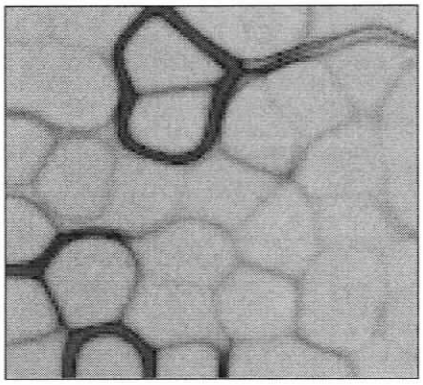


Figure 28. Example of self-organizing map; each data point has a corresponding zone in the map and it is separated from the other points by a "valley" of variable width (e.g. near the top of the picture one cluster made of two patterns is clearly separated from the surrounding patterns by a wide boundary)

The real spatial distances between input patterns are not respected in the sense that two patterns that are close to one another topologically are not necessarily similar from the distance point of view. These patterns will be actually separated by a "valley" (Figure 28) covering potentially a significantly wide input data space void of input patterns. By contrast, two data points that are close in terms of distance will also be close topologically but there will be a thinner boundary separating them. In this way, the 2-dimensional mapping space becomes a warped projection of the multidimensional input space, with the input patterns squeezed into one another and separated by variable width boundaries.

Self-organizing maps can be useful in that their appearance may give a first glimpse of the underlying properties of the data and of the distribution of the data points in the

multidimensional representation space. For example, the degree of sparseness of a dataset could be determined. SOM can also help to identify similarity relations between multidimensional data points and the number of clusters in a data set. Unlike joining cluster analysis algorithms, the SOM do not generally set any limit on the number of patterns in data sets. Besides classification and retrieval, SOM is also used in exploratory data analysis because it creates a visual description of multidimensional data that has the potential to point to data errors.

The relevance of SOM to DDAM is significant. Like SOM, the DDAM model is an associative memory (albeit a deterministic one) whose structure, when displayed appropriately using force-directed automated graph layout algorithms, gains structural and functional capabilities which are highly similar to those of SOM.

2.7.6.2 Information retrieval

One could argue that the ultimate purpose of creating representations of information is to enable their subsequent retrieval and processing on similarity principles (i.e., information retrieval, similarity-based retrieval, associative recall). Typically, this entails having some partial information or context in the form of a query and then retrieving the representations that are most similar/relevant to the representation of that query. In a conceptual space whose organization obeys similarity principles, this kind of similarity-based retrieval would be just a read function that returns the items residing within a predefined search radius from a given query.

One of the best analogies of this kind of associative recall is that of a hyperspace telescope, which can be centered on a query, allowing one to visualize the high dimensional similarity neighbourhood of that query, within a predefined recall radius expressed usually in bits. The radius value corresponds to the associative recall radius that limits the number of items that can be visualized at one time. For example, in Figure 29 to Figure 34, a hypothetical hyperspace telescope with increasingly larger associative recall radius up to 3.9 bits is centered on the query "abcde" in the context of a collection of 10,000 strings of lengths up to 9 characters, generated randomly from the alphabet

$\{a,b,c,d,e,f,g,h,i,j\}$. The representation space of the randomly generated strings is therefore highly sparse as it contains only 10^4 elements out of all possible 10^9 strings, that is just 0.001% of all possible strings. Through the hyperspace telescope we are able to visualize the items in the associative memory which are similar to the query, within a specified radius.

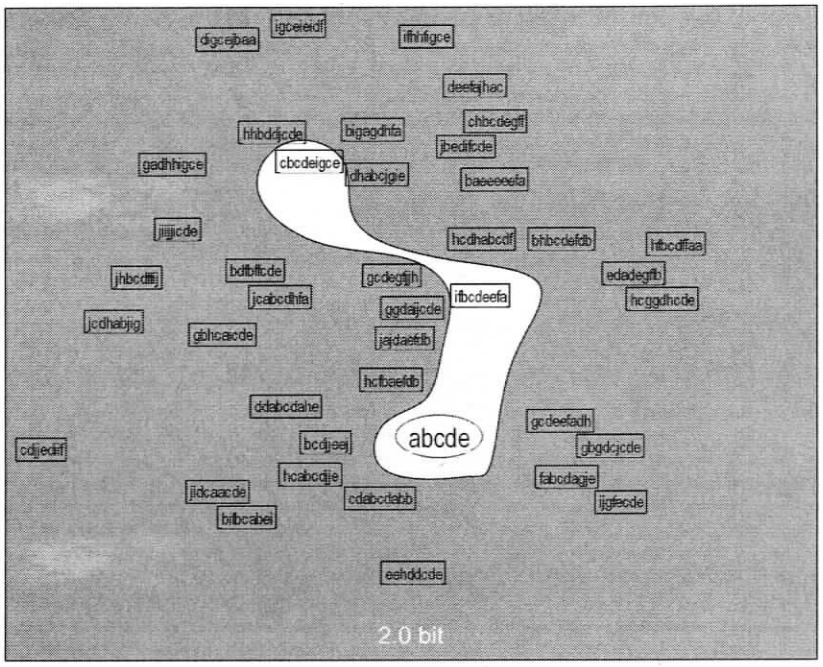


Figure 29. Hypothetical hyper-space telescope having an objective radius (i.e., recall radius) of 2.0 bits which allows the observation (i.e., retrieval) of the most similar items to the query "abcde"; increasing the objective radius allows the visualization of additional objects

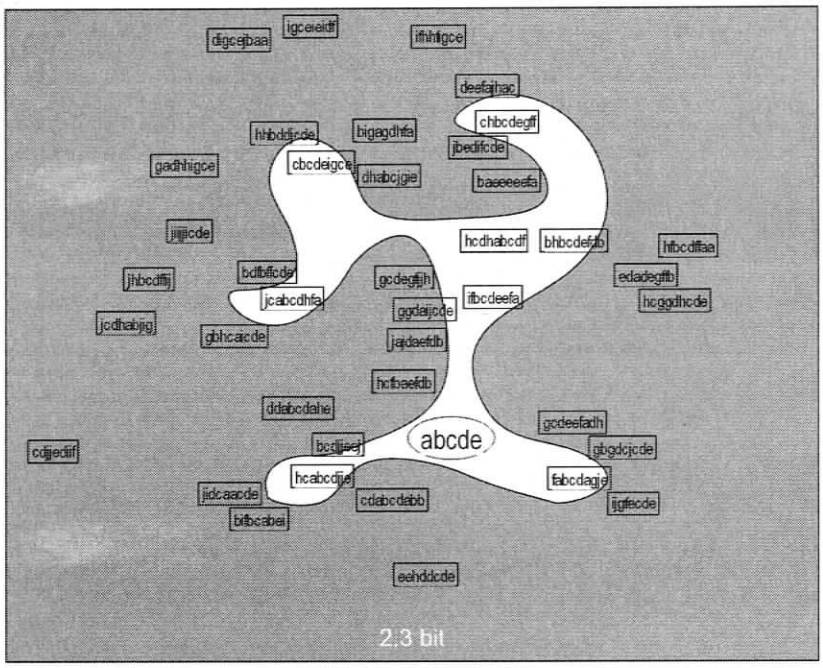


Figure 30. Hypothetical hyper-space telescope having an objective radius (i.e., recall radius) of 2.3, bits which allows the observation (i.e., retrieval) of the most similar items to the query "abcde"; increasing the objective radius allows the visualization of additional objects

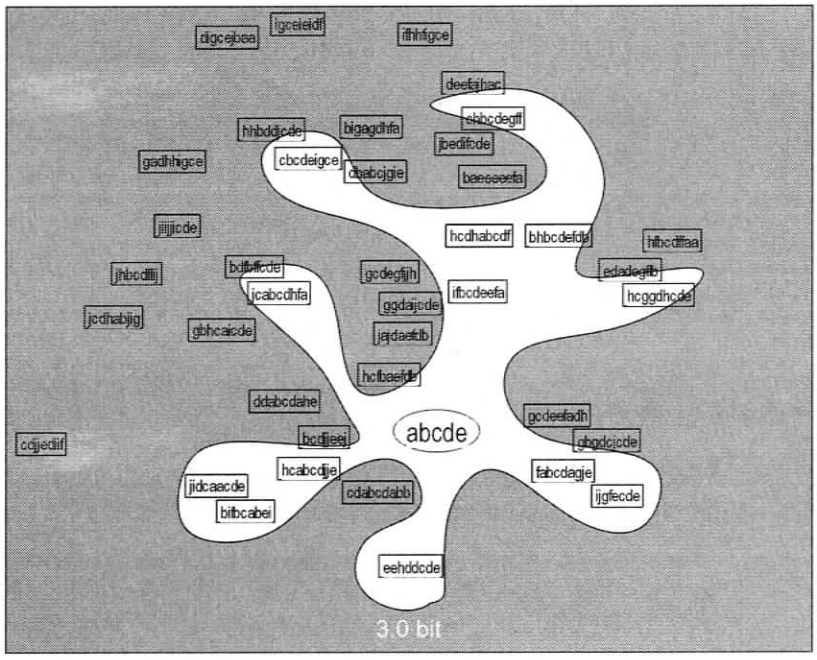


Figure 31. Hypothetical hyper-space telescope having an objective radius (i.e., recall radius) of 3.0 bits which allows the observation (i.e., retrieval) of the most similar items to the query "abcde"; increasing the objective radius allows the visualization of additional objects

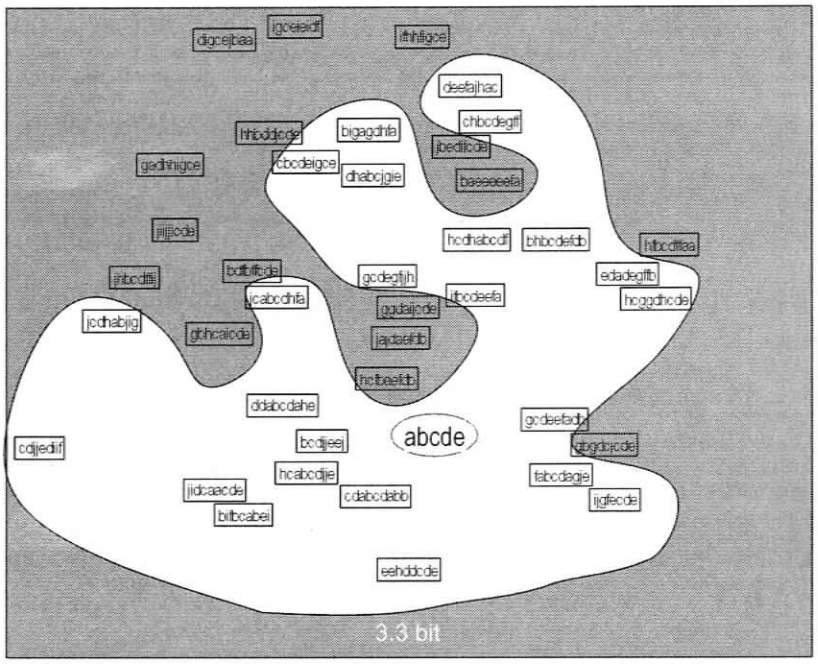


Figure 32. Hypothetical hyper-space telescope having an objective radius (i.e., recall radius) of 3.3, bits which allows the observation (i.e., retrieval) of the most similar items to the query "abcde"; increasing the objective radius allows the visualization of additional objects

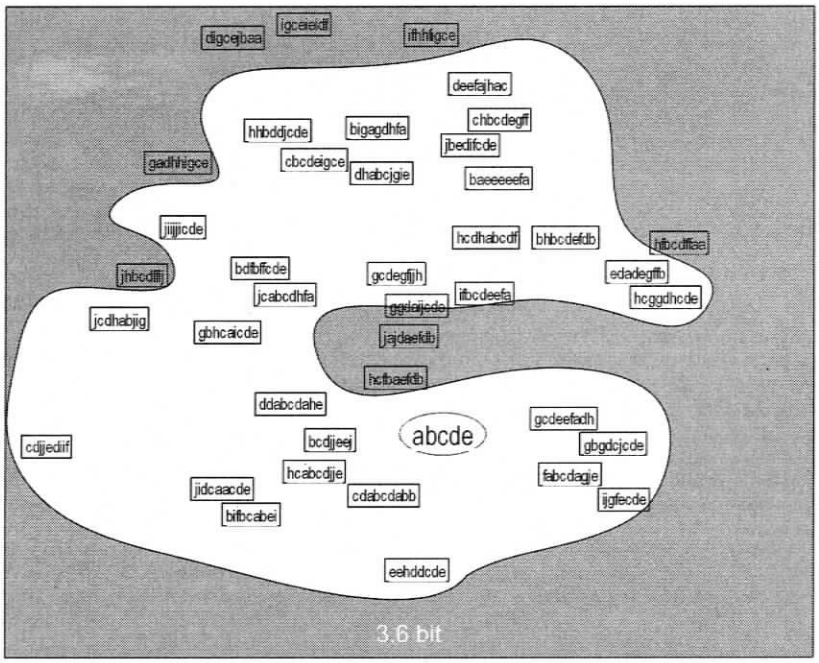


Figure 33. Hypothetical hyper-space telescope having an objective radius (i.e., recall radius) of 3.6 bits which allows the observation (i.e., retrieval) of the most similar items to the query "abcde"; increasing the objective radius allows the visualization of additional objects

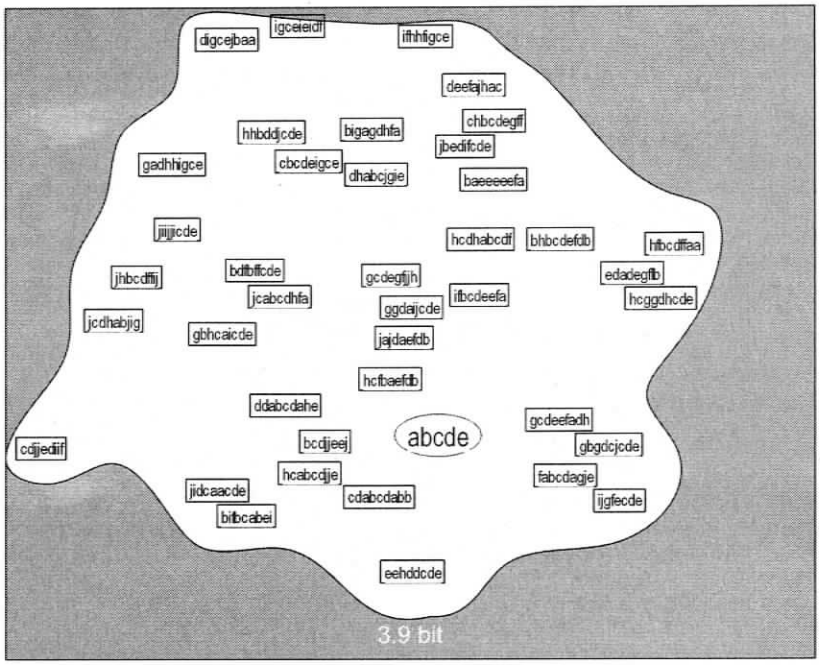


Figure 34. Hypothetical hyper-space telescope having an objective radius (i.e., recall radius) of 3.9 bits which allows the observation (i.e., retrieval) of the most similar items to the query "abcde"; increasing the objective radius allows the visualization of additional objects

The digitated shapes of the objective of the hyperspace scope are projections of a multi-dimensional space onto the 2-dimensional display. Their exact form is determined by the properties of the pattern space which, in turn are determined by the algorithmic properties of the patterns that are represented in that space. In Table 10, the retrieved strings are shown on columns that correspond to the distance, in bits, from the query on which the hyperspace telescope was centered.

0.0	2.0	2.3	3.0	3.3	3.6	3.9
abcde	if**bcde***efa*	jc**abcd***hfa*	eehdd**cde**	**cd**jjediif	bdfbff**cde**	hf**bcd**ffaa
	c**bcde***igce*	hc**abcd**jje	ijgfe**cde**	g**cde**gfjjh	hhbddj**cde**	jh**bcd**ffij
		hcdh**abcd**f	jidcaa**cde**	g**cde**efadh	jiijji**cde**	
		f**abcd**agje	hcggdh**cde**	jc**dh**a**b**jig	jbedif**cde**	*igce*ieidf
		ch**bcde**gff	bif**bca**bei		ggdaij**cde**	d*igce*jbaa
		bh**bcde***fdb*		eda**de**gffb	gbgdcj**cde**	if*hhfigce*
				de**ef**ajhac	gbhcai**cde**	gadh*higce*
				dh**abc**jgie		
				cd**abcd**abb		jajd*aefdb*
				dd**abcd**ahe		hcfb*aefdb*
				bcdjjeej		
				baeeee*efa*		
				bigag*dhfa*		

Table 10. The results of the DDAM associative recall on the query "abcde" on a collection of 10,000 random strings constructed from the alphabet {a, b, c, d, e, f, g, h, i, j} using an increasingly large bit radius, from 0.0 bit to 4 bit; direct similarities with the query "abcd" are shown bold and underlined, indirect associative similarities are shown in italics and underlined

Upon closer inspection, a direct consequence of this kind of associative recall becomes clear. Besides the retrieved strings that show obvious similarities with the query (e.g., c**bcde***igce*) there are some others that appear to have nothing in common with the original query (e.g., *igce*ieidf, d*igce*jbaa). Yet they do, not directly, but indirectly through other mediating patterns (e.g., *igce*) and with whom they are associated strongly enough to make it possible to be retrieved within a large enough search radius. Translated in a real information retrieval situation where strings are representations of documents and patterns in them are features (e.g., morphemes, words, phrases, etc.), this mechanism would allow the retrieval of documents which do not necessarily contain the query but may contain other features which are indirectly but strongly enough associated with the query (e.g., synonymous words, similar phrases, relevant contextual cues, etc.).

This kind of functionality likens the DDAM model to existing approaches such as latent semantic indexing (LSI) (Landauer and Dumais 1997; Hofmann 2001; Kintsch 2001; Brants, Chen et al. 2002; Kintsch 2002), which are also known to be able to represent and recall information based on indirect associations (e.g., a query on "streptococcus" may recall documents in a biomedical collection that contain the phrase "throat infection" due to the strong association between the two).

This associative recall mechanism, envisioned in this particular way, was inspired by Kanerva's Sparse Distributed Memory (SDM) model (Kanerva 1988) and is fundamental to the information retrieval possibilities of the DDAM model. At the same time, such a mechanism is extremely relevant to case based reasoning and information retrieval, two apparently different fields of research whose strong association was already argued for in this dissertation.

2.7.7 Conclusions

The tasks reviewed in this section provide a good perspective on the range of functions that the DDAM model is envisioned to perform. Though seemingly disparate, information processing tasks discussed are all difficult forms of inductive inference based on pattern discovery and recognition mechanisms. The fact that DDAM model has the

potential to tackle, to various extents, each and every one of these tasks demonstrates the generality and the validity of the model.

2.8 CONCEPT SPACE REPRESENTATION MODELS AND APPROACHES

In this section the theoretical background discussion continues with a structural perspective that reviews unsupervised models and approaches which share similarities with the DDAM model in representing associative concept spaces.

2.8.1 Introduction

So far it has been argued that dynamic, multiple inheritance systems or information retrieval systems are able to create automatically, useful, content-based classifications of the informational objects stored in them. This reiterates the idea that they must belong to the same area of research as associative memory models such as the DDAM model. However, in the context of pattern recognition and analogy-making, state of the art knowledge processing models are still largely incapable, for example, of dynamically classifying the concept of "claw hammer" into a dynamically created class of "back-scratching devices," in the semantic neighbourhood of the "itch," "scratch" and "claw" concepts. This would require automated information indexing, retrieval and processing capabilities that work on conceptual principles (Woods 1997; Baud, Lovis et al. 2001) whose complexity is still beyond current technology.

Furthermore, it is very unlikely that this kind of dynamic classification capacity of the human semantic processor could work on fixed, static structures constructed beforehand during a learning phase, in human semantic memory. This casts doubts on static models of human knowledge processing such as semantic networks and ontologies built and updated mainly by manual effort. A more plausible hypothesis is that such ad-hoc, context-dependent classifications are circumstantially created using dynamic mechanisms that involve high dimensional, distributed, vector representation of concept spaces. This would be in agreement with neurolinguistic evidence from functional brain imaging studies of the human semantic memory which suggest the existence of distributed feature networks for the representation of concepts (Martin and Chao 2001) and speaks to the

validity of approaches that have the potential to represent dynamic associative concept spaces in an unsupervised fashion, such as:

- 2.8.2 Markov models and n-gram models:
  - 2.8.2.1 PPM (Partial Phrase Matching) data compression
- 2.8.3 Unsupervised language acquisition approaches:
  - 2.8.3.1 Unsupervised morphosemantic decomposition models
  - 2.8.3.2 Grammar induction and segmentation models
- 2.8.4 Latent semantic indexing (LSI)
- 2.8.5 Formal concept analysis (FCA)
- 2.8.6 Connectionist and associative memory models:
  - 2.8.6.1 The Sparse Distributed Memory (SDM)
  - 2.8.6.2 The Self Organizing Map (SOM)

Many of the approaches listed above have the potential to represent dynamic, multidimensional concept spaces where features can vary in importance, evolve or change, accounting for many possible classifications and subtle variations of concept meanings. Such variations could account even for novel, less plausible or potentially humorous meanings that have the power to evoke laughter. Yet, most importantly, the approaches referenced above have the potential to attain the kind of similarity-based retrieval that is needed for advanced information processing.

The dynamicity of concept spaces may offer at least one of the reasons why fixed classification schemes, highly structured semantic representation schemas (e.g. fixed knowledge frames, semantic networks, ontologies), controlled terminology systems or open domain ontologies have failed to capture concept semantics in a way that provides

richness, dynamicity and reusability and have not turned out satisfactory in the long run. It may also explain why existing lexical databases based on carefully handcrafted knowledge, such as WordNet (Miller 1995) while useful for certain tasks, are often said to contain either too fine-grained or too coarse-grained, "static" semantic information (Kintsch 2001). In information intensive domains like medicine, concept space dynamicity may account for the reason why the development of a universal (i.e., one size fits all) clinical terminology system is so difficult (Rector 1999).

2.8.1.1 Scope

Much of the research on concept space representation was historically been directed towards the abstract side of the knowledge spectrum in the area of formal language descriptions. Recently, the emphasis is on approaches in the natural language domain. This is most likely due to the availability of computing power that makes applications feasible as well as to the relative maturity of certain fields of research such as natural language processing.

Reviewing all approaches to concept space representation is beyond the scope of this dissertation. The knowledge spectrum framework shows that human knowledge representations ranges from simple, very abstract symbols to formal grammars and formal language to natural languages to rich imagery, video clips perhaps ending with rich simulations of reality. The sheer multitude of computational approaches makes a comprehensive review, comparison and mapping onto the knowledge spectrum, a very difficult undertaking. Therefore, in the following only a limited subset of approaches will be reviewed in detail, namely those that aim at representing *dynamic concept spaces* in an *unsupervised* or *semi-supervised* manner.

2.8.2 Markov models and n-gram models

Markov models and n-gram models are the workhorses of computational linguistics, two equivalent approaches that have been used and are still being used extensively in state of the art data compression algorithms and in language processing approaches ranging from statistical language modelling to part of speech tagging, speech recognition and machine

translation (Manning and Schütze 1999). The distinction between the two seems to lie only in the manner they are described and used. Most often, Markov models are likened to non-deterministic finite state automata (FSA) and *structurally* represented as such, using directed graphs with vertices and edges labelled with transition probabilities. N-gram models, on the other hand, are usually referred to as feature spaces and most often described through the probability theory that underlies their *functionality*. The two models seem to be the embodiment of the very same approach in which the description of the former includes some *structural* details (e.g., directed graphs) while in the descriptions of the latter the emphasis is mostly on the *functionality*, leaving structural and implementation details transparent.

Because they are similar, Markov and n-gram models share identical objectives and assumptions. They both aim at predicting a certain event (e.g., occurrence of a certain word in a text) given the history of prior events (e.g., a context of a few words occurring just before) making the assumptions (i.e., Markov assumptions) that the history is limited and that the statistical properties of the data are time invariant (or stationary, i.e., that the data describes an *ergodic process*) (Manning and Schütze 1999). Formally, predicting an event e_n based on the history of *n-1* prior events $e_1, e_2, \ldots, e_{n-1}$ is equivalent to estimating the conditional probability function (Manning and Schütze 1999)[16]:

$$P(e_n | e_1, e_2, \ldots, e_{n-1}) = \frac{P(e_1, e_2, \ldots, e_n)}{P(e_1, e_2, \ldots, e_{n-1})}$$

Because this function depends completely on the probability distributions of individual n-grams, the objective of n-grams and Markov models is reduced to the appropriate estimation and representation of the probability distributions of individual n-grams.

The equivalence of the DDAM with n-gram and Markov models is evident upon inspecting the directed graph in Figure 35. This graph is isomorphic to that in Figure 21 and which contains trivial DDAM representations of weekday names. The only

---

[16] See Chapter 6.

distinction between the two is that in Figure 35, intercalated nodes carry transition probabilities identical to bi-gram or 1st order Markov model conditional probabilities. For example, because the character *d* appears 8 times, 7 of which precede the character *a*, the conditional probability of *a* given *d* is $P(a|d)=\frac{P(da)}{P(d)}=\frac{7}{8}=0.88$ while the conditional probability of a *n* given a *d* is $P(n|d)=\frac{P(dn)}{P(d)}=\frac{1}{8}=0.12$. The sum of the two conditional probabilities is 1.0 since *a* and *n* are the only characters that can precede *d*, in the context of the weekday names.

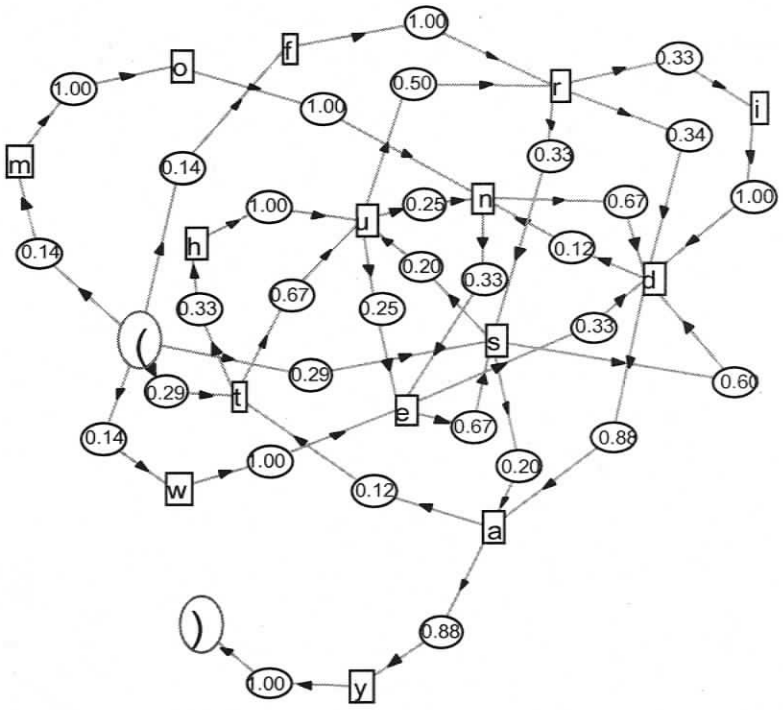


Figure 35. DDAM model induced from and representing trivially the weekday names, demonstrating structural and functional equivalence with bi-gram or first order Markov models; the start node is denoted "(", the stop node ")", the rectangular nodes correspond to characters and the oval nodes to transitions probabilities

In their most generic implementations, Markov and n-gram models have a fixed architecture whose complexity does not exceed 3rd order. This leads to a limited amount of context that they can capture, a necessary limitation because of the growth in number

of parameters which is exponential in the order of the model. For example, a four-gram model (3rd order Markov model) built on an estimated vocabulary of 20,000 words, in a matrix implementation, would require $20,000^3 \times 19,999 = 1.6 \times 10^{17}$ elements in order to store all its parameters (Manning and Schütze 1999). In addition, and extremely important in the context of this dissertation, many parameters are zero due to the sparseness of natural language where most words are never going to be part of the 3rd order history of all other words in a vocabulary.

The DDAM model, implemented on principles that can accommodate the sparseness of natural language (i.e., as a directed graph), overcomes the limitations of fixed context through an adaptive, variable order approach that depends on the statistical and information theoretic properties of data:

- Very ambiguous patterns which occur in multiple contexts are represented using higher order models which capture sufficient context to disambiguate them,
- Less ambiguous patterns, which occur consistently in the same contexts, are represented using lower order models which do not capture unnecessary context for their disambiguation.

From this perspective, the DDAM model is equivalent to an *adaptive n-gram model* or to an *adaptive variable order Markov model.*

To conclude, the hierarchical, adaptive approach in the DDAM model overcomes the high dimensionality and data sparseness problems of general Markov models and n-grams. This makes it similar to Markov models with variable architecture (Stolcke 1994; Guyon and Pereira 1995; Machler and Buhlmann 2002; Murphy 2002) and to hierarchical hidden Markov models (HHMMs) (Fine, Singer et al. 1998; Murphy 2002). The latter are relatively recent generalizations which have also been shown to be equivalent to dynamic Bayesian networks (DBN) (Murphy and Paskin 2001).

2.8.2.1 PPM (Partial Phrase Matching) data compression

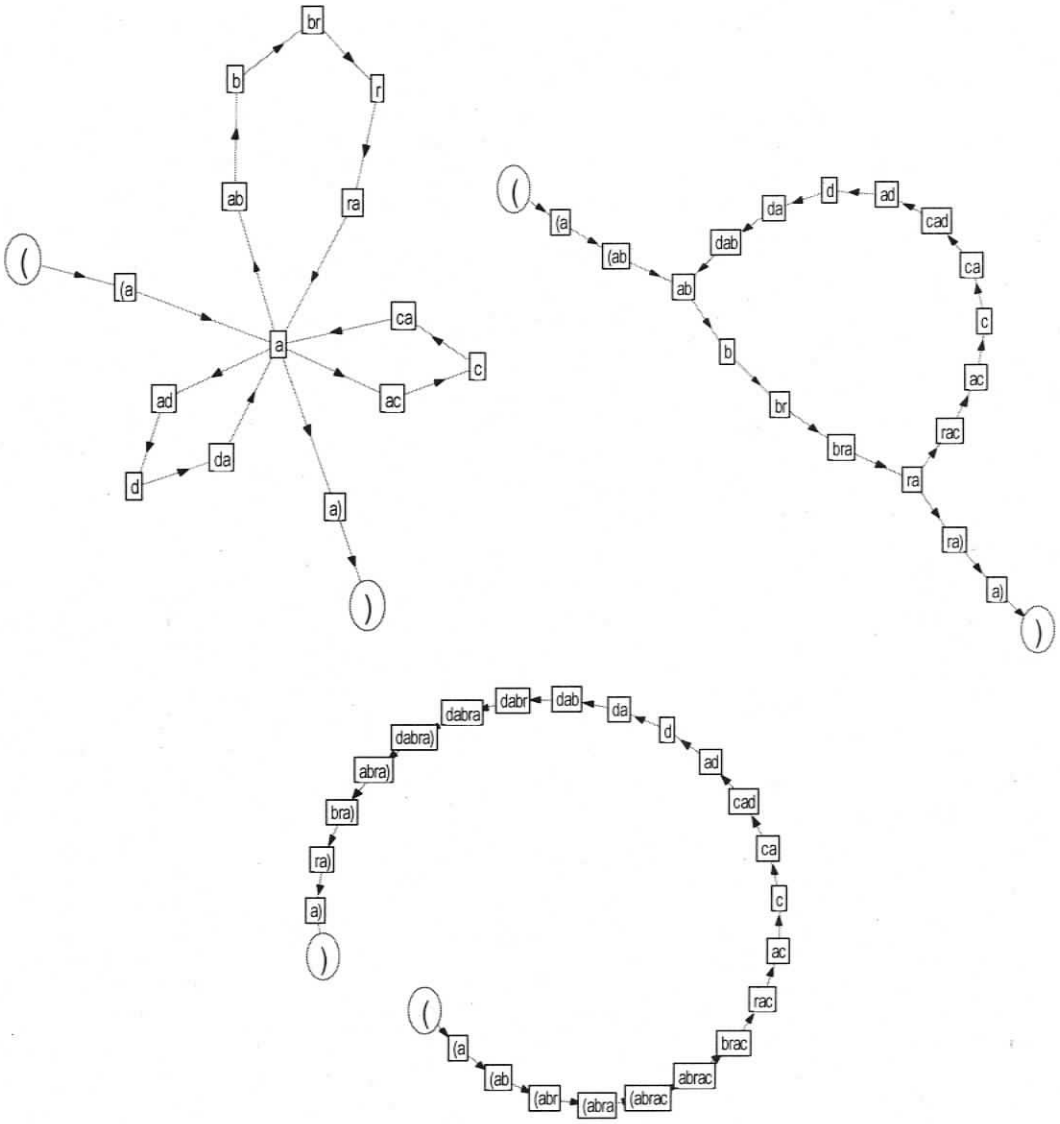
Though the general task of data compression is outside the scope of this dissertation, this section was included due to the existence of a particular data compression model, namely the PPM model (Cleary and Teahan 1998; Teahan 1998) with whom DDAM shares many similarities.

PPM is a family (e.g., PPMC, PPM*, etc.) of models that have defined the state of the art in data compression (Cleary and Teahan 1998) and are based on an adaptive arithmetic coding scheme whose probability estimations come from a context-sensitive statistical model of the data. In PPM, a variable number of characters preceding a certain symbol form the "context" of that character and can be used to predict it. This makes PPM essentially an adaptive n-gram (or Markov model) model whose context window length (i.e., history, model order) is variable, depending on the properties of the data. However, unlike DDAM which captures contexts on both directions (i.e., prior and next), PPM is a classic n-gram approach based only on prior context. In PPM the longest contexts are the most significant and hence used for prediction when available. This resembles well the DDAM model which also attempts to capture regularities which are as long as possible and hence as significant as possible.

Furthermore, the implementation of the PPM model is based on a suffix trie memory model (i.e. context trie) (Cleary and Teahan 1998). This is very similar to the DDAM model where a key design principle is a memory-based approach where the trade-off between space and time complexity is favouring the latter at the expense of the former. The fact that PPM is structurally similar to DDAM speaks to the validity of memory-based approaches to information processing and to the potential usefulness to data compression of models that aim at extracting and representing information as completely as possible:

> *PPM* encodes an explicit list of minimal strings that have occurred one or more times in the input. In a sense this is all the information that can be extracted from the input. This information is sufficient to implement all compression techniques that have the finite context property (that is, the probability estimate depends only on some finite length of preceding context). This includes a very wide range of compression techniques (Cleary and Teahan 1998).*

Finally, since the DDAM model is a generalization of trie memory models, it is natural to assume that DDAM is a generalization of the PPM model, structurally and functionally. For example, in Figure 36, the sequence "abracadabra" often used as an example to describe PPM models (Cleary and Teahan 1998), is represented in DDAM using variable order contexts in a generalized trie memory model from which, subsequently, a possible grammar is induced. The DDAM representation is structurally a generalization of the PPM suffix tries and the grammar induction is functionally equivalent to PPM segmentation capabilities.



		Rule							Expansion
3	→	a	b						a|b
2	→	3	r						ab|r
1	→	2	a						abr|a
4	→	(	1	c	a	d	1	)	(|abra|c|a|d|abra|)

Figure 36. DDAM representations comprising variable order contexts and an example of machine induced grammar for the sequence "abracadabra"

However, this seems as far as the similarities between DDAM and PPM go for now. The fact that PPM was used mostly in data compression and *supervised* information processing tasks (e.g., supervised word segmentation for Chinese language) renders a direct comparison with DDAM inappropriate for the time being.

2.8.3 Unsupervised language acquisition approaches

An incredible amount of work on unsupervised language acquisition approaches for general natural language has been done so far and reviewing it all is probably an impossible task. A fairly comprehensive but still not exhaustive collection of many approaches is surveyed in (Clark 2001).

2.8.3.1 Unsupervised morphosemantic decomposition models

The earliest account for the unsupervised learning of general natural language morphology (Harris 1955) is presented in detail together with an excellent review of other existing approaches by Goldsmith (Goldsmith 2001) many of which fall under the MDL framework, including his own for which he reports a precision of 85.9% and recall of 90.4%. Additional work on morphological decomposition is reported in (Creutz and Lagus 2002) for English as well as for highly inflectional language such as Finish with segmentation accuracy results of around 50%. A notable innovative approach to morphosemantic decomposition is (Schone and Jurafsky 2000) where a semantic processing component similar to latent semantic indexing (LSI) is used to improve the segmentation.

Medical terminology lends itself well to supervised morphosemantic decomposition due to the relative stability of the medical morphemes and to the existence of comprehensive medical terminology systems such as UMLS (McCray, Razi et al. 1996). This may also explain why research on unsupervised morphosemantic decomposition for medical terminologies is virtually non-existent. However, existing limitations of morphosemantic parsing (Baud, Rassinoux et al. 1999) suggest that work on representational approaches in the area of unsupervised morphosemantic processing are still desirable. In particular, the facts that multilingual morpho-semantem lexicons are not yet completely available, that their construction is labour intensive and that rule-based approaches require laborious fine tuning, maintenance and upgrades in order to cope with unexpected situations, speak to the need for unsupervised and semi-supervised approaches to representation and processing. Though explored only to a limited extent in this dissertation, the application of the DDAM model to morphosemantic decomposition in biomedicine is of particular

interest especially because it could lead to the integration of information from various existing morphosemantic knowledge sources.

2.8.3.2 Grammar induction and segmentation models

The literature on grammar induction and text segmentation is too rich to allow a review of all approaches. A look at the chronology of the publications in Table 11 reveals the longstanding interest in these topics starting with the work of Z. Harris in 1955 (Harris 1955). The list also shows the predominance of n-gram models and minimum description length (MDL) approaches.

Reference	Model/Principle	Title
(Harris 1955)	Not reviewed	From phoneme to morpheme. Language, 31(2): 190–222
(Olivier 1968)	Not reviewed	Stochastic grammars and language acquisition mechanisms, Harvard University, PhD thesis
(Stolz 1965)	Not reviewed	A probabilistic procedure for grouping words into phrases. Language & Speech 8, 219-235.
(Wolff 1975)	MK10, MDL	An algorithm for the segmentation of an artificial language analogue. British J Psychology 66 (1), 79-90
(Wolff 1977)	MK10, MDL	The discovery of segments in natural language. British J Psychology 68, 97-106
(Wolff 1980)	MK10, MDL	Language acquisition and the discovery of phrase structure. Language and Speech 23 (3), 255-269).
(Wolff 1982)	SNPR, MDL	Language acquisition, data compression and generalisation. Language & Communication 2 (1), 57-89.
(Wolff 1988)	MK10, SNPR, MDL	Learning syntax and meanings through optimization and distributional analysis. In "Categories and Processes in Language Acquisition", Y. Levy, I. M. Schlesinger and M. D. S. Braine (Eds), Hillsdale, NJ: Lawrence Erlbaum
(Rapp, Zimmerman et al. 1994)	Not reviewed	The algorithmic complexity of neural spike trains increases during focal seizures. J of Neuroscience 14 (8), 4731-4739.
(Hutchens 1994)	MDL, entropy	Natural language grammatical inference. University of Western Australia, PhD thesis
(Nevill-Manning 1996)	**SEQUITUR, MDL**	**Inferring Sequential Structure. University of Waikato. PhD thesis**
(Marcken 1996)	NG-HMM, MDL	Unsupervised Language Acquisition. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. PhD thesis.
(Brent and Cartwright 1996)	DR (Distributional Regularity)	Distributional regularity and phonological constraints are useful for segmentation. Cognition 61: 93–125.
(Brent 1999)	MBDP, MDL, NG-HMM, DP	An efficient, probabilistically sound algorithm for segmentation and word discovery. Machine Learning Journal 34: 71-106.
(Vervoort 2000)	EMILE (Entity Modeling Intelligent Learning Engine)	Games, walks and Grammars. University of Amsterdam. PhD thesis.
(Kit 2000)	DLG (description length gain), DP	Unsupervised Lexical Learning as Inductive Inference, University of Sheffield. PhD thesis
(Schone 2001)	NG-HMM, DP, MLE	Toward Knowledge-Free Induction of Machine-Readable Dictionaries. Department of Computer Science, University of Colorado. PhD thesis.
(Goldsmith 2001)	LINGUISTICA, MDL	Unsupervised Learning of the Morphology of a Natural Language. Computational Linguistics 27(2): 153-198.
(Venkataraman 2001)	MBDP, NG-HMM, DP, MDL	A Statistical Model for Word Discovery in Transcribed Speech. Computational Linguistics 27(3): 352-372.
(Batchelder 2002)	Bootlex, NG-HMM, NG-HMM, MDL	Bootstrapping the lexicon: A computational model of infant speech segmentation. Cognition 83: 167-206.
(Van Zaanen 2002)	ABL (alignment based learning)	Bootstrapping Structure into Language: Alignment-Based Learning. Leeds, UK, University of Leeds. PhD thesis.
(Creutz and Lagus 2002)	MDL, ML (maximum likelihood)	Unsupervised discovery of morphemes. Workshop on Morphological and Phonological Learning of ACL'02. 21–30
(Hammerton 2003)	SOM (Self organizing Map)	Learning to segment speech with self-organising maps. Computational Linguistics in the Netherlands 2002. Selected Papers from the Thirteenth CLIN Meeting. T. GAUSTAD. Amsterdam/New York, NY. VII: 159.
(Solan, Horn et al. 2004)	**ADIOS (Automatic Distillation of Structure)**	**Unsupervised context sensitive language acquisition from a large corpus. Proc. 2003 Conf. on Neural Information Processing Systems (NIPS), MIT Press.**

Table 11. Non-exhaustive chronological list of unsupervised approaches and models for grammar induction and text segmentation (based on (Wolff 2004)) (MDL=Minimum Description Length, DP=Dynamic Programming, Viterbi search, MLE=Maximum Likelihood Estimation, NG-HMM=n-gram, Hidden Markov Model)

The few exceptions which do not explicitly advocate MDL as a guiding principle and are particularly relevant to DDAM, are (Hammerton 2003) where segmentation is based on self-organizing maps and (Van Zaanen 2002) and (Solan, Horn et al. 2004), which fall under the sequence alignment paradigm.

G. Wolff had a long track record and interest in the problems of text segmentation, grammar induction and was one of the early developers of chunking algorithms (i.e., MK10 and SNPR models) which were improvements of previously published methods such as that of Olivier (Olivier 1968). He has used MDL as a guiding principle (Wolff 1982) and has recently proposed a unifying view in form of his ICMAUS framework (information compression by multiple alignment, unification and search), SP theory and models (Wolff 2003), in which the sequence alignment paradigm is essential.

In his dissertation (Hutchens 1994) J. Hutchens has proposed a chunking model based on information theoretic principles. Without any prior knowledge about separators in a text, by monitoring the entropy level in an n-gram model, the algorithm was able to pick up chunks (e.g., words) based on the increase in entropy that occurred naturally around text separators such as blanks. Later Hutchens realized the similarities of his models with those of Wolff.

In his dissertation (Nevill-Manning 1996), C. Nevill-Manning has proposed an algorithm named SEQUITUR, which was successfully applied to a diversity of tasks. Though it shared similarities with Wolff's MK10 model, SEQUITUR was incremental and had a linear time and space complexity which made it an efficient and elegant approach that was an improvement over MK10.

By the same time as C. Nevill-Manning, C. deMarcken explored in his dissertation (Marcken 1996), linguistically plausible mechanisms for grammar induction and text segmentation models based on MDL principles. The fact that his algorithms required multiple passes through data is a major difference from SEQUITUR and DDAM whose processing is local to the current input. However, the approach of deMarcken is extremely relevant to DDAM from the perspective of its aim of finding increasingly larger patterns in a bottom-up fashion and of creating hierarchical representations of strings. With respect to performance, deMarcken reported segmentation precisions of over 95% (Marcken 1996), a truly remarkable result which seemed to have settled the

unsupervised text segmentation problem. However, on closer inspection of the evaluation methodology, it has been observed ((Schone 2001)[17] (Brent 1999)[18]) that there was a complete lack of commitment to a particular segmentation from a hierarchical representation. This has caused a re-estimation of the segmentation precision to about 17%, based on the probability that a particular segmentation out of several ones is the correct one.

Starting with the work on the DR (Distributional Regularity) (Brent and Cartwright 1996) and later MBDP (Model Based Dynamic Programming) (Brent 1999), M. Brent has set the state of the art in text segmentation and has inspired a whole family of unsupervised text segmentation models ((Kit 2000; Schone 2001; Venkataraman 2001; Batchelder 2002)) all based on the very same principles: MDL, n-grams and Viterbi search (dynamic programming). Nearly all of these models have been successfully used to approach a very specific task, namely modelling speech segmentation and child language acquisition for which the estimated average word segmentation precision and recall ranged between 65% and 80% respectively. To attain these results, MBDP models have been specifically optimized (e.g., MBDP "makes very good use of sentence boundaries [...]" (Brent and Tao 2001)) to work best on phonetic transcriptions of spoken language such as those in the CHILDES corpus (MacWhinney 2000) containing very short utterances with many repetitions. However, this high specificity has rendered MBDP models less applicable to other types of data that contains longer utterances with limited repetition (Schone 2001)[19]. Finally, despite their success at modeling language acquisition in children, the fact that they are all based on the explicit, top down search in the solution space of the representations that satisfy the MDL principle, makes MBDP family of algorithms implausible biologically when compared to self-organizing map approaches or to models which construct a bottom up, hierarchical representation of data (e.g., deMarcken's model, SEQUITUR and DDAM).

---

[17] Page 54
[18] Page 43
[19] Page 58

EMILE (Entity Modeling Intelligent Learning Engine) was originally a supervised grammar induction model first described in P. Adriaans' dissertation (Adriaans 1992). Its adaptation to the unsupervised grammar induction paradigm was achieved by Vervoort (Vervoort 2000).

Van Zaanen explored in his dissertation (Van Zaanen 2002) the acquisition of structure from sequences. However his work was done in paradigm of alignment based learning (ABL), a relatively new approach, highly relevant to bioinformatics and natural language processing. In the natural language processing realm the principles of ABL can be found in various equivalent formulations and the structures relevant to sequence alignment are referred to in various ways such as, for example, *sausage graphs*, *word lattices*, *lexical chains*, etc. (Barzilay and Lee 2002; Barzilay and Lee 2003). As argued earlier, the DDAM model also can be reformulated and regarded in the sequence alignment paradigm.

Finally, one of the most recent models of inducing structure from unstructured data is ADIOS (Automatic Distillation of Structure) (Solan, Horn et al. 2004; Edelman, Solan et al. 2005). Besides general similarities, the relevance of ADIOS to DDAM is also significant in the specific aim at creating hierarchical structures in a bottom up fashion and in the overall similarity of learning with sequence alignment procedures.

Of the multitude of reviewed models, the focus in this dissertation is on those which have been particularly successful in solving a certain task and for which enough information was available (e.g., literature background, software simulations, results) to enable comparison with the DDAM model. As a result, the primary models chosen for direct comparison with DDAM are SEQUITUR (Nevill-Manning 1996) and ADIOS (Solan, Horn et al. 2004; Edelman, Solan et al. 2005)).

2.8.3.3 SEQUITUR

SEQUITUR (Nevill-Manning 1996; Nevill-Manning and Witten 1997; Nevill-Manning and Witten 1997) is a deterministic (i.e., represents data completely), online, one pass algorithm which creates hierarchical structures from sequential data. SEQUITUR has a

linear temporal and spatial complexity, and essentially, performs a form of grammar induction that is limited by two constraints:

1. the digram uniqueness constraint, and

2. the rule utility constraint.

The first constraint has to do with making sure that every sequence of two symbols (i.e., a digram) present in the input (including rewritten input) is unique. Upon detecting any digram repetition (i.e., at most twice), SEQUITUR replaces both diagrams with a non-terminal symbol and continues processing of the rewritten sequence. This is a recursive constraint and applies also to the rewritten sequence, which subsequently may exhibit new digram repetitions comprising various combinations of terminal and non-terminal symbols. This constraint results in a processing mechanism that is able to capture what was defined earlier, in the context of algorithmic information theory, as *significant patterns*. This makes SEQUITUR very relevant to DDAM which also aims at capturing regularities which are as long as possible and hence as significant as possible, regardless whether they might be very rare, i.e., occurring only twice.

The second constraint has the purpose of reducing the resulting grammar and results in the deletion of any single rule that is not used more than once, followed by copying the content of that rule in the appropriate places, a process which results in the creation of new, longer rules.

For example, applying SEQUITUR to the sequence in the example in Table 8 used to make a point about dimensionality reduction, results in the grammar in Table 12 written in *Chomsky's normal form* (i.e., using rules of the form A→ B C) and which corresponds to a hierarchical representation of the original string.

1	→	3	6									abcdef
2	→	6	3									cdefab
3	→	a	b									ab
4	→	e	f									ef
5	→	2	7									cdefabcd
6	→	7	4									cdef
7	→	c	d									cd
S	→	1	2	3	4	5	5	4	4	4	1	abcdef | cdefab | ab | ef | cdefabcd | cdefabcd | ef | ef | ef | abcdef

Table 12. SEQUITUR induced grammar for abcdefcdefababefcdefabcdcdefabcdefefefabcdef

SEQUITUR was applied to a variety of both artificial and natural sequences, textual data, information retrieval, DNA sequences where it was proven to be more effective than other general-purpose algorithms (Nevill-Manning and Witten 1997).

The strengths of SEQUITUR, i.e., its incremental nature and linear complexity, are also the source of its shortcomings, especially in the case of natural data whose algorithmic complexity may require multiple passes or representations that go beyond the limit of a linear function of the input size. A greedy algorithm is defined as an algorithm whose decisions are based on local, limited information that may render them globally non-optimal. Therefore, the necessarily greedy approach of SEQUITUR causes it to exhibit what appear to be local minima problems (Hopkins 1999). Concretely, as demonstrated in Chapter 4, this behaviour is caused by the premature commitment of SEQUITUR to certain rules and patterns, with no chance to reconsider or change them subsequently, since any incremental (i.e., online) algorithm is allowed only one-pass though the data. Essentially, this limitation manifests in the impossibility to capture certain regularities present in the data.

As will also be shown later in Chapter 4, by contrast, the DDAM model is able to overcome these limitations but this is possible only through the additional temporal and spatial complexity of the model, which, in the worst case, is a quadratic function. However in the case of highly sparse data, the temporal and spatial complexity of the DDAM model is very likely to become a linear function of the input (i.e., linear complexity).

2.8.3.4 ADIOS

ADIOS (Automatic Distillation Of Structure) (Solan, E. Ruppin et al. 2003; Pedersen, Edelman et al. 2004; Solan, Horn et al. 2004; Edelman, Solan et al. 2005) is a non-deterministic (i.e., input data is not represented exactly), grammar induction model able to induce, represent and generalize complex regularities existent in data through three principles (Solan, E. Ruppin et al. 2003):

1) Detection of pattern significance by probabilistic means,

2) Context sensitive generalization, and

3) Recursive construction of complex patterns.

The representation in ADIOS is in the form of a directed graph in which sequences are aligned. Initially this graph is highly redundant but subsequently, upon the identification of significant patterns, representations are updated to make use of the detected patterns and hence to reduce the redundancy. The first principle, essentially, equates to the mechanism used to detect the significance of a pattern. Because the detection is heuristic in nature, the performance, flexibility and robustness of the algorithm seems to depend heavily on the appropriate tune-up of the parameters (e.g., the p parameter) that govern this mechanism.

The second principle stands behind the ADIOS capability of creating equivalence classes or sets of items in the input data, on the basis that items in an equivalence set must share the same context. This results in the creation of classes, which are highly specific to a particular context. This subsequently allows combinatorial generalizations to new sequences which share similarities to original ones, on the basis that patterns are constrained to their appropriate contexts. Essentially, this principle stands behind the possibility to generalize and generate data in a context-sensitive fashion.

The third principle has the purpose of capturing long-range dependencies and accounting for patterns which are recursive (i.e., recursive patterns).

ADIOS also has provisions for a motif extraction processing mode (MEX) that can be used in order to process sequences which are not tokenized (i.e., segmented in individual chunks named tokens), such as DNA and aminoacid sequences.

Functionally and structurally ADIOS is considered superior to Variable Order Markov (VOM) models (Guyon and Pereira 1995) and equivalent to Tree Adjoining Grammars (TAG) which are situated at the interface between Context Free and Context Sensitive Languages in Chomsky's hierarchy (Pedersen, Edelman et al. 2004). From a methodological standpoint ADIOS is also considered to abide to the principles of learning and representation of language advocated by cognitive science research (Pedersen, Edelman et al. 2004).

ADIOS was also applied to a variety of sequence processing problems with notable results on a 300,000 subset of CHILDES collection (MacWhinney 2000) comprising 1.3 million tokens that took over 14 days of processing and which has resulted in the identification of 3400 patterns and 3200 equivalence classes which have been subjectively judged to have significant potential to yield semantically appropriate equivalence sets (Pedersen, Edelman et al. 2004). When the model has subsequently been subjected to the grammaticality judgment test component of CASL (Comprehensive Assessment of Spoken Language), a test used widely in USA to assess language comprehension in children, it scored as a child in the age interval 7-0 and 7-2. In a similar setup but for an ESL (English as a Second Language) grammaticality test ADIOS has scored about 60%, the average human being 65% (Pedersen, Edelman et al. 2004).

The major drawbacks of ADIOS reside in its markedly heuristic nature and heavy dependence on the appropriate tune-up of its parameters. Unlike SEQUITUR and DDAM, this causes unexpected results, especially when data is sparse. For example, when applied to the same sequence (which had to be appropriately preprocessed and presented in multiple copies) the resulting grammar captures only a few significant patterns and is significantly different from those of SEQUITUR and DDAM.

	Rule	Rewrite
P9	(a,b,c,d,e,f)	abcdef
P10	(e,f)	ef
P11	(a,b)	ab
P12	(c,d)	cd
P13	(P9, P12, P10, P11, P11, P10, P12, P10, P11, P12, P12, P10, P9, P10, P10)	abcdef | cd | ef | ab | ab | ef | cd | ef | ab | cd | cd | ef | abcdef | ef | ef

Table 13. ADIOS induced grammar for a collection of 9 copies of the sequence "a b c d e f c d e f a b a b e f c d e f a b c d c d e f a b c d e f e f e f a b c d e f"

Because it tries to find patterns in a greedy manner, ADIOS is affected by the particular ordering of the input data, a symptom that it may suffer from local minima problems.

As will be shown later, by contrast, DDAM is a deterministic approach that uses no heuristics and has only two, semantically clear parameters. Its outcome is not affected by the ordering of the input data and is less prone to exhibit local minima problems.

2.8.4 Latent semantic indexing (LSI)

LSI (Landauer and Dumais 1997; Hofmann 2001; Kintsch 2001; Brants, Chen et al. 2002; Kintsch 2002) is a method of dimensionality reduction (a projection method) which aims at capturing semantics of documents and improving information retrieval systems, in particular improving their recall. LSI belongs to the same group of methods as principal components analysis (PCA) and is based on the statistical technique of singular value decomposition (SVD) that is applied to a document-feature matrix (e.g., documents vs. words, phrases). The features (e.g., words, phrases) are automatically extracted from documents and each document is represented as a feature vector in a multidimensional conceptual space.

The relevance of LSI to DDAM is in the context of information retrieval and consists of the possibility to capture indirect associations between features. This translates into the retrieval of documents that do not necessarily need to contain the exact query, but may contain other features, which the query is strongly associated with. For example, LSI can potentially account for the retrieval of a document containing the phrase "throat infection" given the query "streptococcus" even though the document does not contain the query as such and the association between the two is not explicitly encoded in the

retrieval system. This is possible only because of the strong association between the query and the concept in the document.

The most important problem with LSI is that, being a dimensionality reduction method, it leads to an important loss of information in representations. Secondly LSI has the shortcoming of not being dynamic. Once created, the LSI semantic space is rather static and adding representations to it implies rerunning computationally expensive algorithms, for all representations.

In the case of DDAM, increasing the recall radius allows for weaker associations that could increase retrieval recall but may lose precision. However, unlike DDAM, where the sequential nature of textual data is preserved and associations are between proximal patterns, LSI is capable of discovering complex (and sometimes too complex) indirect associations and dependencies between distal words that may not even occur in the same sentence or may potentially occur in reverse order since LSA is oblivious to the sequence of words in a document. In evaluations, this capability may translate in increased recall values of IR systems but could potentially lead to decreased precision values due to spurious associations (Manning and Schütze 1999).

Because LSI has the potential to discover similarities between documents that sometimes are difficult to explain, its power is somewhat misdirected towards being able to capture semantic nuances and variations that may be too subtle even to humans. In a nutshell, LSI could be considered, to a certain extent, too powerful a method.

2.8.5 Formal concept analysis (FCA)

Formal Concept Analysis is a subfield of Applied Mathematics and represents the mathematization of *concepts*, *concept hierarchies* (Wille 2005) and *type hierarchies*. As a formal *theory of concepts* FCA seems to be the missing link that may "function in the sense of *transdisciplinary mathematics*, i.e., allowing mathematical thought to aggregate with other ways of thinking and thereby supporting human thought and action" (Wille

2005). In other words, FCA is considered the missing piece that could connect theoretical and applied research. The fundamental thesis of FCA (Wille 2005) states the following:

> *The aim and meaning of Formal Concept Analysis as mathematical theory of concepts and concept hierarchies is to support the rational communication of humans by mathematically developing appropriate conceptual structures which can be logically activated.*

An excellent introduction to FCA is (Wille 2005). Central to FCA theory are *objects* and *attributes* and the notion of *formal context* first introduced by R. Wille in 1982 (Wille 1982) which defines formally and connects the notions of *concept* and *context*. The mathematical structures used by FCA are the *partial order sets*, in particular the *lattice structure*. The partial ordered set structure of formal contexts is established by the Basic Theorem on Concept Lattices (Wille 2005). A formal context is best visualized as a *cross table* (Table 14) and a concept lattice as a *graph* (Figure 37). For example, the context of the grammar induction and segmentation models reviewed earlier in this chapter (i.e., ADIOS, SEQUITUR, SP, MBDP, DDAM) as well as FCA itself, together with some of their properties (MDL – minimum description length, ABL – alignment based learning, BOTTOM-UP – bottom up processing, TOP-DOWN – top down processing, HIERARCHY – hierarchical structure, GREEDY – greedy model) can be visualized as the cross table in Table 14 where models are represented as rows and their attributes as columns.

	MDL	ABL	NH-HMM	POSET	BOTTOM-UP	TOP-DOWN	HIERARCHY	GREEDY
FCA	.	.	.	×	.	.	×	.
MBDP	×	.	×	.	.	×	.	.
SP	×	×	.	.	.	.	.	.
ADIOS	.	×	.	.	×	.	×	×
SEQUITUR	.	.	×	×	×	.	×	×
DDAM	.	×	×	×	×	×	×	.

Table 14. Cross table representing the formal context of models and their properties

The same information as in Table 14 can be represented as the graph in Figure 37 where models are defined through direct or indirect links to attribute nodes.

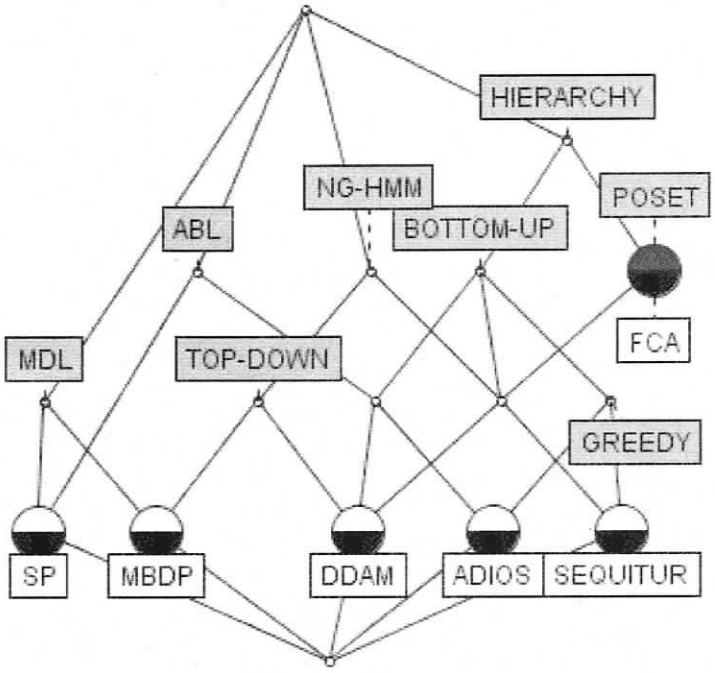


Figure 37. Concept lattice representing the formal context of models and their properties visually

The conceptual information conveyed by concept lattices is powerful and allows for the immediate visualization of complex relations between concepts. For example, it is clearly shown that the DDAM model is more related to ADIOS and SEQUITUR as well as with FCA, that DDAM is both a BOTTOM-UP and TOP-DOWN approach and is not a GREEDY model like ADIOS and SEQUITUR but is related to ABL, like the ADIOS and SP model which is based on MDL principle, like MDBP. Finally the fact that MBDP, DDAM and SEQUITUR are all versions of n-gram or Markov models is also evident from the diagram.

The representational power of FCA has led to its application to a multitude of areas including but not being limited to: computational linguistics (Priss 2005; Valverde-Albacete 2005), information retrieval (Cigarran, Peitas et al. 2005) and text mining (Carpineto and Romano 2005), data mining (Lakhal and Stumme 2005), knowledge processing and ontology development (Joslyn 2004; Joslyn and Mniszewski 2004), conceptual information systems (Becker and Correia 2005; Ducrou and Eklund 2005), help systems (Eklund and Wormuth 2005), generic pattern mining (Zaki, Parimi et al.

2005), economics (Wille 2005), software engineering (Cole and Becker 2005; Godin and Valtchev 2005; Hesse and Tilley 2005; Snelting 2005; Tilley, Cole et al. 2005), analysis of aeronautical incidents (Maille, Statler et al. 2005) and case based reasoning (Belèn and Pedro 2001; Belèn, Pablo et al. 2003).

The problems associated with concept lattices stem from their representational power, which allows them to exhaustively represent all concepts embedded in a dataset. The sheer size of concept lattice built on real datasets makes their spatial complexity too big to be handled by current technology. This has caused the development of approaches that aim at pruning concept lattices of nodes which do not meet certain criteria (Ventos and Soldano 2005).

The relevance of FCA to DDAM arises from the similarity of the underlying mathematical concepts that the two share, namely that of *partial order set* (poset). Formally, a poset is a base set together with a *reflexive*, *antisymetric* and *transitive* binary relation (to be defined formally in Chapter 3) on that base set. The highly abstract nature of this definition allows posets to be constructed in various ways depending on how one chooses the base set and the binary relation that defines the partial order on that set. The most common choice in FCA is to choose a base set as a collection of smaller unordered sets and the set inclusion as the binary relation. For example, a base set X containing the sets $\{\ \}$ (i.e., the empty set), $\{a,c\}$ and $\{a,b,c,d\}$ (i.e., $X=\{\{\ \},\{a,c\},\{a,b,c,d\}\}$) together with the set inclusion relation $\subseteq$ could form the partial order set $P(X,\subseteq)$ where $\{\ \}\subseteq\{a,b,c,d\}$, $\{\ \}\subseteq\{a,c\}$, $\{a,c\}\subseteq\{a,b,c,d\}$, $\{a,c\}\subseteq\{a,c\}$ and $\{a,b,c,d\}\subseteq\{a,b,c,d\}$.

By contrast, the DDAM partial order sets are finite word posets (Erdos, Sziklai et al. 2001) based on the more restricted binary relation "is subsequence of" which implies the use of a base set which is a collection of strings (i.e., a language). The fact that strings are sets of characters where order is important (i.e., strings are ordered sets, tuples) implies that the DDAM partial order set has an inherent ability to capture and represent the sequential data and, at the same time, to restrict the complexity of the structure. For

example, a base set of strings $X=\{\lambda,ac,abcd\}$ (lambda denotes the empty string), together with the binary relation "is subsequence of" denoted by $\prec$, could form the poset $P(X,\prec)$ where $\lambda \prec ac$, $\lambda \prec abcd$, $ac \prec ac$, $abcd \prec abcd$ but $ac \not\prec abcd$, unlike in the case of previous example.

To summarize, the high representational power of FCA is also the source of its problems which are largely related to the spatial complexity of the posets. Because they are based on unrestricted binary relations such as the set inclusion, FCA models are powerful but oblivious to the sequential nature of many real world datasets. The choice of a restricted partial order relation in case of DDAM overcomes some of the spatial complexity while maintaining the possibility to represent data deterministically, without loss of information.

2.8.6 Connectionist and associative memory models

A discussion of unsupervised concept space representation models and approaches would not be complete without reviewing connectionist and associative memory models, even though many of them fall under the supervised learning paradigm. The focus here is on connectionist information processing models that work on parallel and distributed processing (PDP) principles (Rumelhart, Hinton et al. 1986), and which are able to represent and recall certain classes of patterns (or regularities) present in input data, in an on-line and unsupervised manner. Secondarily, the focus is on associative memory models that attain representations which are hierarchical and compositional in nature.

A comprehensive search and review of connectionist models relevant to this dissertation has yielded a number of references in the order of hundreds, most in the area of Cognitive Science. Narrowing down to associative memory models has revealed about forty references and about two dozens researchers and associative memory models which are relevant (Murdock; Kanerva 1988; Koberle 1989; Pollack 1990; Murdock 1993; Murdock 1993; Flachs and Flynn 1994; Greene 1994; Krikelis and Weems 1994; Mcnamara and Diwadkar 1996; Okada 1996; Fan and Wang 1997; Gough 1997;

Manevitz and Zemach 1997; Reimann 1998; Vogel 1998; Bassi 1999; Negishi, Bullock et al. 1999; Pomi Brea and Mizraji 1999; Sommer and Palm 1999; Bassi 2000; Bolle, Dominguez et al. 2000; Hirahara, Oka et al. 2000; Hirahara, Oka et al. 2000; Ma and Isahara 2000; Greene and Tussing 2001; Kohonen 2001; Sandberg, Lansner et al. 2001; Sommer and Wennekers 2001; Wichert 2002; Anwar and Franklin 2003; Federici 2003; McElree, Foraker et al. 2003; Murphy 2003; Nomura, Aoyagi et al. 2003; Sandberg, Tegner et al. 2003). Reviewing and comparing them in detail with the DDAM model is a considerable endeavour that is outside the reasonable workload that must characterize a dissertation. Therefore this work is left for the future but references are kept in order to allow this author and other interested readers to easily follow up on this research path. As a consequence of this decision, this review of associative memory models was drastically (and somewhat artificially) reduced to only two models with which DDAM shares the greatest similarities: Kanerva's Sparse Distributed Memory (SDM) and Kohonen's Self Organizing Map (SOM).

2.8.6.1 The Sparse Distributed Memory (SDM)

One of the most relevant publications to this dissertation is Sparse Distributed Memory (SDM), a monograph of P. Kanerva (Kanerva 1988) where the memory model with the same name (SDM) is described. However, even though SDM was physically realized beyond the form of a theoretical model (Nordström 1991; Hamalainen, Klapuri et al. 1997; Anwar and Franklin 2003) and subsequently improved by other researchers (Hely, Willshaw et al. 1999), the relevance of Kanerva's work to the DDAM model has remained mostly theoretical and conceptual in nature and has manifested itself throughout this manuscript in form of theoretical principles that apply generally to any associative memory model. Three such fundamental principles are:

- The explicit recognition of the sparse nature of representations which impacts fundamentally the design of associative memory models,
- The requirement for associative recall capabilities that work on a similarity basis and are able to retrieve the best matches within a given bit radius from a query, and

- The storage of sequential data in the memory in form of k-fold transitions which subsequently can be used in prediction.

The last principle essentially describes the equivalent of a Markov model and forms the basis of what appears to be nothing but case based reasoning:

> *We seek to answer the question of how information be stored in memory so that it can be retrieved later when the situation warrants it. This requires two things: that the present situation be recognized as being similar to some situation in the past and that the consequences of that past situation be retrieved (Kanerva 1988)*[20].

The mathematical foundation of the SDM model, not surprisingly, gravitates around the notion of a *n*-dimensional binary space (i.e., Hamming space) denoted as $\{0,1\}^n$. As with all memory models, SDM supports read and write operations. However, unlike computer random access memories, SDM generalizes representation spaces to extremely large ones such as $\{0,1\}^{1,000}$. Given the immense number of possible addresses (i.e., 2^{1000} in the earlier example) and hence the impossibility to physically construct such a memory, the *hard locations* (i.e., the locations where data can effectively be written) are set to a reasonable number, such as for example, 10 million, but their addresses are distributed randomly throughout the $\{0,1\}^{1,000}$ space. Storing data in SDM implies distributing it over all hard locations whose address have a similarity to that data (i.e. the locations which are within a specified Hamming distance, or bit radius from the data). This approach essentially leads to a distributed (Kanerva 1997) and, at the same time, extremely sparse representation of information (Kanerva 1993).

Roughly, retrieving data from SDM implies providing a piece of the data and reading back all the hard locations whose addresses are within a certain "recall radius." While this mechanism is, to a certain extent, the same in the case of DDAM, this is as far as the similarity goes. Unlike DDAM, the SDM model brings the concept of distributed representations to a level which causes it to become non-deterministic. In addition, SDM is a static model where the number and addresses of the hard locations, once established,

---

[20] Chapter 8, page 79

do not change dynamically. However these very properties make the SDM model particularly resistant to noise and attractive to applications that show a particular fit with biological information processing models, also known to be non-deterministic and robust to destruction. An application that stands out is the modeling of the immune system (considered to share similar associative properties to SDM), such as, for example, the case of a lymphocyte that is able to recognize imprecisely (i.e., non-deterministically) a set of antigens which are similar (i.e., a best match problem) (Hart and Ross 2003).

2.8.6.2 The Self Organizing Map (SOM)

Another associative memory model relevant to DDAM is the Self Organizing Map proposed by Kohonen (Kohonen 2001), a stochastic model of artificial neural network which falls into the more general class of competitive connectionist models.

SOM has been successfully used in various domains including language processing tasks such as document clustering (Kaski 1997; Dittenbach, Rauber et al. 2002) and word clustering (Hodge and Austin 2002). Its capabilities have also been explored by this author by applying it to an unsupervised clustering and instance-based decision task (Pantazi, Kagolovsky et al. 2002).

The most attractive properties of SOM is the capability to create highly suggestive, visual representations and projections of multidimensional data sets. This makes it a useful tool for exploratory data analysis and a first step that should be taken when analyzing new data. This is so because besides giving a first impression about the data itself it can also point to potential data errors. But SOMs can also be used for similarity based retrieval and classification given the fact that a self-organizing map is able to place new patterns in the context of the known, correctly classified patterns by finding the closest match and assuming that the new pattern belongs to the same class as the match. This functionality was used to test a SOM classifier on the Wisconsin Breast Cancer (Blake and Merz; Mangasarian and Wolberg) dataset which contained 699 instances of cytological analysis of fine needle aspiration from breast tumours. Each case comprised 11 attributes: a case ID, cytology data (normalized, with values in the range 1-10) and a benign/malignant attribute (Table 15).

#	Attribute	Domain
ID	Sample ID code	integer
A1	Clump Thickness	1 - 10
A2	Uniformity of Cell Size	1 - 10
A3	Uniformity of Cell Shape	1 - 10
A4	Marginal Adhesion	1 - 10
A5	Single Epithelial Cell Size	1 - 10
A6	Bare Nuclei	1 - 10
A7	Bland Chromatin	1 - 10
A8	Normal Nucleoli	1 - 10
A9	Mitoses	1 - 10
A10	Class: 2 for benign, 4 for malignant	2, 4

Table 15. The attributes of the data set

Given the domain of variation of the values, the theoretical number of patterns in this analysis would have been 10^9, that is one billion patterns. The fraction of instances in the data set is only 699/1,000,000,000, which causes this problem space to be sparse since the number of theoretical points is very high compared to that of actual data points. The number of benign instances was 458 (65.52%) and the number of malignant instances is 241 (34.48%). Some sixteen cases (14 benign, 2 malignant) have been removed from the data because of missing values.

Upon performing the exploratory analysis, a zone of high homogeneity (i.e. cluster) was detected in the low-right corner of the map while low homogeneity characterized the rest of the map (Figure 38 left). Using a lower threshold, the clustering became more evident (Figure 38 right).

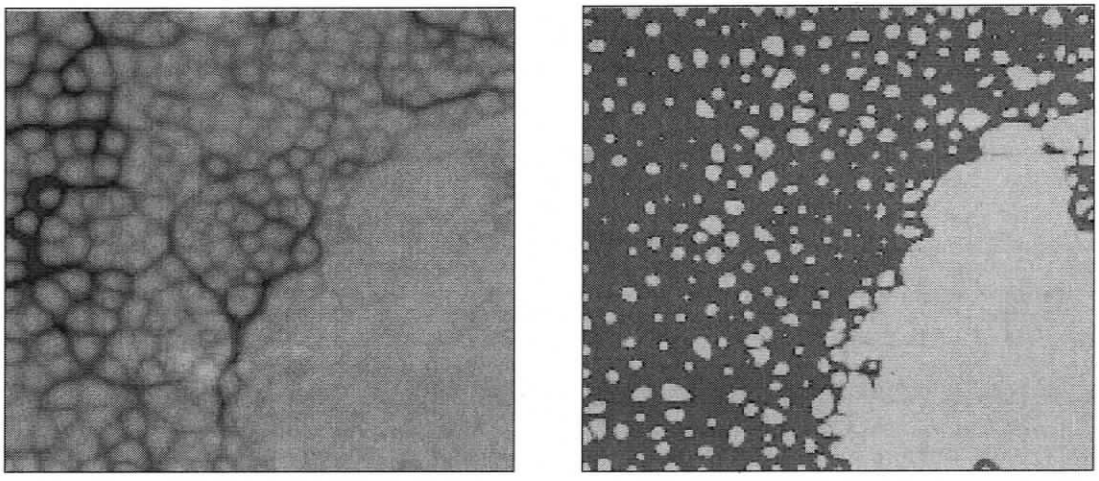


Figure 38. 180x180 points self-organizing maps of the dataset rendered using different clustering thresholds (a – left – higher clustering threshold, b – right – lower clustering threshold)

The information provided by the color-coding (Figure 39), indicate clearly that the benign cases are located mostly in the homogeneous cluster. This also indicated that differentiating beginning from malignant instances could be achieved only with a similarity-based classifier. The visualization also shows that the two classes exhibit a degree of overlap with each other, as some benign case patterns lie outside the benign case cluster and some malignant case patterns are very close or even inside the benign case cluster.

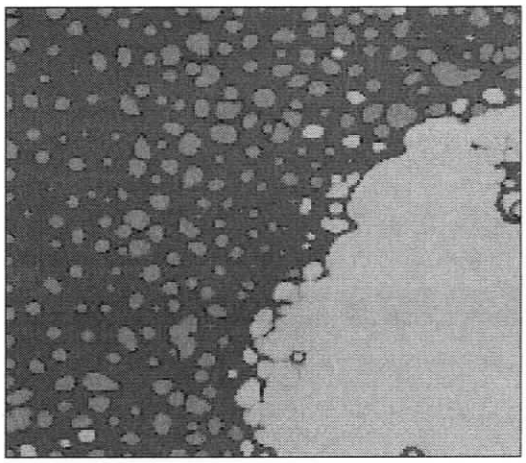


Figure 39. 180x180 points self-organizing maps rendered by color-coding on the benign/malignant class attribute (dark – malignant, light – benign)

For classification and evaluation purposes the dataset was randomly split into two subsets, one for training and one for testing, each of them containing 341 and 342 patterns, respectively. The training subset was used to create a new map and the testing subset was used to evaluate the classification capabilities of the map. For the 342 cases in the test subset, the linear distance based classifier accuracy was 95.32%, i.e. 326 patterns were correctly classified. Given the relatively homogeneous clustering of the benign case patterns, the high accuracy of such a simple classifier was no surprise and demonstrated the validity of the approach.

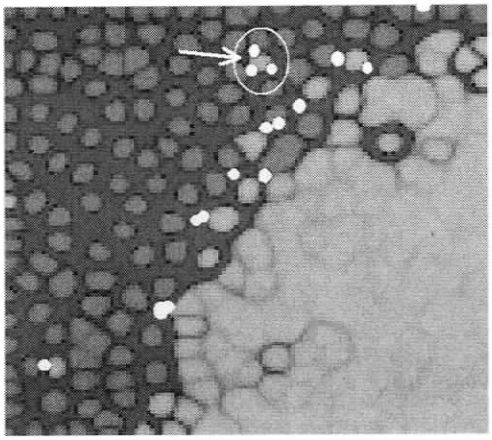


Figure 40. 120x120 points self-organizing maps rendered using the first half of the data set and color-coded (dark – malignant, light – benign); the misclassified patterns are marked by white dots

The sixteen misclassified cases, when plotted on the map, showed that most of the patterns were malignant cases (14) having as the closest match a benign case pattern lying either near the border of the benign case cluster (Figure 40) or far from the cluster. Analysis of the data for three of these misclassifications (indicated by the arrow Figure 40 and summarized in Table 16) showed that the three test patterns indeed have as the closest match a training pattern sitting far from the benign cases cluster and this strongly indicates a problem with the training pattern, which would needs to be reviewed.

Pattern	Class	ID	A1	A2	A3	A4	A5	A6	A7	A8	A9
Test pattern 1	M	837480	7	4	4	3	4	10	6	9	1
Test pattern 2	M	1171710	6	5	4	4	3	9	7	8	3
Test pattern 3	M	63375	9	1	2	6	4	10	7	7	2
Closest match (training pattern)	B	1213375	8	4	4	5	4	7	7	8	2

Table 16. Three misclassified patterns and their closest match (training pattern)

As it can be seen in Figure 40, this case is not singular and actually many of the misclassifications may be explained as being the results of outliers alone.

What was demonstrated here is the representational power of self-organizing maps and of associative memories in general when dealing with sparse spaces. But most importantly, the experiment has demonstrated the possibility to achieve complex information processing functions such as clustering and classification, only by means of representation and pattern matching. This fits very well the goals of dissertation and brings more evidence for the validity of case based reasoning as an approach to

information processing in general and in particular for the validity of proposed DDAM model which, in conjunction to force-directed graph layout optimization methods, appears to attain, to a certain extent, the structural and functional equivalence of SOMs. This conclusion is also supported by the observation (Frick, Ludwig et al. 1995) that force directed automatic layout algorithms on general graphs exhibit functional and structural similarities with SOMs.

2.8.7 Conclusions

This subsection was a review of the models and approaches to concept space representation to which the DDAM model is similar. The review shows that the DDAM model is very general and stands comparison with a multitude of models, most of which are highly specialized for solving one particular task (e.g., grammar induction, text segmentation). Equally important is the fact that none of the reviewed models appears to be able to match the breadth of functional capabilities of the DDAM model which has the potential to perform such seemingly disparate functions as grammar induction, sequence alignment, unsupervised classification and similarity based retrieval, at the same time. This evidence of the generality and validity of the DDAM model will be further subjected to empirical evaluations in Chapter 4.

Chapter 3 THE DETERMINISTIC DYNAMIC ASSOCIATIVE MEMORY (DDAM)

In this chapter the proposed associative memory model and processing algorithms are presented formally and their functionality demonstrated by means of examples.

3.1 FUNDAMENTAL DEFINITIONS

This section introduces fundamental mathematical concepts that describe the DDAM model, in form of definitions and pointers to additional material available in Appendix 1.

The definitions begin with the most fundamental and build upon each other in order to arrive at the definitions of increasingly complex DDAM structures. The material is structured in five subsections: "Set theory," "Binary relations," "The set theory of strings," "Partial order sets" and "Combinatorics." The basic notation and the most fundamental definitions cited from literature are formally given in Appendix 1. These definitions are only briefly, informally reviewed in this chapter while the original contributions are presented in complete detail.

3.1.1 Set theory

The set theory subsection starts with the most fundamental definitions, that of a *set* and *cardinality of a set*. These lead to defining two additional concepts, that of *multiset* and *cardinality of a multiset* that emphasize on the *multiplicity property of sets* which lies at the core of the DDAM model. Further, this leads to two new definitions, that of an *abstract set of a multiset* and that of a *compression set of a multiset* which are fundamental to and determine the formal notation of DDAM model components.

Definition 1. The *abstract set* of a multiset M, denoted as $\tilde{M}$ is a set obtained by ignoring the element multiplicity in M. This causes $|M| \geq |\tilde{M}|$. Because this operation loses information about how many multiple instances of an element are there in a multiset, the abstraction set of a multiset could be considered a lossy compression of that multiset. For example, $\{a,b,c\}$ is the abstraction of all $\{a,b,c\}$, $\{a,a,b,c\}$ and $\{b,b,a,a,a,c,c,c,c\}$.

Definition 2. The *compression set* of a multiset M, denoted as $\hat{M}$ is the set of pairs (x,n) where $x \in \tilde{M}$ and $n \in \mathbb{N}$ is the number of distinct instances of x in M. Because this operation does not lose any information about the how many multiple instances of an element are there in a multiset, the compression set of a multiset could be considered a lossless compression of that multiset. For example, the compression sets of $\{a,b,c\}$, $\{a,a,b,c\}$ and $\{b,b,a,a,a,c,c,c,c\}$ are $\{(a,1),(b,1),(c,1)\}$, $\{(a,2),(b,1),(c,1)\}$ and $\{(b,2),(a,3),(c,4)\}$ respectively. It follows that the abstract set of a multiset is isomorphic to the compression set of that multiset. It also follows that if $\hat{M}$ is the compression set of a multiset M then the sum of all instance counts in $\hat{M}$ is equal to the cardinality of M:

$$\sum_{(x,n)\in\hat{M}} n = |M|$$

3.1.2 Binary relations

This subsection is dedicated to fundamental definitions of *binary relations* and of their properties. All of these definitions are gleaned from literature and hence, are formally presented in Appendix 1. These definitions must precede and are necessary in the context of *partial order sets* whose formal definition depends on that of *partial order relation* which, in turn, depends on the definition of a *binary relation* and its properties. Therefore, the first definition is that of a *binary relation* on a set followed by the definitions of the *reflexivity*, *anti-symmetry* and *transitivity* properties. Then partial order relation on a set is defined followed by the definition of the *cover relation* as the *transitive reflexive reduction of a partial order*. The cover relation is important in the context of the visualization of DDAM models in form of graphs whose edges stand for the transitive reflexive reduction of the partial order.

3.1.3 The set theory of strings

This subsection is necessary to establish formally the definition of a string as well as of relations between strings. Some of the formal definitions are available in Appendix 1, starting with that of an *alphabet* and followed by that of a *list* (*sequence*, *n-tuple*) and those of a *string* and *length of a string*. Building upon the formal definition of a *language*, an extension is proposed in the form of the definition of a *multiset language* necessary in order to formalize the concept of a set of strings where repeating strings are allowed and accounted for.

Definition 3. A *multiset language* is a language in which multiplicity of strings matters and therefore duplicates or multiple instances of a string are considered distinct. For example, $\{\lambda, ab, bb, cd\}$ is a language over the alphabet $\{a,b,c,d\}$ and $\{\lambda, \lambda, ab, a, a, bb, bb, cd, a, cd\}$ is a multiset language over the same alphabet.

Following the definitions of string *concatenation* and that of the relations *is substring of* as well as the more restricted one of *is a proper substring of*, a new definition which lies at the core of the DDAM model is proposed.

Definition 4. A *substring multiset of a language* is the multiset of all substrings of all strings in a language or multiset language. Formally, if L is a language or multiset language over some alphabet Σ, then $M = \{x | x \text{ is a substring of } y, y \in L\}$. It follows that if $L = \{\ \}$ then $M = \{\ \}$ and that if $L = \{\lambda\}$ then $M = \{\lambda\}$ as well. For example, if $L = \{\lambda, ab, ab, abc\}$ is a multiset language over $\Sigma = \{a,b,c\}$ then $M = \{\lambda, ab, a, b, \lambda, ab, a, b, \lambda, abc, ab, bc, a, b, c, \lambda\}$. Written as a compression set, M is $\hat{M} = \{(a,3),(b,3),(c,1),(ab,3),(bc,1),(abc,1),(\lambda,4)\}$.

If M is the substring multiset of a multiset language L consisting of one string $x, x \in L$, then the cardinality of M, $|M|$ is:

$$|M| = \left(\sum_{i=1}^{\|x\|} i \right) + 1 = \frac{\|x\|(\|x\|+1)}{2} + 1$$

This result is based on the simple observation that a string x of length $\|x\|$ has $\|x\|$ substrings of length 1, $\|x\|-1$ substrings of two characters, $\|x\|-2$ substrings of three characters, and so on. In addition, the empty string λ is a substring of any string, including λ itself. Therefore the sum is the well-known formula of the sum of the first n consecutive positive integers (i.e., $n(n+1)/2, n \in \mathbb{Z}^+$) (Grimaldi 2004) to which we add one in order to account for the empty string as well. As a direct consequence, for a multiset language L consisting of multiple strings $x \in L$ various lengths $\|x\|$, the cardinality of M, $|M|$ is given by:

$$|M| = \sum_{x \in L} \left[\left(\sum_{i=1}^{\|x\|} i \right) + 1 \right] = \sum_{x \in L} \left[\frac{\|x\|(\|x\|+1)}{2} + 1 \right] = \left[\sum_{x \in L} \frac{\|x\|(\|x\|+1)}{2} \right] + |L|$$

The definition of a substring multiset of a language is followed by three fundamental definitions of all possible relations between strings in a DDAM models, together with their reflexive and transitive reflexive reductions.

Definition 5. A string $x \in \Sigma^*$ is a *left substring* (prefix) of $w \in \Sigma^*$ if there exist $y \in \Sigma^*$ such that $w = xy$. If $x \neq w$ or $y \neq \lambda$ then x is a *proper left substring* (proper prefix) of w (reflexive reduction). If $y \in \Sigma$, i.e., $\|y\| = 1$ then x is the *longest proper prefix* (LPP) of w (transitive reflexive reduction). For example, *a* is a prefix of all *a*, *ab* and *abc*, a proper prefix of only *ab* and *abc* and a LPP of only *ab*.

Definition 6. A string $y \in \Sigma^*$ is a *right substring* (suffix) of $w \in \Sigma^*$ if there exist $x \in \Sigma^*$ such that $w = xy$. If $y \neq w$ or $x \neq \lambda$ then y is a *proper right substring* (proper suffix) of w (reflexive reduction). If $x \in \Sigma$, i.e., $\|x\| = 1$ then y is the *longest proper suffix* (LPS) of w

(transitive reflexive reduction). For example, c is a suffix of all c, bc and abc, a proper suffix of only bc and abc and a LPS of bc only.

Definition 7. A string $x \in \Sigma^*$ is a *proper middle substring* (or proper diafix) of $w \in \Sigma^*$ if x is both a proper left substring and a proper right substring of w. If x is also LPP and LPS of w then x is also the *longest proper diafix* (LPD) of w (transitive reflexive reduction). For example, b is a proper diafix of all abc, and $abcd$, but a LPD of abc only.

The binary relations "*is substring of*", "*is left substring of*", "*is right substring of*" are reflexive, anti-symmetric and transitive and form partial orders on Σ^*.

The binary relations LPP (denoted as $\hat{\preceq}$), LPS (denoted as $\hat{\preceq}$) and LPD (denoted as $\hat{\preceq}$) are all *string cover relations* $\hat{\prec}$ which, by analogy with the general cover relation defined earlier, is the transitive reflexive reduction of the generic "*is substring of*" relation. If L is a language over some alphabet, a string $z \in L$ is said to cover a string $x \in L$ if there is no string $y \in L$, such that $y \neq z$ and x is a substring of y and y is a substring of z. In other words, a string cannot cover itself (the relation is not reflexive) and if $x \hat{\prec} y$ and $y \hat{\prec} z$ then x is not covered by z (the cover relation is not transitive), for all $x, y, z \in L, y \neq z$.

3.1.4 Partial order sets (posets)

All formal definitions of this subsection are available in Appendix 1 and are gleaned from literature on partial order sets. They begin with the formal definition of a special type of *poset* where the binary relation is the *is substring of* relation. Further, highly relevant concepts such as that of a *maximal element*, *greatest element*, *minimal element* and *least* element are also defined, followed by a series of additional definitions marked by black diamonds and which are important in the context of poset literature but which are not directly being used in this dissertation.

The sequence of definitions is continued with additional highly relevant concepts such as that of a *totally ordered set*, as well as those of *chain, saturated chain* and *maximal chain*, ending with the definition of the *length of a partial ordered set*.

The sequence of definitions is further continued with additional highly relevant concepts such as *totally unordered set* as well as those of *anti-chain* and *width of a partial ordered set*.

The section concludes with the important definition of *ranked* poset, *rank function* and *graded* poset followed by a series of additional secondary definitions marked by black diamonds.

3.1.5 Combinatorics

The Combinatorics fundamentals subsection begins with the definition of the *n*th *central binomial coefficient* followed by that of *Catalan numbers*, *integer partition* and *combinatorial compositions*. Further, the mathematical objects named *Dyck paths* and *Dyck words* are defined and, in this context, Dyck path features such as *peaks*, *valleys*, *ascents* and *descents* are also introduced. These definitions are important in order to characterize the DDAM model from a combinatorial point of view and ultimately to have a grasp on its combinatorial complexity. Because the total number of Dyck paths of semilength n and is given by the Catalan number C_n one now possesses a clear formula for counting how many different representations are possible in a DDAM model, for a sequence of a given length.

Finally, as an original contribution to this section, it is proposed that Dyck paths and Dyck words could be regarded as *generalizations of combinatorial compositions*. This is based on the observation that a Dyck word of semilength n represents the integer n as a sum of positive as well as negative integers given by replacing the peaks and the valleys in a Dyck word by positive height and negative depth values, respectively. For example, given the five Dyck words of semilength 3, the five generalized combinatorial compositions of 3 are: $(1-0+1-0+1)$, $(2-0+1)$, $(1-0+2)$, $(2-1+2)$, as well as (3) itself.

3.2 THE UNCONSTRAINED SUBSTRING POSET

In this section the building block of the DDAM model is formally defined and followed by additional important definitions (e.g., string compositions, composition ambiguity, etc.). These definitions form the core of the theoretical model.

Definition 8. Let M be the substring multiset of a multiset language L over some alphabet Σ. Let $\hat{M}$ be the compression set of M and $\prec$ be a binary relation between pairs of elements (x_1,n_1), (x_2,n_2) in $\hat{M}$, such that x_1 is a substring of x_2. The poset $P=(\hat{M},\prec)$ is an *unconstrained substring poset* which:

- is a ranked poset with the rank function $\phi((x,n))=\|x\|$, $(x,n)\in\hat{M}$,
- has a least element (λ,n),
- does not necessarily have a greatest element,
- has a height given by the length of the longest string in the multiset language L.

Perhaps most importantly, though the unconstrained substring poset is easy to construct theoretically, it turns out that it has a quadratic space complexity which translates into unfeasibility of real world implementations. This is so because, for a string of length n the upper bound on the number of nodes in the unconstrained substring poset is a quadratic function of n. As an example, in the case of a language with 100 distinct strings each of length 4096 characters and each composed of distinct 4096 characters from an alphabet of 100×4096 characters, the unconstrained substring poset will have $100\times4096(4096+1)/2+100$, or 839,065,700 nodes. If one adds a linear overhead of 100 bytes/node in order to implement the structure in a real computer, the amount of memory needed would be around 80,020 megabytes which clearly is an unreasonable amount given the relative small size of the representation problem. Because of this, the unconstrained substring poset has been evolved into a more feasible approach, the

constrained substring poset, which will be addressed in the subsequent sections of this chapter.

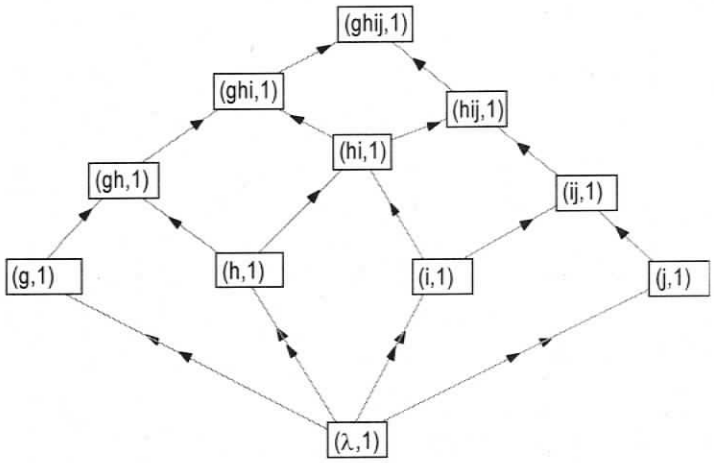


Figure 41. Unconstrained substring poset of the multiset language $L=\{ghij\}$

Example. If $L=\{ghij\}$ is a multiset language over the alphabet $\Sigma=\{g,h,i,j\}$ then $M=\{ghij,ghi,hij,gh,hi,ij,g,h,i,j,\lambda\}$ is the substring multiset of L and $\hat{M}=\{(ghij,1),(ghi,1),(hij,1),(gh,1),(hi,1),(ij,1),(g,1),(h,1),(i,1),(j,1),(\lambda,1)\}$ is the compression multiset of the substring multiset M. Therefore $P=(\hat{M},\prec)$ is an unconstrained substring poset and the cardinality of M, $|M|$ is 11, i.e., the number of substrings of the only string in L including the one instance of the empty string.

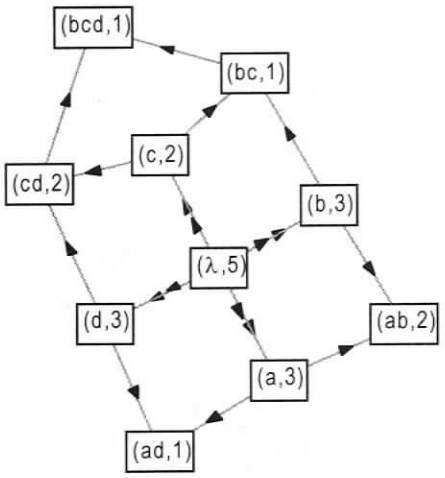


Figure 42. Unconstrained substring poset of the multiset language $L=\{ab,ab,bcd,cd,ad\}$

Example. If $L=\{ab,ab,bcd,cd,ad\}$ is a multiset language over the alphabet $\Sigma=\{a,b,c,d\}$ then

$M=\{ab,a,b,\lambda,ab,a,b,\lambda,bcd,bc,cd,b,c,d,\lambda,cd,c,d,\lambda,ad,a,d,\lambda\}$ and

$\hat{M}=\{(ab,2),(a,3),(b,3),(bcd,1),(bc,1),(cd,2),(c,2),(d,3),(ad,1),(\lambda,5)\}$.

$P=(\hat{M},\prec)$ is an unconstrained substring poset and the cardinality of M, $|M|$ is 4+4+7+4+4=2+3+3+1+1+2+2+3+1+5=23, i.e., the number of substrings of all strings in L including the five instances of the empty string.

Definition 9. A *prefix chain* in an unconstrained substring poset is a saturated chain in which the total ordering relation is the LPP relation $\hat{\preceq}$. Similarly, a *suffix chain* is a saturated chain in which the total ordering relation is the LPS relation $\hat{\preceq}$ and a *diafix chain* is a saturated chain in which the total ordering relation is the LPD relation $\hat{\preceq}$. For example, in Figure 41, $(\lambda,1)\hat{\preceq}(g,1)\hat{\preceq}(gh,1)\hat{\preceq}(ghi,1)\hat{\preceq}(ghij,1)$ is a prefix chain, $(\lambda,1)\hat{\preceq}(i,1)\hat{\preceq}(hi,1)\hat{\preceq}(ghi,1)$ is a suffix chain and $(\lambda,1)\hat{\preceq}(ij,1)$ is a diafix chain.

Definition 10. A *LPP function* denoted $\nearrow$ in an unconstrained substring poset $P=(\hat{M},\prec)$ is a function $\nearrow:\hat{M}\to\hat{M}$ such that $\nearrow((x,n))\hat{\preceq}(x,n)$ for all $(x,n)\in\hat{M}$. Similarly, a *LPS function* denoted $\nwarrow$ is a function $\nwarrow:\hat{M}\to\hat{M}$ such that $\nwarrow((x,n))\hat{\preceq}(x,n)$ for all $(x,n)\in\hat{M}$.

For example, in Figure 41, $\nearrow((g,1))=(\lambda,1)$, $\nearrow((gh,1))=(g,1)$, $\nearrow((ghi,1))=(gh,1)$ and $\nwarrow((i,1))=(\lambda,1)$, $\nwarrow((hi,1))=(i,1)$, $\nwarrow((ghi,1))=(hi,1)$. A LPD function is the composition of a LPP function and a LPS function, and can be written either $\nearrow(\nwarrow((x,n)))$ or $\nwarrow(\nearrow((x,n)))$, both forms being equivalent.

3.2.1 String compositions

Definition 11. A *string composition* of a string $x\neq\lambda$ in the unconstrained substring poset $P=(X,\prec)$ of a multiset language is a $2\|x\|+1$-tuple (or sequence, list) $C=\left((x_1,n_1),(x_2,n_2)\ldots\left(x_{(2\|x\|+1)},n_{(2\|x\|+1)}\right)\right)$, of elements from X such that $x_i\hat{\preceq}x_{i+1}$ or $x_{i+1}\hat{\preceq}x_i$, i.e., such that the strings in every pair of consecutive elements in C are related through the LPP (is longest proper prefix) or LPS (is longest proper suffix) relation. The composition also forms a Dyck path of semilength $\|x\|$ in P and can be encoded by Dyck words. The path starts from and ends at the least element of the poset P and consists of rise and fall steps corresponding to the LPP and LPS relations, respectively. It follows that the number of string compositions of a string x is equal to total number of Dyck paths of semilength $\|x\|$ which is given by the Catalan number $C_{\|x\|}$.

Definition 12. The *rank* $\Xi(C)$ of a string composition C is the sum of the ranks of all elements in C. Because the rank of an element in the unconstrained substring poset is the length of the string in that element then:

$$\Xi(C) = \sum_{(x,n)\in C} \phi(x) = \sum_{(x,n)\in C} \| x \|$$

Definition 13. Let $P = (X, \prec)$ be the unconstrained substring poset of a language L over an alphabet Σ, let (x,n) be an element in X, $(x,n) \in X$, and $S \subseteq X$ a subset of X such that x is a LPP of y, i.e., $x \hat{\preceq} y$, for all $(y,m) \in S$. The *LPP ambiguity* or *prefix ambiguity* of (x,n) is a function $\underset{\sim}{\psi} : X \to [0, |\Sigma|]$ such that $\underset{\sim}{\psi}((x,n)) = |S|$. The prefix ambiguity can be also expressed in bits by applying the base two logarithm to the result of the function $\underset{\sim}{\psi}$, providing that $\log_2(0)$ is defined and equal to 0.

Therefore, the definition of the prefix ambiguity of a substring x is context dependent and depends on the strings y to which the string x is a LPP. Given that $\|y\| - \|x\| = 1$, i.e., the strings x and y can only differ by at most one character in Σ, the maximum value of the prefix ambiguity is therefore $|\Sigma|$, or $\log_2(|\Sigma|)$ bits. Therefore the prefix ambiguity of a string in a language L is bounded only by the size of the alphabet over which the language L is constructed. It also follows that $\underset{\sim}{\psi}((x,n))$ is zero (or 0 bit) when (x,n) is a maximal element in P and that $\underset{\sim}{\psi}((x,n))$ is 1, or 0 bit when x is a LPP of at most one string y, in which case x and y are said to be strongly, or *non-ambiguously associated*. If $1 < \underset{\sim}{\psi}((x,n)) \leq |\Sigma|$, i.e., x is the longest proper prefix of more than one string, then x and y are said to be *ambiguously associated* and the association ambiguity is equal to $\log_2(\underset{\sim}{\psi}((x,n)))$.

The *LPS ambiguity* or *suffix ambiguity* $\underset{\sim}{\psi}$ and the *LPD ambiguity* or *diafix ambiguity* $\underset{\sim}{\psi}$ are defined analogously around the LPS and LPD relations respectively, and have similar properties as the prefix ambiguity.

For example, in the substring poset in Figure 42, $\underline{\psi}\left((a,3)\right)=2=1$ bit, $\underline{\psi}\left((a,3)\right)=0=0$ bit, $\underline{\psi}\left((a,3)\right)=0=0$ bit, $\underline{\psi}\left((b,3)\right)=1=0$ bit, $\underline{\psi}\left((b,3)\right)=2=1$ bit and $\underline{\psi}\left((b,3)\right)=0=0$ bit.

Definition 14. The *prefix ambiguity* $\underline{\Psi}(C)$, *suffix ambiguity* $\underline{\Psi}(C)$ and *diafix ambiguity* $\underline{\Psi}(C)$ of a string composition C are given by the sum of the LPP, LPS and LPD ambiguities expressed in bits, of each element in C, respectively, assuming that $\log_2(0)=0$:

$$\underline{\Psi}(C)=\sum_{(x,n)\in C}\log_2\left(\underline{\psi}\left((x,n)\right)\right)$$

$$\underline{\Psi}(C)=\sum_{(x,n)\in C}\log_2\left(\underline{\psi}\left((x,n)\right)\right)$$

$$\underline{\Psi}(C)=\sum_{(x,n)\in C}\log_2\left(\underline{\psi}\left((x,n)\right)\right)$$

For example, in Figure 42, for the string *ab*, the prefix and suffix ambiguities of the composition $C=\left((\lambda,5),(a,3),(\lambda,5),(b,3),(\lambda,5)\right)$, are

$$\underline{\Psi}(C)=3\log_2\left(\underline{\psi}\left((\lambda,5)\right)\right)+\log_2\left(\underline{\psi}\left((a,3)\right)\right)+\log_2\left(\underline{\psi}\left((b,3)\right)\right),$$

$$\underline{\Psi}(C)=3\log_2\left(\underline{\psi}\left((\lambda,5)\right)\right)+\log_2\left(\underline{\psi}\left((a,3)\right)\right)+\log_2\left(\underline{\psi}\left((b,3)\right)\right) \text{ and}$$

$$\underline{\Psi}(C)=3\log_2\left(\underline{\psi}\left((\lambda,5)\right)\right)+\log_2\left(\underline{\psi}\left((a,3)\right)\right)+\log_2\left(\underline{\psi}\left((b,3)\right)\right).$$

Therefore $\underline{\Psi}(C)=3\times2+1+0=7$ bits, $\underline{\Psi}(C)=3\times2+0+0=6$ bits and $\underline{\Psi}(C)=3\times2+0+0=6$ bits. However, for the same string *ab* the prefix and suffix ambiguities of the composition $C=\left((\lambda,5),(a,3),(ab,2),(b,3),(\lambda,5)\right)$ are slightly lower

because $\underset{\sim}{\psi}((ab,2))=0$ and $\underset{\sim}{\psi}((ab,2))=0$ and therefore, $\underset{\sim}{\Psi}(C)=2\times 2+1+0=5$ bits and $\underset{\sim}{\Psi}(C)=2\times 2+0+0=4$ bits.

Definition 15. A *min rank (or trivial) string composition* of a string $x\neq\lambda$ is the string composition consisting of alternating rise and fall steps corresponding to the Dyck word $\underbrace{1010\ldots10}_{2\|x\|}$ of semilength $\|x\|$. Alternatively, the trivial string composition of the string x is the string composition that contains $\|x\|$ peaks and $\|x\|-1$ valleys. It follows that the maximum length of any ascent or descent in trivial compositions is 1 and that the rank of the trivial composition of any string x is $\|x\|$. For example, compositions #1 in Table 17 and Table 18 are trivial compositions.

Definition 16. A *max rank composition* of a string $x\neq\lambda$ is the string composition consisting of one ascent and one descent and which corresponds to the Dyck word $\underbrace{111\ldots1}_{\|x\|}\underbrace{0\ldots000}_{\|x\|}$ of semilength $\|x\|$, which contains $\|x\|$ consecutive 1s and $\|x\|$ consecutive 0s. Alternatively, the max rank string composition of the string x is the string composition that contains one peak and no valleys. It follows that the maximum length of the ascent and of the descent in a complete composition is $\|x\|$. In addition, the value of the rank of the max rank composition of the string x is $\|x\|^2$. The proof is trivial if one observes that rank of such a composition is calculated as the sum:

$$0+1+2+\ldots+(\|x\|-1)+\|x\|+(\|x\|-1)+\ldots+2+1+0$$

which is $\|x\|$ plus twice the sum of the first $\|x\|-1$ natural numbers. Therefore,

$$\|x\|+\frac{2(\|x\|-1)(\|x\|-1+1)}{2}=\|x\|+\|x\|(\|x\|-1)=\|x\|+\|x\|^2-\|x\|=\|x\|^2$$

For example, composition #10 in Table 18 is a max rank composition.

Definition 17. A *non-overlapped string composition* of a string $x \neq \lambda$ is a string composition of the string x whose Dyck word of semilength $\|x\|$ corresponds to a combinatorial composition of $\|x\|$, i.e., in which the valleys have a zero depth. For example, in Table 17, most of the string compositions are non-overlapped. For a string x, there are $2^{\|x\|-1}$ non-overlapped compositions. The rest of the compositions, up to the Catalan number $C_{\|x\|}$ are therefore overlapped compositions, in which the depth of at least one valley in the corresponding Dyck path is higher than zero. As a direct consequence of this definition, trivial and maximal rank compositions are always non-overlapped compositions.

An important property of the non-overlapped string compositions is that they can be rendered in a form that is easily legible, by displaying only the strings of the elements that correspond to the peaks in the Dyck word encoding of that non-overlapping composition, without fear of information loss in the output (e.g., Table 17, Table 18). This property makes non-overlapped string compositions of extreme importance in the context of automatic lexical acquisition and text segmentation experiments that form the object of the next chapter.

Example. Let $L = \{abc\}$ be a multiset language. The unconstrained substring poset of L is $P = \left(\{(abc,1),(ab,1),(bc,1),(a,1),(b,1),(c,1),(\lambda,1)\},\prec\right)$ and the number of string compositions is the Catalan number C_3 which is equal to 5. Table 17 lists all possible compositions of *abc* as well as their ambiguity, corresponding Dyck path and word and generalized combinatorial composition.

#	Generalized combinatorial composition of 3	Non overlapping	Ambig. (bits)	Rank	Composition	Dyck path and word
1	1-0+1-0+1	|a|b|c|	12.68	3	| a | b | c |	101010
2	2-0+1	|ab|c|	9.51	5	| a ab b | c |	110010
3	1-0+2	|a|bc|	9.51	5	| a | b bc c |	101100
4	3	|abc|	6.34	9	| a ab abc bc c |	111000
5	2-1+2	-	6.34	7	| a ab b bc c |	110100

Table 17. The compositions of the string *abc* (the empty string is denoted by the pipe character "|"); the greyed composition (#5) has the lowest ambiguity value (6.34 bits) and for that ambiguity value, a minimal rank (7)

Example. If $L=\{abcd\}$ then the number of string compositions in the unconstrained substring poset of L is the Catalan number C_4 which is equal to 14.

$$C_4=\frac{1}{5}\binom{8!}{4}=\frac{8!}{5!\times 4!}=\frac{8\times 7\times 6\times 5\times 4\times 3\times 2\times 1}{5\times 4\times 3\times 2\times 1\times 4\times 3\times 2\times 1}=\frac{56}{4}=14$$

Table 18 lists all possible compositions of *abcd* as well as their ambiguity, corresponding Dyck word and generalized combinatorial composition.

#	Generalized combinatorial composition of 4	Non-Overlapping	Ambig. (bits)	Rank	Composition	Dyck word
1	1-0+1-0+1-0+1	|a|b|c|d|	20	4	| a | b | c | d |	10101010
2	2-0+1-0+1	|ab|c|d|	16	6	| a ab b | c | d |	11001010
3	1-0+1-0+2	|a|b|cd|	16	6	| a | b | c cd d |	10101100
4	1-0+2-0+1	|a|bc|d|	16	6	| a | b bc c | d |	10110010
5	3-0+1	|abc|d|	12	10	| a ab abc bc c | d |	11100010
6	1-0+3	|a|bcd|	12	10	| a | b bc bcd cd d |	10111000
7	2-0+2	|ab|cd|	12	8	| a ab b | c cd d |	11001100
8	2-1+2-0+1	-	12	8	| a ab b bc c | d |	11010010
9	1-0+2-1+2	-	12	8	| a | b bc c cd d |	10110100
10	4	|abcd|	8	16	| a ab abc abcd bcd cd d |	11110000
11	3-2+3	-	8	14	| a ab abc bc bcd cd d |	11101000
12	3-1+2	-	8	12	| a ab abc bc c cd d |	11100100
13	2-1+3	-	8	12	| a ab b bc bcd cd d |	11011000
14	2-1+2-1+2	-	8	10	| a ab b bc c cd d |	11010100

Table 18. The compositions of the string *abcd* ; the greyed composition (#14) has the lowest ambiguity value (8 bits) and for that ambiguity value a minimal rank (10)

3.3 CONSTRAINED SUBSTRING POSETS

In this section two variants of constrained substring posets are formally defined and their functionality demonstrated. This provides the transition from purely theoretical models such as the constrained substring poset towards more practical ones.

Constraining poset structures to the elements that exhibit certain properties is desirable because creating and storing unconstrained substring posets of languages with potentially long strings is unfeasible due to the quadratic space complexity in the length of those strings. Empirical results at the end of this chapter suggest that, by doing this, the prohibitive memory requirements may be reduced significantly for high dimensional natural sequences (e.g., text, DNA) which are common in Medical Informatics and which form extremely sparse pattern spaces.

3.3.1 The simple constrained substring poset

The following definition introduces the first version of the constrained substring poset, the simple constrained substring poset (Pantazi and Moehr 2005) which was used to approach the problem of *unsupervised lexical acquisition*.

Let $\delta, \beta \in \mathbb{N}$ be two parameters. A *constrained poset* relative to δ, β is an unconstrained substring poset $P = (X, \prec)$ for which the following conditions are true for all $(x_1, n_1) \prec (x_2, n_2)$, $(x_1, n_1), (x_2, n_2) \in X$:

1. $\|x_2\| \log_2 (n_2 + \beta) \geq \|x_1\| \log_2 (n_1)$
2. $n_1, n_2 > \delta$

In real prototype implementations of this model, the quadratic space complexity prevents the trivial possibility of creating the unconstrained substring poset first and then deriving the constrained poset from it. In addition, in many real problem instances such as those encountered in biomedicine, the language from which the substring poset is to be derived

is not specified (i.e., unknown) upfront and is revealed sequentially, usually one string at a time (e.g., when reading a text file). The alternative suggests itself and consists of creating the constrained substring poset gradually, in an online manner, from the partial information provided from each individual string in a language, as each of these strings become an input in the model. A direct consequence of this approach and also an *unsolved problem as of writing this dissertation* is that the substring counts in a poset are not 100% accurate. The substring counts are actually only lower bound estimations of the real counts.

The problem of determining the base set X of a simple constrained substring poset P, given a certain language, is similar to the problems of *unsupervised lexical acquisition* and *grammar induction*. The constraints provide ways of coping with space complexity issues through the *constraint parameters* δ and β. An intuitive way to think of the two parameters is that they encode a combination of inhibiting and facilitating forces that drive the online, hierarchical growth of the poset. The configurations of the two parameters will be referred to as *normal constraints* $(\delta=1,\beta=0)$, *tight constraints* $(\delta>1)$, *relaxed constraints* $(\delta=1,\beta>0)$ and *no constraints* $(\delta=0,\beta\geq 0)$. For example, for the input string *abcdefcdefababefcdefabcdcdefabcdefefefabcdef* of length 44, normal constraint conditions result in

$$X=\left\{\begin{matrix}(a,6),(b,6),(c,7),(d,7),(e,8),(f,8),\\(ab,6),(fa,4),(ef,8),(bc,4),(cd,7),(de,5),(cde,4),(\lambda,1)\end{matrix}\right\}$$

No constraints would cause the cardinality of X to become $|X|=44\times(44+1)/2+1=991$ elements compared to just 14 in the normally constrained case. The unconstrained X would include many long substrings which appear only once in the input (e.g., *befc*). The constraints are therefore important for dealing with combinatorial issues and to drive the growth process of the poset.

Hebbian learning is a simple, effective and biologically plausible strategy for associative learning: "if two nodes on either side of a synapse (connection) are activated

simultaneously (i.e., synchronously) then the strength of that synapse is selectively increased" (Haykin 1994). The counts of related substrings (e.g., *a* and *ab*) in the substring poset could be thought of a model of such a Hebbian synapse. These counts indeed form a time dependent (i.e., dynamic, changing with time), local and correlational mechanism: the higher and closer the counts of related strings (e.g., *(ab,6)* with *(a,6)* and*(b,6)*), the stronger their "association."

The model of simple constrained substring poset has been used in a series of experiments of unsupervised lexical acquisition (Pantazi and Moehr 2005) that are presented in the following chapter and has since evolved into a more advanced model, the $\alpha\lambda\omega$ deterministic constrained substring poset. Before delving into the new model, a series of additional important preliminary definitions and an important theorem (i.e., the flip theorem) must be given.

3.3.2 Constrained string compositions

Definition 18. An *optimal string composition* C of a string $x \neq \lambda, x \in L$ in the context of the multiset language L is a string composition whose prefix and suffix ambiguities and rank $\Xi(C)$ are minimal. An alternative way to define an optimal string composition is by using a unique cost function defined as the product of the square root of its rank $\Xi(C)$ and the sum of its prefix $\underset{\sim}{\Psi}(C)$ and suffix $\underset{\sim}{\Psi}(C)$ ambiguities (in bits) and whose value has to be minimal for an optimal string composition:

$$\min\left[\left(\underset{\sim}{\Psi}(C)+\underset{\sim}{\Psi}(C)\right)\sqrt{\Xi(C)}\right]$$

As a consequence, the trivial composition of x can only be optimal in the case when $\|x\|=1$, in which case the trivial composition of x $C=\left((\lambda,n),(x,m),(\lambda,n)\right)$ is identical to the max rank composition of x.

Definition 19. Let C be a string composition of a string $x, \|x\| > 1$ in the unconstrained substring poset $P = (X, \prec)$ with X built on an alphabet Σ, and let $(x_{i-1}, n_{i-1}), (x_i, n_i), (x_{i+1}, n_{i+1}) \in C$, $i \in \mathbb{Z}^+, i < 2\|x\|$ be a 3-tuple of consecutive elements in C such that they form a valley in the corresponding Dyck path, i.e., $(x_i, n_i) \hat{\preceq} (x_{i-1}, n_{i-1})$ and $(x_i, n_i) \hat{\preceq} (x_{i+1}, n_{i+1})$. A *flip up operation* on the 3-tuple $((x_{i-1}, n_{i-1}), (x_i, n_i), (x_{i+1}, n_{i+1}))$ is the operation by which the valley becomes the peak $((x_{i-1}, n_{i-1}), (x'_i, n'_i), (x_{i+1}, n_{i+1}))$, i.e., by which (x_i, n_i) is replaced with $(x'_i, n'_i) \in X$ such that, $(x_{i-1}, n_{i-1}) \hat{\preceq} (x'_i, n'_i)$, $(x_i, n_i) \hat{\preceq} (x'_i, n'_i)$ and $(x_{i+1}, n_{i+1}) \hat{\preceq} (x'_i, n'_i)$. From a Dyck word encoding perspective, a flip up operation causes a 01 to become a 10 in that Dyck word (Figure 43). The direct consequence of a flip up operation is that the rank of C increases by 2 and that, potentially, the ambiguity of C decreases by a value up to $2\log_2(|\Sigma|)$.

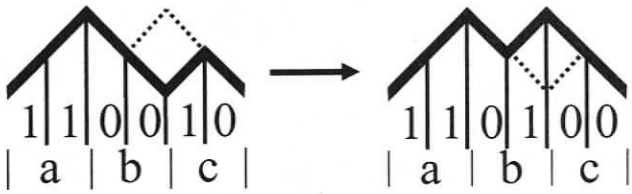


Figure 43. Illustration of the flip up operation on the 3-tuple ((b,1),(|,4),(c,1)) in the composition "| a ab b | c |" (rank 5); the 3-tuple valley becomes the peak ((b,1),(bc,1),(c,1)) and the composition becomes "| a ab b bc c |" (rank 7)

Definition 20. Let C be a string composition of a string $x, \|x\| > 1$ in the unconstrained substring poset $P = (X, \prec)$ with X built on an alphabet Σ, and let $(x_{i-1}, n_{i-1}), (x_i, n_i), (x_{i+1}, n_{i+1}) \in C$, $i \in \mathbb{Z}^+, i < 2\|x\|$ be a 3-tuple of consecutive elements in C such that form a peak in the corresponding Dyck path, i.e., $(x_{i-1}, n_{i-1}) \hat{\preceq} (x_i, n_i)$ and $(x_{i+1}, n_{i+1}) \hat{\preceq} (x_i, n_i)$. A *flip down operation* on the 3-tuple $((x_{i-1}, n_{i-1}), (x_i, n_i), (x_{i+1}, n_{i+1}))$ is the operation by which the peak becomes the valley $((x_{i-1}, n_{i-1}), (x'_i, n'_i), (x_{i+1}, n_{i+1}))$, i.e., by which (x_i, n_i) is replaced with $(x'_i, n'_i) \in X$ such that $(x'_i, n'_i) \hat{\preceq} (x_{i-1}, n_{i-1})$,

$(x_i', n_i') \hat{\preceq} (x_i, n_i)$ and $(x_i', n_i') \hat{\preceq} (x_{i+1}, n_{i+1})$. From a Dyck word encoding perspective, a flip down operation causes a 10 to become a 01 in that Dyck word. The direct consequence of a flip down operation is that the rank of C decreases by 2 and that, potentially, the ambiguity of C increases by a value up to $2\log_2(|\Sigma|)$.

Definition 21. A *flip down of a prefix chain* $(x_1, n_1) \hat{\preceq} (x_2, n_2) \hat{\preceq} \ldots \hat{\preceq} (x_k, n_k)$, $\|x_1\| > 1$ represents the replacement of each element (x_i, n_i) in that chain with its LPD $\nwarrow(\nearrow((x_i, n_i)))$ which results in

$$\nwarrow(\nearrow((x_1, n_1))) \hat{\preceq} \nwarrow(\nearrow((x_2, n_2))) \hat{\preceq} \ldots \hat{\preceq} \nwarrow(\nearrow((x_k, n_k))).$$

Similarly a *flip down of a suffix chain* $(x_1, n_1) \hat{\preceq} (x_2, n_2) \hat{\preceq} \ldots \hat{\preceq} (x_k, n_k)$, $\|x_1\| > 1$ represents the replacement of each element (x_i, n_i) in that chain with its LPD $\nwarrow(\nearrow((x_i, n_i)))$ which results in

$$\nwarrow(\nearrow((x_1, n_1))) \hat{\preceq} \nwarrow(\nearrow((x_2, n_2))) \hat{\preceq} \ldots \hat{\preceq} \nwarrow(\nearrow((x_k, n_k))).$$

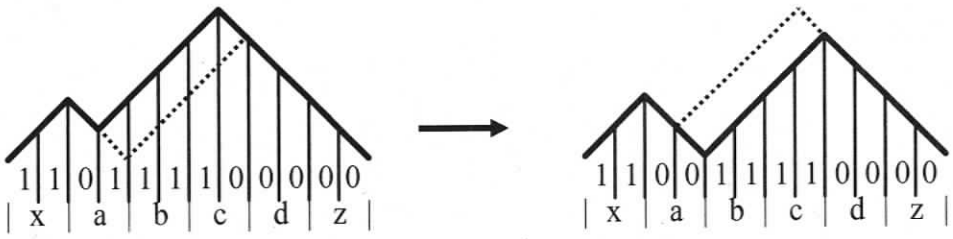


Figure 44. Illustration of the flip down of the prefix chain $(ab,1) \hat{\preceq} (abc,1) \hat{\preceq} (abcd,1) \hat{\preceq} (abcdz,1)$ in the composition "| x xa a ab abc abcd abcdz bcdz cdz dz z |" (rank 28); elements of the chain are replaced by their LPDs, and the chain becomes $(\lambda,7) \hat{\preceq} (b,1) \hat{\preceq} (bc,1) \hat{\preceq} (bcd,1)$ and the composition becomes "| x xa a | b bc bcd bcdz cdz dz z |" (rank 20)

For example, in Figure 44, the flip down of a four-element prefix chain in a composition results in decreasing the rank of that composition by 8, from 28 to 20. The definition of

the chain flip down is of high importance in the context of the decomposition algorithm presented later in this chapter.

3.3.3 The "flip theorem"

Let C and C' be two perfectly aligned compositions of the same string but with different ranks such that either $\|x_i\| \le \|x_i'\|$ or that $\|x_i'\| \le \|x_i\|$ but not both at the same time, for all $(x_i, n_i) \in C$ and $(x_i', n_i') \in C'$, $0 \le i < |C| - 1$. Essentially this premise states that we can only perform flip operations in one direction (up or down) in order to arrive at C' from C or vice versa. The flip theorem states that the number f of flip operations necessary to transform the composition C into the composition C' is equal to the absolute semi-difference of the ranks of the two compositions:

$$f = \frac{abs\left(\Xi(C') - \Xi(C)\right)}{2}$$

The proof is based on the previous definition from which it is easy to see that flip operations always change the rank of the flipped element (and of the whole composition for that matter) by 2. Hence, the number of flips of the ith element in C is the absolute semi-difference of the ranks of the two elements $f_i = abs\left(\|x_i'\| - \|x_i\|\right)/2$. Therefore, the total number of flip operations is:

$$f = \sum_{\substack{(x,n)\in C \\ (x',n')\in C'}} \frac{abs\left(\|x_i'\| - \|x_i\|\right)}{2} = \frac{1}{2} \sum_{\substack{(x,n)\in C \\ (x',n')\in C'}} abs\left(\|x_i'\| - \|x_i\|\right) = \frac{1}{2} abs\left[\sum_{(x',n')\in C'} \|x_i'\| - \sum_{(x,n)\in C} \|x_i\|\right]$$

Finally, because the rank of a string composition C is defined as $\Xi(C) = \sum_{(x,n)\in C} \|x\|$, it follows that

$$f = \frac{abs\left(\Xi(C') - \Xi(C)\right)}{2}.$$

From the "flip" theorem it follows that the upper bound of the number f of flip operations necessary to transform a composition of a string x into another composition of the string x and vice-versa is equal to the absolute semi-difference of the rank values of the max rank and the min rank (i.e., trivial) compositions:

$$f = \frac{abs\left(\Xi\left(C_{\max}\right) - \Xi\left(C_{\min}\right)\right)}{2} = \frac{\|x\|^2 - \|x\|}{2} = \frac{\|x\|\left(\|x\| - 1\right)}{2}$$

The main consequence of this upper bound is that it gives an estimation of the time complexity of the adaptive composition algorithm that will be presented later and which, in the worst case, must perform a number of flip operations which is a quadratic function of the length of a given string. On the positive side however, the worst-case scenario is unlikely to occur in case of natural, long sequences such as text. Since this bound is a function of only the length of a string, it can be very little dependent of the number of strings in a language. This is in perfect agreement with the general knowledge that the complexity of content addressable memories and secondary key retrieval models and algorithms, in general, does not depend on the number of items stored in such models but mostly on the length of the items. The length of items could be kept relatively small through structures which are hierarchical and compositional in nature.

As a general approach to constraining the substring poset model and to build associative memory models for sparse concept space representation, we will generally aim at creating and dynamically maintaining hierarchical compositional representations of strings through flip-up and flip-down operations on the compositions of each string.

3.3.4 The $\alpha\lambda\omega$ deterministic constrained substring poset

The $\alpha\lambda\omega$ deterministic constrained substring introduced in the following forms the building block of the dynamic deterministic associative memory model that was the main object of research of this dissertation. As a deterministic model, the associative memory must be capable of a fixed mode of operation which proceeds in a fashion entirely predictable by theory and which does not exhibit any approximate or probabilistic

behaviour. In this context, we require the deterministic constrained substring posets to allow accurate, non-ambiguous, bijective representation of languages. Therefore, before we formally define the model and the adaptive composition algorithm responsible for its functionality, it is necessary to explain that a certain feature of the model that gives the name $\alpha\lambda\omega$ came about from the need to alleviate the lack of accuracy of the substring instance counts in real implementations, as shown in the previous section. To put it in other words, the unconstrained substring poset makes it difficult to exploit substring instance count information when a multiset language L contains *strings which are already substrings* of other strings in L. For example, if L were the multiset language $\{abc, abc, abd, ab, ab, a, a\}$, the only way to tell from its unconstrained substring poset $\left(\{(abc,2),(ab,5),(bc,2),(a,7),(b,5),(c,2),(abd,1),(bd,1),(d,1),(\lambda,7)\},\prec\right)$ how many of the five instances of *ab* were part of longer strings such as *abc* or *abd* and how many instances were standalone *ab*'s, is by taking the difference between the count of *ab*'s (i.e., 5) and the sum of the counts of the strings to which *ab* is a proper substring (i.e., *abc* and *abd*) which is 2+1=3. But this requires accurate count information currently not achievable by the current model. A way to cope with this but at the expense of an additional linear amount of memory is to augment all strings in L with special start and end characters that are *not* part of the alphabet on which L is based, and to build the unconstrained substring poset from the augmented strings. Therefore the $\alpha\lambda\omega$ *deterministic constrained substring poset* of a multiset language L on an alphabet Σ, $\alpha, \omega \notin \Sigma$, is going to be constructed not from L but from the language L' on the augmented alphabet $\Sigma' = \Sigma \cup \{\alpha, \omega\}$. Therefore, L' is obtained from L by concatenating each string in L with special *begin* and *end* characters α and ω which are not included in Σ, i.e., $L' = \{x \mid x = \alpha y \omega, y \in L, \alpha \notin \Sigma, \omega \notin \Sigma\}$. As a result, in the $\alpha\lambda\omega$ *deterministic constrained substring poset*, each trivial composition of a string from L will contain the required begin and end characters as well as an additional number of instances of the empty string which is also a substring of the begin and end characters themselves. In addition to the properties of unconstrained substring posets, the $\alpha\lambda\omega$ unconstrained substring poset gains another important one. Because now we can distinguish between all

instances of strings in a language L without resorting to count information (which lacks accuracy in the current model implementations), regardless of them already being substrings of longer strings, the subset of maximal elements in the unconstrained version of a $\alpha\lambda\omega$ deterministic substring poset is going to be isomorphic to the abstract set $\tilde{L}$ of the initial multiset language for any language L. What this means is that that the $\alpha\lambda\omega$ deterministic constrained substring poset now allows accurate, non-ambiguous, unique, bijective, compositional representations of any string in a given language.

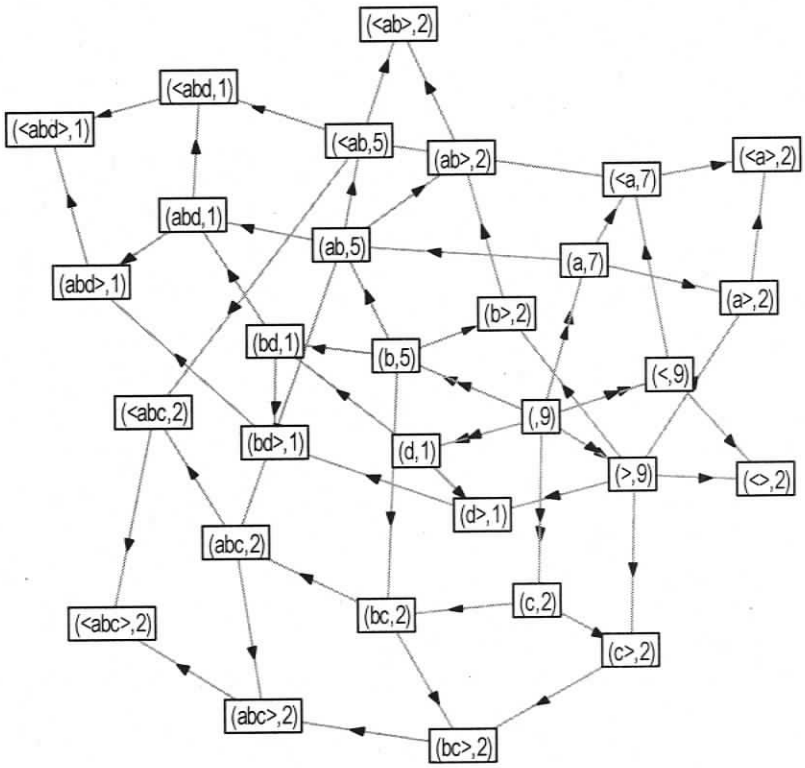


Figure 45. Example of unconstrained version of a $\alpha\lambda\omega$ deterministic substring poset; the alpha and omega symbols are denoted by "<" and ">" symbols respectively

For example, the unconstrained version of a $\alpha\lambda\omega$ deterministic substring poset of the language $L = \{abc, abc, abd, ab, ab, a, a, \lambda, \lambda\}$ is

$$P = \left(\left\{ \begin{array}{l} (\alpha abc\omega,2),(\alpha abc,2),(abc\omega,2)(\alpha ab,5),(abc,2),(bc\omega,2),(\alpha a,7),(ab,5), \\ (bc,2),(c\omega,2),(\alpha,9),(a,7),(b,5),(c,2),(\omega,9),(\lambda,9), \\ (\alpha abd\omega,1),(\alpha abd,1),(abd\omega,1),(abd,1),(bd\omega,1),(bd,1),(d\omega,1),(d,1), \\ (\alpha ab\omega,2),(ab\omega,2),(b\omega,2), \\ (\alpha a\omega,2), \\ (\alpha\omega,2) \end{array} \right\}, \prec \right)$$

The abstract set of L is $\tilde{L} = \{abc, abd, ab, a, \lambda\}$ and the subset of maximal elements in P is $\{\alpha abc\omega, \alpha abd\omega, \alpha ab\omega, \alpha a\omega, \alpha\omega\}$ which is clearly isomorphic to $\tilde{L}$, despite some strings in L already being substrings of other strings in L (e.g., ab, a, and the empty string). Therefore, the deterministic nature of the $\alpha\lambda\omega$ substring posets is preserved and every string in L has a unique representation despite the fact that the model lacks accuracy of substring counts in current prototype implementations.

3.3.5 The adaptive $\alpha\lambda\omega$ composition algorithm

The adaptive $\alpha\lambda\omega$ composition algorithm aims at creating an $\alpha\lambda\omega$ deterministic, constrained substring poset of a language L on an arbitrary alphabet Σ providing that $\alpha, \omega \notin \Sigma$. The input and internal parameters used by the algorithm are, in the order of their appearance in the algorithm:

- The $\alpha\lambda\omega$ deterministic, constrained substring poset P which is initially empty;
- The language L whose strings are to be represented in P;
- The LPP and LPS ambiguity thresholds $\underline{a}$ and $\underline{a}$ respectively;
- Two Boolean flag variables u and v to signal changes in string compositions;
- A string variable x_i which holds the augmented ith string from L;
- An array C_i which holds the elements of composition of string x_i, in P;

- Two dynamic arrays $\underline{C}$ and $\underline{C}$ which hold the elements of a prefix and suffix chain in P, respectively;
- Two integer index variables i and j.

$COMPOSE(P, L, \underline{a}, \underline{a})$

```
01    repeat
02          u := true;
03          for i := 0 to |L| - 1 do
04          begin
05                x_i := αL[i]ω ;
06                C_i[0] := λ ;
07                for j := 0 to ||x_i||+1 do
08                begin
09                      C_i[2j+1] := x_i[j] ;
10                      C_i[2j+2] := λ ;
11                end;
12                repeat
13                      v := true;
14                      for j := 1 to 2||x_i|| do
15                      begin
16                            if (C_i[j] ≳ C_i[j-1]) and (C_i[j] ≳ C_i[j+1]) then
17                            begin
18                                  C := C_i[j] ≳ C_i[j-1] ≳ ...
19                                  C := C_i[j] ≳ C_i[j+1] ≳ ...
20                                  if (Ψ(C) > a) or (Ψ(C) > a) then
21                                  begin
22                                        FLIP_UP(C_i[j-1],C_i[j],C_i[j+1]) ;
23                                        u := false;
24                                        v := false;
25                                  end;
26                            end;
27                      end;
28                until v;
29          end;
30    until u;
```

Lines 01-30 are the main loop of the algorithm. As long as there are changes in the composition of any string as consequences of *FLIP_UP* operations, this loop keeps iterating through all strings in L. This approach is not very efficient for languages with many strings and could be improved by taking into account that the change in the composition of a string should only affect *a subset of* the strings in L, i.e., those that share

similar substrings. Though this approach is inefficient, its simplicity makes it appropriate for the purpose of explaining the algorithm functionality.

Lines 03-29 are the main **for** loop which iterates through every single string in L in order to represent it in P.

In **Line 05**, the internal variable x_i is assigned the value of the ith string in L, concatenated with the *begin* and *end* characters α and ω.

Lines 06-11 of the algorithm are responsible for creating the trivial $\alpha\lambda\omega$ composition C_i of the ith string x_i in L. The length of this composition is $2\|x_i\|+1$ and, per the definition of compositions, it starts with the empty string and ends with the empty string.

Lines 12-28 form the inner loop that takes care of the composition of a certain string x_i in L. As with the main outer **repeat** loop, the approach may not be very efficient for languages with long strings and could be improved by taking into account that the change in the composition of a string x_i affects only *some* parts of x_i i.e., those that are adjacent to the triple undergoing the last flip-up operation. Though this approach is inefficient its simplicity makes it appropriate for the purpose of explaining the algorithm functionality.

Lines 13-27 are the **for** loop which iterates through the triples of elements $\left(C_i[j-1], C_i[j], C_i[j+1]\right)$ in the composition C_i in order to modify the composition if necessary.

The code in **Line 16** is necessary to detect whether the current triple $\left(C_i[j-1], C_i[j], C_i[j+1]\right)$ is actually a valley, as only valleys can undergo flip up operations.

In **Lines 17-18** two internal arrays $\underrightarrow{C}$ and $\underleftarrow{C}$ are initialized to the elements of a prefix and suffix chains in P that form the two ascents at the bottom of which lies the valley formed by $\left(C_i[j-1], C_i[j], C_i[j+1]\right)$.

In **Line 20,** the decision whether a flip up operation is necessary or not is made on the basis that the prefix and suffix ambiguities $\Psi(\underset{\sim}{C})$ and $\Psi(\underset{\sim}{C})$ of the prefix and suffix chains $\underset{\sim}{C}$ and $\underset{\sim}{C}$ are higher than the prefix and suffix ambiguity thresholds $\underset{\sim}{a}$ and $\underset{\sim}{a}$.

Lines 22, 23 and 24 are the innermost instructions and are responsible with the flip-up operation on the triple $(C_i[j-1], C_i[j], C_i[j+1])$ as well as with resetting the u and v flags in order to indicate the changes that have been made to the structure of the poset P require additional iterations in the **repeat** loops.

3.3.5.1 Trivial compositions examples

Let $L = \{Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday\}$ be the set of the seven strings that name the weekdays. The first operation on each of these strings consists of augmenting them with the *begin* and *end* characters α and ω in order to obtain their trivial $\alpha\lambda\omega$ compositions. From this point on, in order to facilitate the incorporation of algorithm outputs in the text, table and figures, the Greek characters α, λ and ω will often be replaced by the open and closed round parentheses **"(" and ")"** respectively and by the pipe symbol "|" as in Table 19.

Original string length n	Trivial composition	Ambig	Rank n+2	Trivial composition length 2*n+5	Dyck word
6	|(|M|o|n|d|a|y|)|	86.7	8	2*6+5=17	1010101010101010
7	|(|T|u|e|s|d|a|y|)|	100.8	9	2*7+5=19	101010101010101010
9	|(|W|e|d|n|e|s|d|a|y|)|	121.7	11	2*9+5=23	10101010101010101010
8	|(|T|h|u|r|s|d|a|y|)|	109.3	10	2*8+5=21	1010101010101010101010
6	|(|F|r|i|d|a|y|)|	86.7	8	2*6+5=17	1010101010101010
8	|(|S|a|t|u|r|d|a|y|)|	110.3	10	2*8+5=21	1010101010101010101010
6	|(|S|u|n|d|a|y|)|	91.3	8	2*6+5=17	1010101010101010

Table 19. Trivial compositions of the seven strings that name the seven days of week

All trivial $\alpha\lambda\omega$ compositions in Table 19 have a length of $2n+5$, where n is the length of the original string. Therefore, the memory requirements to store the trivial compositions are linear. However, the upper bound of the memory requirements of non-trivial compositions is quadratic and is given by the number of elements in the unconstrained substring poset built from the sequences. But this is an extreme situation that occurs only in the case of a very compact representation space, in which case the

constraints cannot restrict the memory requirements of the model. For example, in such a compact space, all 2^n strings of length n composed of a's and b's should be in the language that is to be represented. The fact that natural sequences such as text have special properties such as an extreme sparseness of pattern space (e.g., not all possible strings are part of natural language) renders the approach feasible, at least for reasonable amounts of data such as sets of short sequences which are used frequently such as the weekday names, months names, proper names, dictionaries, short texts, etc.

3.3.5.2 Optimal compositions examples

Trivial compositions are often ambiguous and usually it is not possible for one element in them to allow the recall of the entire composition, unequivocally. For example, a *d* which is part of a composition in the context of a language made up of the weekdays names does not allow us to unequivocally recall any particular weekday. However, an *F*, in the very same context provides us with enough information to determine that the composition is that of the string *Friday*. In the case of the *d*, in order to be able to do the same thing, we would need additional information or context in order to disambiguate and determine which *d* was meant, the one in *day* or the one in *Wednes*.... Having additional context such as in the form of the next character after *d* could help us disambiguate the composition completely if the additional character was a *n* but leaves the pattern still ambiguous if the additional character was an *a*. One could say that, in the context of guessing the weekday names, knowing a *d* implies a prefix ambiguity of at least 1 bit, given by the choice between the two choices *dn* and *da* which are considered equiprobable. Therefore, assuming no errors in input, the pattern *dn* reduces the ambiguity/uncertainty to zero (i.e., *Wednesday*) while the pattern *da* does not reduce the uncertainty at all and requires additional context for its disambiguation since all seven sequences in our language contain it. The aim of the adaptive $\alpha\lambda\omega$ composition algorithm is to obtain optimal string compositions (Table 20) or, with prefix and suffix ambiguity levels below certain desired thresholds.

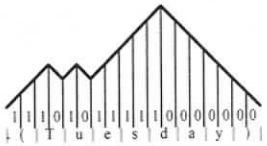
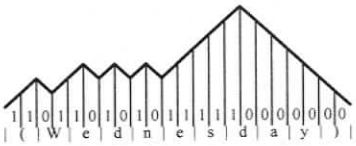
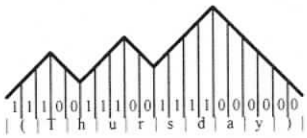
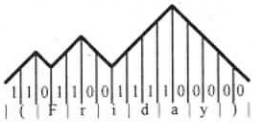
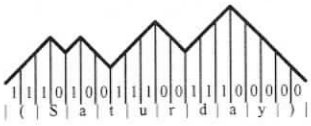
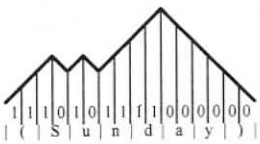
Length	Optimal composition	Dyck path	Amb	Rank
17	| ((M M Mo o on ond onda onday onday) nday) day) ay) y)) |		22.3	42
19	| ((T (Tu Tu Tue ue ues uesd uesda uesday uesday) esday) sday) day) ay) y)) |		24.3	59
23	| ((W W We Wed ed edn dn dne ne nes nesd nesda nesday nesday) esday) sday) day) ay) y)) |		23.3	67
21	| ((T (Th Th h hu hur hurs urs rs rsd rsda rsday rsday) sday) day) ay) y)) |		23.3	56
17	| ((F F Fr Fri ri i id ida iday iday) day) ay) y)) |		21.3	36
21	| ((S (Sa Sa Sat at t tu tur turd urd rd rda rday rday) day) ay) y)) |		22.3	50
17	| ((S (Su Su Sun un und unda unday unday) nday) day) ay) y)) |		23.3	46

Table 20. Optimal compositions of the seven strings that name the seven days of week

3.3.5.3 Visualization

For small languages such as the collection of the weekday names, the constrained substring poset structures can be successfully rendered as directed graphs. The relative small number of nodes and edges resulting from the adaptive composition algorithm on this particular data set, allow us to have a complete view of the underlying structure of the $\alpha\lambda\omega$ constrained substring poset.

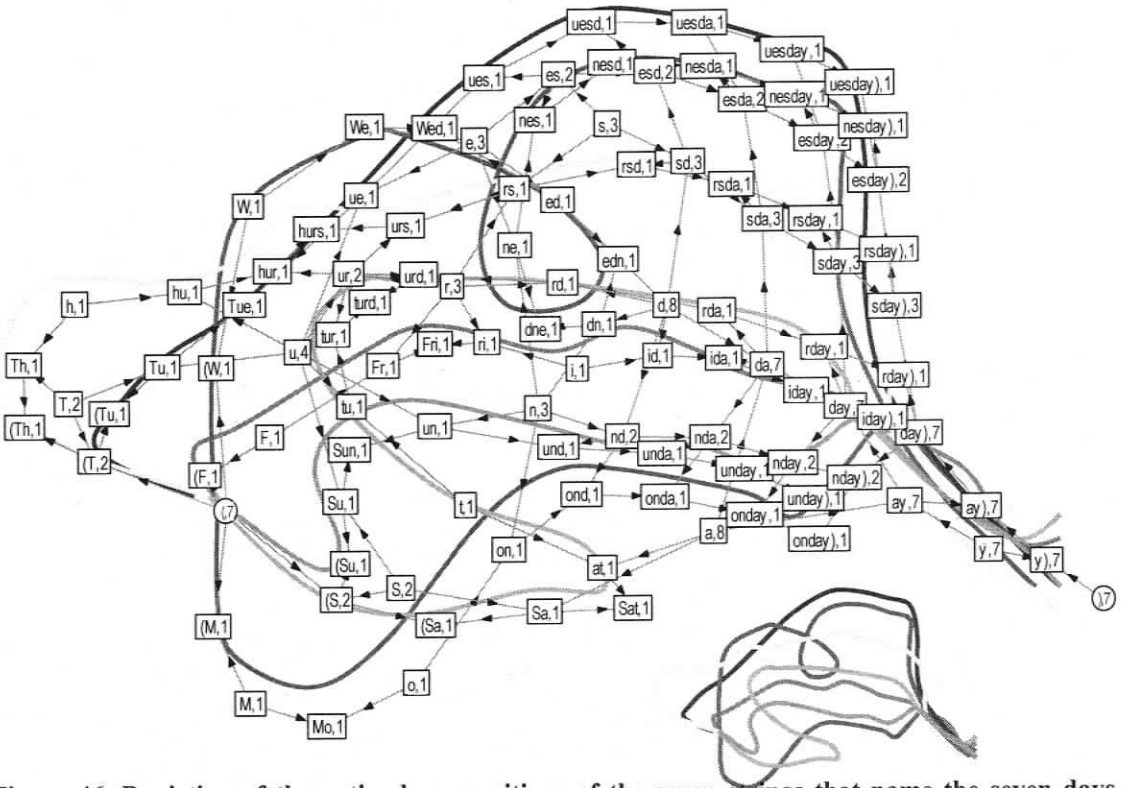


Figure 46. Depiction of the optimal compositions of the seven strings that name the seven days of week; the depiction shows all the substring elements that comprise the constrained substring poset

In addition, force directed automated graph layout approaches have been employed in order to obtain more aesthetic depictions such as the one in Figure 46. The self-organization of the graph structure results in a similarity preserving, two-dimensional display where similar elements in a graph are represented closely. By tracing the compositions of each string, one can also see how the paths that trace sequences sharing certain similarities such as *Tuesday* and *Wednesday* or *Sunday* and *Monday* are closely drawn and that all paths are very closely mapped toward the end that contains the pattern *day*. This particular combination of the substring posets and automatic layout algorithms comse fairly close to the paradigm of self organizing maps (Kohonen 2001) a fact also noticed by researchers on force directed automatic layout algorithms on general graphs (Frick, Ludwig et al. 1995).

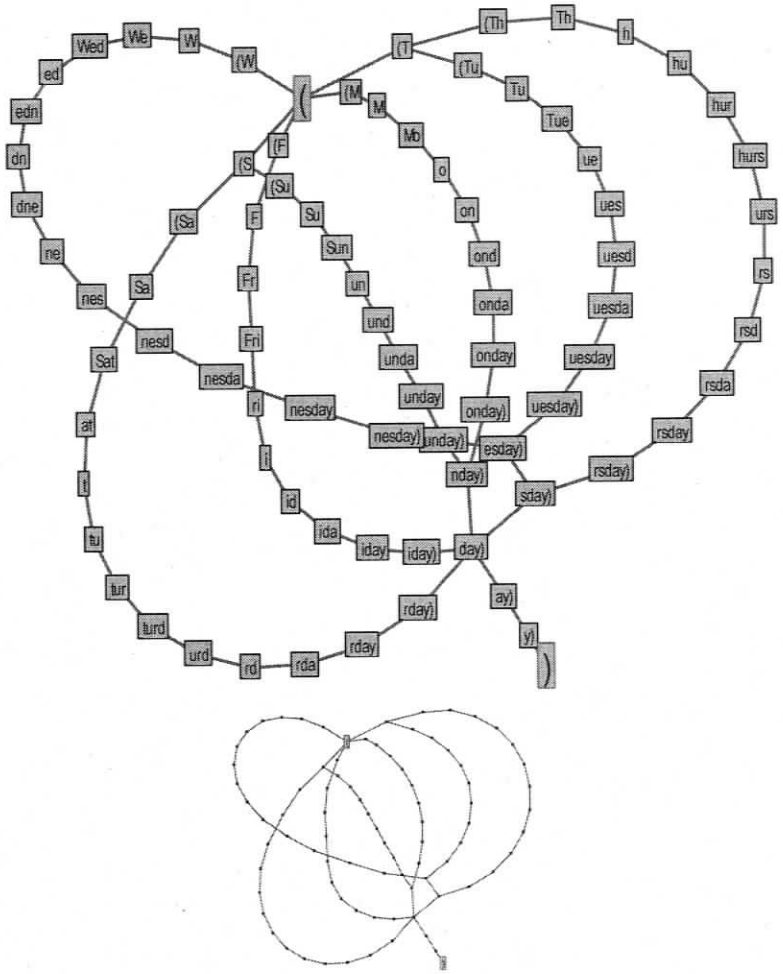


Figure 47. Depictions of the optimal compositions of the seven strings that name the seven days of week; the depiction shows only the elements that make the compositions

Retaining only the composition elements in the graphical display yields simpler structures such as those in Figure 47 which, as argued in the previous chapter, are a generalization of prefix and suffix trie structures that can be derived from the strings in the same language.

3.3.5.4 Composition alignment

Lines 12 to 28 of the composition algorithm are responsible for arriving at an optimal composition from the trivial composition of a string. Because $\alpha\lambda\omega$ compositions always have a length of $2n+5$, where n is the length of the original string, all compositions of a given sequence can be completely "aligned" and can be stored in dimensionally identical

arrays with $2n+5$ elements, from the beginning to the end of the string composition part of the algorithm (lines 5 to 29). For example, in Table 21, trivial, non-trivial (i.e., intermediary) and optimal compositions of the sequence *Monday* are completely aligned. The purpose of the adaptive composition algorithm, i.e., arriving from a trivial composition to an optimal composition, can be achieved only through flip-up operations involving the ambiguous elements in a composition. The | symbol (which denotes the empty string) is always an ambiguous element in a composition and will always be replaced by appropriate elements such as *Mo* through flip up operations involving, in this particular case, the character *M*, the empty string | and the character *o*.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
triv.	|	(	|	M	|	o	|	n	|	d	|	a	|	y	|	)	|
inter.	|	(	(M	M	Mo	o	on	n	nd	d	da	a	ay	y	y)	)	|
optim.	|	(	(M	M	Mo	o	on	ond	onda	onday	onday)	nday)	day)	ay)	y)	)	|

Table 21. Aligned trivial, intermediary and optimal compositions of the sequence *Monday* in the context of the days of week names

For this walkthrough through the algorithm steps, for presentation purposes, we made use of a known, optimal composition in the context of the seven sequences. However, in case of previously unknown sequences, arriving at the optimal representation of a sequence represents the very last step of the algorithm. In Table 22, the actual substring poset elements required to represent the optimal composition are greyed. Most of these elements, excepting some additional elements in the optimal compositions, are ambiguous and represent substring patterns that are present in other sequences as well. For example, substrings such as *n*, *nd*, *nda*, *nday*, are ambiguous because they are also present in the sequence *Sunday*. By highlighting the unique patterns that are to be part of the constrained substring poset, one can get an indication of the amount of memory saved by the constraints applied to the substring poset. In order to store the optimal $\alpha\lambda\omega$ composition of the sequence *Monday* in the context of the weekday names, the number of nodes required is 26 (the count of grey cells in Table 22). The theoretical upper bound needed for the complete, unconstrained poset to store an $\alpha\lambda\omega$ composition of length 8 is $8(8+1)/2+1=37$. Applying the constraints has therefore yielded a saving of 10 compositional elements, or 10/37=27%, which is a significant number for such a short sequence. As demonstrated in the next chapter, for sequences of hundreds of characters,

the savings in compositional elements are much more impressive and make the difference between feasibility and non-feasibility of a certain application.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
optim.	|	(	(M	M	Mo	o	on	ond	onda	onday	onday)	nday)	day)	ay)	y)	)	|
4th	|	(	(M	M	Mo	o	on	ond	onda	onday	nday	nday)	day)	ay)	y)	)	|
3rd	|	(	(M	M	Mo	o	on	ond	onda	nda	nday	day	day)	ay)	y)	)	|
2nd	|	(	(M	M	Mo	o	on	ond	nd	nda	da	day	ay	ay)	y)	)	|
1st	|	(	(M	M	Mo	o	on	n	nd	d	da	a	ay	y	y)	)	|
triv.	|	(	|	M	|	o	|	n	|	d	|	a	|	y	|	)	|

Table 22. Alignment of the trivial, intermediary and optimal compositions of the sequence *Monday* in the context of the days of week names; the elements that form the substring poset corresponding to the composition are shown in grey

3.3.5.5 Non-optimal compositions examples

The algorithm also allows the possibility to achieve compositions which are non-optimal through the LPP and LPS ambiguity thresholds $\underline{a}$ and $\underline{a}$. By setting the ambiguity thresholds to values higher than 0 bit, ambiguous elements are allowed in some of the compositions (e.g., *Thursday* and *Saturday* in Table 23). Though, ambiguous such compositions will have lower ranks and memory requirements.

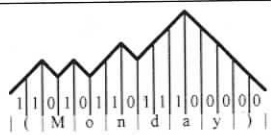
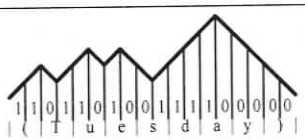
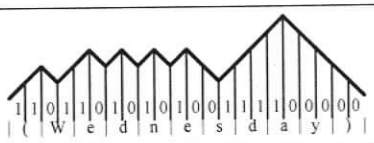
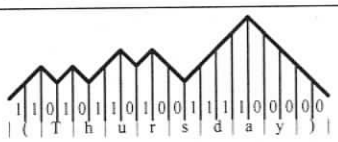
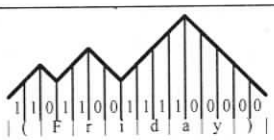
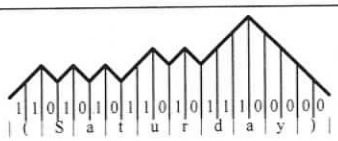
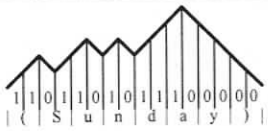
Length	Non-optimal composition	Dyck path	Amb	Rank
17	| ((M M Mo o on ond nd nda nday nday) day) ay) y)) |		22.3	36
19	| ((T T Tu Tue ue ues es s sd sda sday sday) day) ay) y)) |		24.3	41
23	| ((W W We Wed ed edn dn dne ne nes es s sd sda sday sday) day) ay) y)) |		23.3	51
21	| ((T T Th h hu hur ur urs rs s sd sda sday sday) day) ay) y)) |		25.3	44
17	| ((F F Fr Fri ri i id ida iday iday) day) ay) y)) |		21.3	36
21	| ((S S Sa a at t tu tur ur urd rd rda rday rday) day) ay) y)) |		26.3	44
17	| ((S S Su Sun un und nd nda nday nday) day) ay) y)) |		23.3	38

Table 23. Non-optimal compositions of the seven strings that name the seven days of week

Instilling a controlled amount of ambiguity of compositions, besides contributing to saving memory has an additional interesting side effect. For example, in Figure 48, by following the directed graphs paths for the compositions of the strings *Saturday* and *Thursday* one could easily deviate towards slight variations of the originals, such as *Satursday* and *Thurday*. Such deviations seem to possess a certain "naturalness" and could serve as an interesting example of "inventions" that such an associative memory model would be capable of.

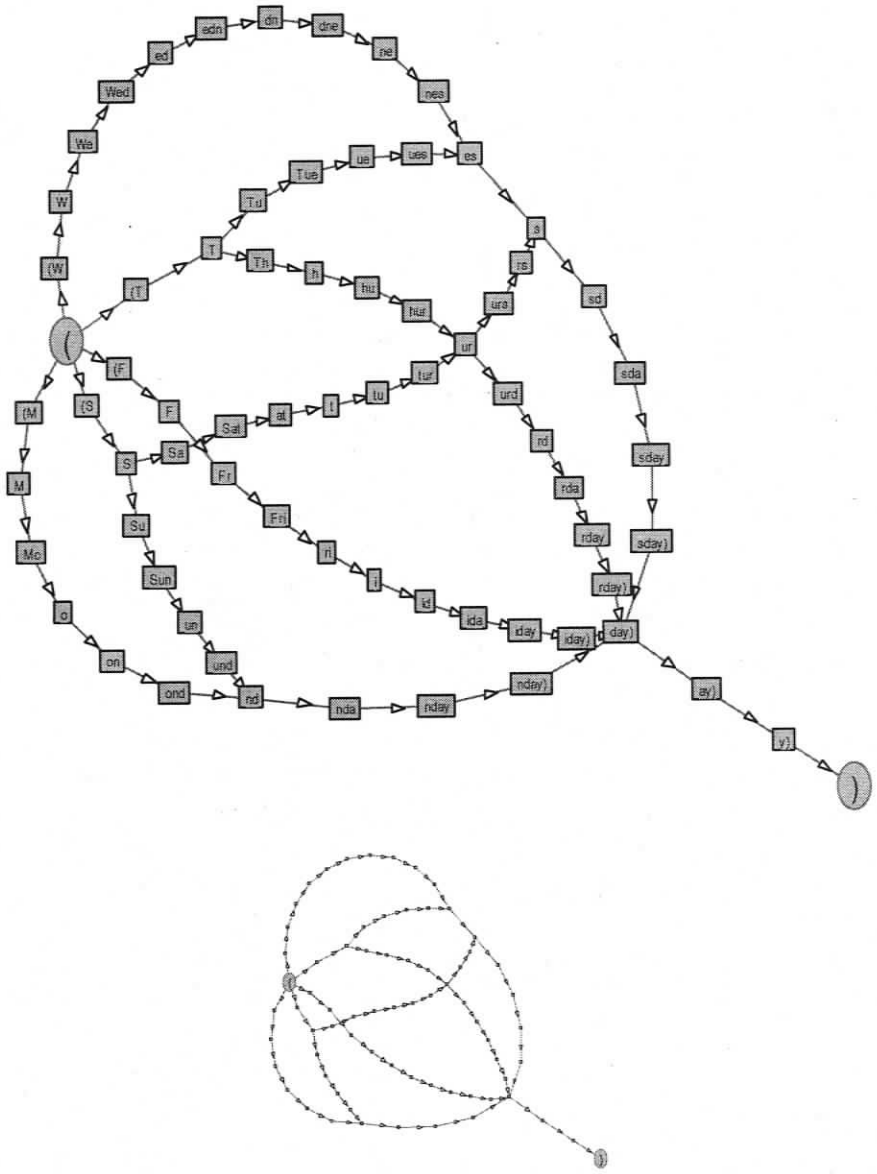


Figure 48. Depictions of non-optimal compositions (ambiguity thresholds $\underline{a}$ and $\underline{a}$ of 1 bit) of the seven strings that name the seven days of week; only elements that make the compositions are shown

In fact, it can be easily shown that by further increasing the ambiguity of the compositions (Figure 49), one could "invent" others, even more drastic variations such as *Turiday*, *Suriday*, *Thuriday*, *Sunesday*, *Mondneday*, *Satundnesday*, *Weday*, etc. Additional experiments done with other sets of sequences, such as the names of the US states, have also yielded interesting variations such as *Tennesota* and *Vermontana*.

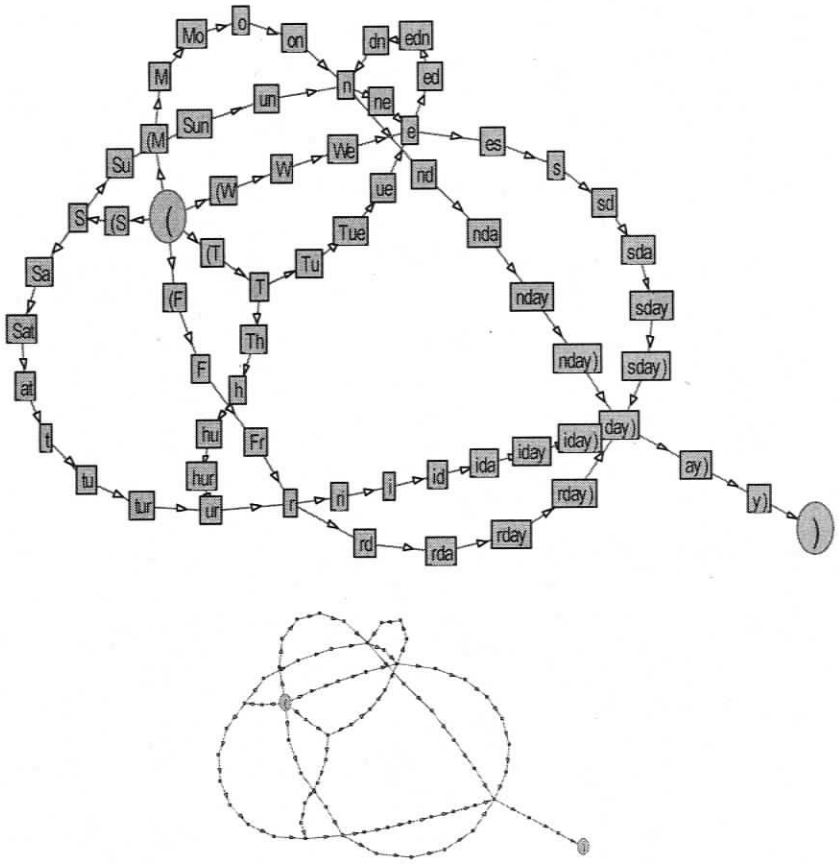


Figure 49. Depictions of non-optimal compositions (ambiguity thresholds $\underline{a}$ and $\underline{a}$ of $Log_2(3)=1.58$ bits) of the seven strings that name the seven days of week; only elements that make the compositions are shown

3.3.5.6 A possible mechanism for inventiveness

What we seem to have here is a mechanism that is able to create novel but "natural" compositional variations of existing patterns from actual real sequences, as a result of instilling a controlled amount of ambiguity into representations. One of the most important implications of this result is the possibility to explain and approach the difficult problem of case adaptation, an important part of Case Based Reasoning. Though not specifically addressed in this dissertation, this question is an important avenue for future research.

A second immediate question is whether this mechanism could serve as a possible explanation for inventiveness in general. The answer may be yes if one defines

inventiveness as the creation of a new artefact which should contain, at the same time, a certain amount of novelty as well as a fair amount of old features. The old, recognizable patterns taken from existing sequences would allow us to recognize and put in a context the "invention" while the new features will "tickle" our novelty detector and keep us interested and entertained. In addition, it looks like achieving a good balance between the amount of old is crucial: a lack of compositional novelty (i.e., too many recognizable, predictable, patterns) may prove uninteresting or boring, while too much novelty (i.e., lack of recognizable patterns) could cause us not understand/recognize the "invention" and ignore it the same way we ignore noise or randomness which by definition never repeat and have the property to be always new, patternless, full of novelty. To conclude, it may very well be that a successful natural extension of old ideas (i.e., innovation) could be achieved naturally by instilling a controlled amount of ambiguity in the representations of existing ideas. The implications of this finding however far reaching they might be for arts and science, are beyond the scope of this dissertation and hence the discussion will be limited to the remark that ambiguity, in appropriate (rather small) amounts, might actually be a positive thing.

3.3.6 The $\alpha\lambda\omega$ non-overlapping de-composition algorithm

The non-overlapping de-composition algorithm, as the names suggests, aims at de-composing optimal, overlapping, *useful compositions* of strings stored in constrained substring posets, into more ambiguous but non-overlapping, human readable, *usable compositions*. A goal of this algorithm is also the discovery of patterns and rules that allow meaningful, potentially compressed compositional representations of sequences. The specification of how to compose a sequence as well as others similar to that it, is equivalent to a "grammar" that generates those sequences. The task to be solved is therefore equivalent to a form of *grammar induction* which creates useful representations by changing the properties of the representation space in which sequences which are similar are represented. This is achieved by discovering sequence structure and hence reducing the algorithmic complexity of general, high dimensional representations to a more manageable, lower dimensional representations that live in a feature subspace.

The input and internal parameters used by the algorithm are:

- An $\alpha\lambda\omega$ deterministic, constrained substring poset P;
- An array C_i which holds the elements that make an overlapped composition of a string in P;
- A Boolean flag variable u to signal changes in string composition;
- Two dynamic arrays $\underset{\sim}{C}$ and $\underset{\sim}{C}$ which hold the elements of the prefix and suffix chains that are to be flipped down;
- Four integer index variables i, j, m and n.

$DE-COMPOSE(P,C)$

```
01    repeat
02          u := true;
03          for i := 0 to |C| - 3 do
04          begin
05                if ( φ(C[i+1]) > 0 ) then
06                begin
07                      if ( C[i+1] ≿ C[i] ) and ( C[i+1] ≿ C[i+2] ) then
08                      begin
09                            u := false;
10                            C := C[i+1] ≿ C[i] ≿ C[i−1] ≿ ... ≿ C[m];
11                            C := C[i+1] ≿ C[i+2] ≿ C[i+3] ≿ ... ≿ C[n]
12                            γ = 0 ; γ = 0 ;
13                            for j := i+1 downto m
                                    do γ = γ + ψ(C[j]) / ψ(C[j]) ;
14                            for j := i+1 to n do γ = γ + ψ(C[j]) / ψ(C[j]) ;
15                            if  γ/|C| < γ/|C|  then for j := i+2 downto m
                                    do C[j] := ↗(↖(C[j]))
16                            else for j := i+2 to n do C[j] := ↖(↗(C[j])) ;
17                      end;
18                end;
19          end;
20    until u;
```

Lines 01-20 are the main loop of the algorithm. As long as there are changes in the composition of any string as consequences of chain flip down operations (lines 7-17) this loop keeps iterating again through the elements of the composition C.

Lines 03-19 are the main **for** loop which iterates through every single 3-tuple of elements in C.

In **Line 05** it is made sure that only 3-tuples with a middle element having a rank higher than zero are processed further since 3-tuples that do contain the least element cannot be flipped down anyway.

Lines 07-17 are executed only for 3-tuples which have a valley configuration and whose middle element gives the overlap of the composition. This overlap is going to be removed through flip down operation on the prefix or suffix chains that form partially the descent and the ascent of the 3-tuple valley.

In **Line 10** the internal array $\underline{C}$ is initialized to the elements of the suffix chain in P that forms the descent of the 3-tuple valley. Because the total order of the chain is the opposite of the index order the array is listed in backward order (i.e., the index m is smaller than i+1).

In **Line 11** the internal array $\underline{C}$ is initialized to the elements of the prefix chain in P that forms the ascent of the 3-tuple valley. Because the total order of the chain matches the index order the array is listed in forward order (i.e., the index n is bigger than i+1).

Lines 12-14 contain the calculation of two important parameters:

- the prefix/suffix ambiguity ratio sum $\underline{\gamma}$ of the suffix chain and
- the suffix/prefix ambiguity ratio sum $\underline{\gamma}$ of the prefix chain;

The parameters are used in the chain flip down decision in Line 15.

Lines 15-16 contain the important decision on whether the next flip down must be made on either the prefix or the suffix chain. If the prefix/suffix ambiguity ratio sum $\underset{\sim}{\gamma}$ normalized by the cardinality of the suffix chain $\underset{\sim}{C}$ is smaller than the suffix/prefix ambiguity ratio sum $\underset{\sim}{\gamma}$ normalized by the cardinality of the prefix chain $\underset{\sim}{C}$, then $\underset{\sim}{C}$ is flipped down by replacing its elements (except its lowest element) with their LPD's. If the opposite is true, then $\underset{\sim}{C}$ is flipped down by replacing its elements (except its lowest element) with their LPD's. Either operation causes the overlap of the composition to be reduced by 1. The decision is local and is made only on the basis of the suffix and prefix ambiguity values of the elements of the suffix and the prefix chains.

In addition, and not part of the pseudo-code, an external routine verifies whether the current 3-tuple is a peak with a prefix and suffix ambiguity values are below their respective ambiguity thresholds. If this is so, then the peak is flipped down and the decomposition algorithm is repeated. This operation is necessary in order to create non-overlapping, easily readable, useful string compositions which possess controlled levels of ambiguity.

A slight modification of this algorithm which appears conceptually simpler instead of calculating the suffix/prefix and prefix/suffix ambiguity sums and normalizing them by the cardinality of the prefix and suffix chains, is to just determine the (winner) element with the maximum ratio as in the following:

```
12          γ = 0 ; γ = 0 ;
13          for j := i+1 downto m
                  do if γ < ψ(C[j])/ψ(C[j]) then γ := ψ(C[j])/ψ(C[j]) ;
14          for j := i+1 to n
                  do if γ < ψ(C[j])/ψ(C[j]) then γ := ψ(C[j])/ψ(C[j]) ;
```

and then perform the appropriate chain flip down operation.

```
15          if γ < γ then for j := i+2 downto m do C[j] := ↗(↖(C[j]))
16          else for j := i+2 to n do C[j] := ↖(↗(C[j]));
```

3.3.6.1 Non-overlapped compositions examples

The purpose of the de-composition algorithm is to arrive at a non-overlapped composition from the optimal composition. This is possible because the non-overlapped compositions can be perfectly aligned with their optimal counterparts (Table 24).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
optim.		(	(M	M	Mo	o	on	ond	onda	onday	onday)	nday)	day)	ay)	y)	)	
Non-o.		(		M		o		n	nd	nda	nday	day	day)	y		)	

Table 24. Aligned optimal and non-overlapping compositions of the sequence *Monday* in the context of the days of week names

As a result, non-overlapped compositions have lower ranks but are more ambiguous and hence can never be optimal. In addition, the intrinsic property of being composed of non-overlapping substrings makes such compositions easy to concatenate in "peaks and valleys" form (Table 25) which is equivalent to a rewrite rule in a formal grammar. Though the elements of such compositions can be concatenated into the original string by removing begin, end and empty string symbols, the compositions could also be displayed in a way that shows the particular chunking of a string.

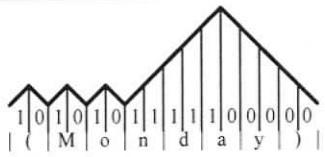
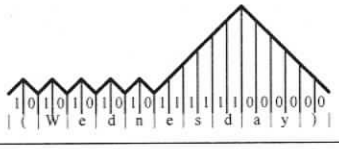
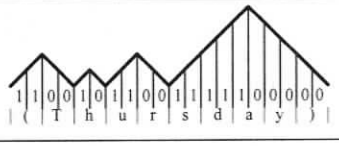
"peaks and valleys"	Non-overlapping composition	Dyck path	Amb	Rank
| (| M | o | nday) |	| (| M | o | n nd nda nday nday) day) ay) y)) |		53.4	28
| (T | u | esday) |	| ((T T | u | e es esd esda esday esday) sday) day) ay) y)) |		52.5	41
| (| W | e | d | n | esday) |	| (| W | e | d | n | e es esd esda esday esday) sday) day) ay) y)) |		80.9	41
| (T | h | ur | sday) |	| ((T T | h | u ur r | s sd sda sday sday) day) ay) y)) |		62.0	34
| (| F | r | i | day) |	| (| F | r | i | d da day day) ay) y)) |		65.2	20
| (S | a | t | ur | day) |	| ((S S | a | t | u ur r | d da day day) ay) y)) |		74.8	26
| (S | u | nday) |	| ((S S | u | n nd nda nday nday) day) ay) y)) |		50.5	30

Table 25. Non-overlapping compositions (ambiguity thresholds $\underrightarrow{a}$ and $\underleftarrow{a}$ of 1 bit) of the seven strings that name the seven days of week

3.3.6.2 Equivalence of non-overlapped compositions to formal grammars

From a formal grammar standpoint, deriving non-overlapped compositions is equivalent to specifying rewrite rules in which the chunks are represented by non-terminal symbols (Table 26).

Rewrite rule	Rule expansion
4 → d a	d|a
3 → 4 y	da|y
2 → 3)	day|)
1 → n 2	n|day)
5 → (M o 1	(|M|o|nday)
6 → (S	(|S
7 → u r	u|r
8 → 6 a t 7 2	(S|a|t|ur|day)
9 → 6 u 1	(S|u|nday)
10 → (T	(|T
12 → s 2	s|day)
11 → e 12	e|sday)
13 → 10 u 11	(T|u|esday)
14 → 10 h 7 12	(T|h|ur|sday)
15 → (W e d n 11	(|W|e|d|n|esday)
16 → (F r i 2	(|F|r|i|day)

Table 26. Example of machine induced formal grammar (ambiguity thresholds $\underline{a}$ and $\underline{a}$ of 1.0 bit) of the seven strings that name the seven days of week; the rules that contain only terminals are greyed and the chunks in the rule expansions are explicitly delimited by | symbols

The chunks can also be regarded as features that could be used to dynamically search and organize the representation in context/content-dependent, meaningful categories, a capability especially useful when the strings form sets of hundreds and thousands of items. As a simple example, given the non-overlapped compositions in Table 25 and their formal grammar representation in Table 26, the way to categorize the days of week according to their content (and context) is to create an *inverted representation* which indexes on the features (e.g., **-nday)**, **-sday)**) and leads naturally to a multiple hierarchy showing the relationships between items and their features (e.g., **Sunday** and **–nday**) as well as between features themselves (e.g., **-sday)** and **–esday)**) (Table 27). This multiple hierarchy is identical to the general concepts of *inverted file* (Knuth 1997) and *inverted index* (Manning and Schütze 1999) which are specific to information retrieval. As a consequence, representations as in Table 26 can be successfully used in a search and retrieval on secondary keys.

• **(T-**	**10 → (T**
• (Tuesday)	13 → 10 u 11
• (Thursday)	14 → 10 h 7 12
• **(S-**	**6 → (S**
• (Saturday)	8 → 6 a t 7 2
• (Sunday)	9 → 6 u 1
• **-ur-**	**7 → u r**
• (Saturday)	8 → 6 a t 7 2
• (Thursday)	14 → 10 h 7 12
• **-da-**	**4 → d a**
• **-day)**	**3 → 4 y, 2 → 3)**
• (Friday)	16 → (F r i 2
• (Saturday)	8 → 6 a t 7 2
• **-nday)**	**1 → n 2**
• (Monday)	5 → (M o 1
• (Sunday)	9 → 6 u 1
• **-sday)**	**12 → s 2**
• (Thursday)	14 → 10 h 7 12
• **-esday**	**11 → e 12**
• (Tuesday)	13 → 10 u 11
• (Wednesday)	15 → (W e d n 11

Table 27. Example of inverted (feature indexed) multiple hierarchy derived from the machine induced formal grammar in Table 26; the chunks corresponding to rules that contain only terminals (greyed) are the first level of the hierarchy

To summarize, this particular chunking procedure has allowed us to dynamically create a context/content-dependent categorization of the seven strings that name the seven days of week. According to this particular categorization, all strings are of type **–day)**, some of which are of type **–nday)** and some **–sday)**. Those of type **–sday)** can also be **-esdays)**. Because some strings belong to other categories (e.g., those containing **-ur-**, **(T-** and **(S-**) the hierarchy is multiple. Additional experiments on dynamic, context/content-dependent induction of multiple hierarchies, are presented in the following chapter.

3.4 IMPLEMENTATION CONSIDERATIONS

In this section the discussion of implementation issues begins from the perspective of the four properties of concept representation spaces. The discussion continues with an estimation of the algorithmic complexity of the DDAM model and ends with a description of the visualization techniques used to create many of the graphical representations in the dissertation.

The four properties of natural sequences and concept space representations, i.e., multidimensionality (related to sequence length), sparseness, dynamicity and similarity based organization have all been addressed by the constrained substring poset model which, essentially, is just a huge directed graph, subject to any implementation approach for directed graphs. However, an efficient implementation which can successfully deal with all four fundamental properties of concept space representations would have to abide to the arguments in Chapter 2 which prescribe that certain approaches are more efficient than others. The DDAM model being essentially a graph makes full use of these insights.

3.4.1 Sparseness and dynamicity

Dealing efficiently with two of the afore-mentioned properties, i.e., sparseness and dynamicity, is highly dependent on a certain implementation approach of the DDAM model. This approach is also mentioned by others (Kanerva 1988) in the context of implementing associative memories, goes against all object oriented programming (OOP) principles and consists of making extensive use of pointers and linked lists, as prescribed in Chapter 2 of this dissertation. Though prone to programming errors, the use of pointers and linked lists seems to be the only way to attain the dynamicity required by the constrained substring poset composition/decomposition algorithms working on compositional representations whose extreme sparseness renders array-based approaches too inefficient.

3.4.2 High dimensionality

As prescribed in Chapter 2, the approach to overcome space complexity of algorithms was by using hierarchical models and putting to use their inherent data compression properties. The unconstrained substring poset model introduced is highly hierarchical but at the same time inherently redundant: all substrings of all strings in a language are represented explicitly by a *node* in the substring poset. Nodes are linked by directed *edges* and organized in *sections* that comprise nodes with an identical rank in the substring poset. The constrained substring poset removes some of the redundancy by not representing the many substrings (in case of sparse representations of natural sequences) that do not contribute to the overall reduction of the ambiguity of representations. These design features of the model are generally applicable and hence oblivious to any kind of prior knowledge about the possible input, which can therefore be anything. However, though not without caveats, some additional improvements have been possible by making use of the knowledge about the input data. For example, in case of textual data, an additional way to increase the hierarchicality of the model and shorten the length of the compositions was to make use of the various separators existent in texts (e.g., blank, commas, periods, etc.) that are known to delimitate tokens. Since tokens are relatively short in length and since the efficiency of the model is highly dependent on the length of representations, this was an attractive approach that allowed scaling up the model to useful languages of thousands of strings, by stacking constrained substring posets in multiple layer structures such as the one in Figure 50.

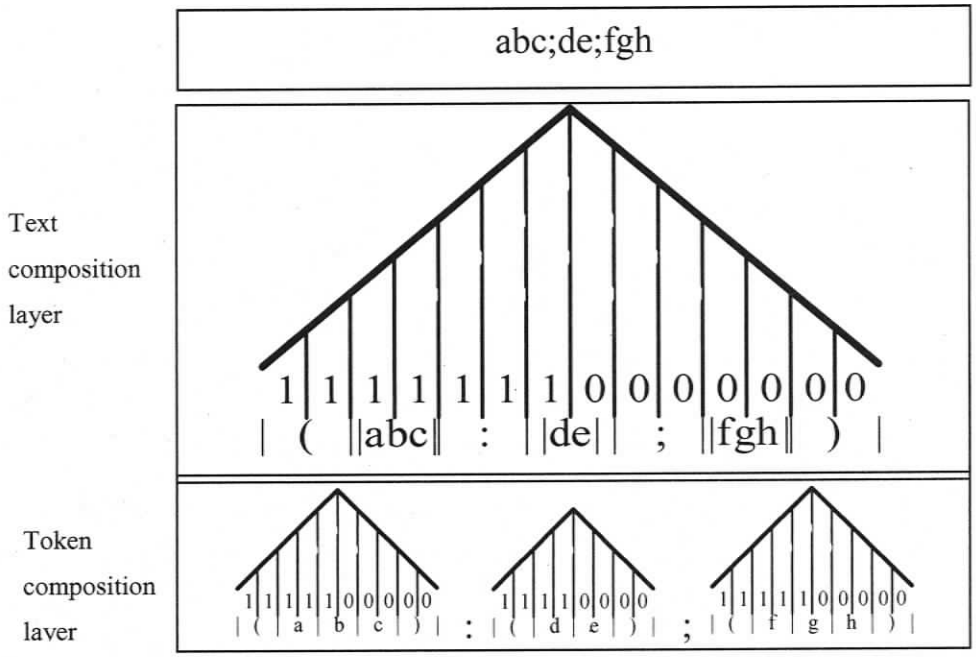


Figure 50. Illustration of the multilayer, hierarchical layout of the associative memory model prototype; the token composition layer contains compositions of tokens (sequences of letters and/or numbers) separated by non token symbols (e.g., colon, semi-colon, comma, period, etc.), the text composition layer contains phrases and longer text patterns and the topmost layer stores the final composition of a sequence

As a consequence of this approach and an important issue that is going to be addressed in future research, is the fact that this approach to layer separation is context-independent and does not discriminate between, for example, a period at the end of a sentence and a period after a capital letter in an acronym, leading to a somewhat artificial separation of language elements that subsequently may hinder processing.

3.4.3 Algorithmic complexity

Even when using hierarchical approaches, it has been estimated in this chapter that the upper bounds of the time and space computational complexities of the constrained substring poset composition algorithm are quadratic in the worst case. This translates in the unfeasibility of naïve implementations of constrained poset models such as the one in Annex 2. In particular, the quadratic space complexity seems to be the most worrisome.

As shown in Chapter 2, the associative memory model introduced in this dissertation can also be regarded as a space-time computational complexity trade-off which is able to

attain reasonable processing times of very long string compositions at the expense of a quadratic space complexity, in the worst case. However, it has also been argued that, contingent on appropriate implementations, the spatio-temporal complexity of the models could be determined only by the length of the strings in a language and by their algorithmic properties: the sparser the representation space of the strings the more efficient the associative memory model and vice versa, the more compact the representation space the less efficient the model. Therefore, while demanding in memory, this approach could lead to the situation where processing time can be considered virtually independent of the number of strings in a data set, especially when the strings form a highly sparse representation space. At the same time, it allows one to liken the model to a model of a human brain whose spatial complexity is very high whose retrieval time of memories does not seem to increase drastically with age and hence is virtually independent of number of items stored in it.

Therefore, the combination of implementation approaches and selection of input data could turn the model into a feasible approach, at least for small size problems such as the analysis of textual data comprising, for example several thousands article abstracts. Furthermore, empirical results on medical textual dataset typical to Medical Informatics applications suggest that, in case of sufficiently sparse patterns, the space complexity becomes close to a linear function of the input length. Since the dynamically allocated memory is used for only three types of data structures (i.e., nodes, edges and sections), the memory usage of the model was estimated separately, for each type of data structure, as a percent the maximum allocated memory which was in the order of 400 Megabytes of RAM. Represented versus input length, the memory necessary to represent the optimal compositions of over 30 thousands ICD10 (International Classification of Disease version 10) strings, demonstrates a curve that is almost linear (Figure 51).

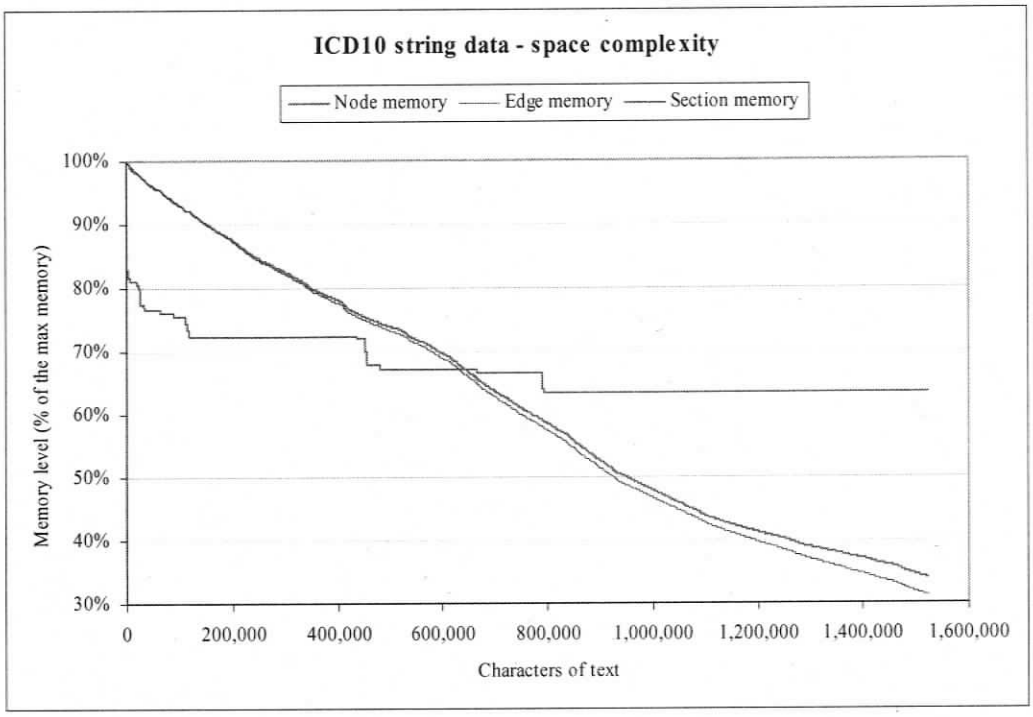


Figure 51. Empirical results estimating the memory requirements of optimal compositions of all 30,000 ICD10 (International Classification of Disease version 10) strings; the memory usage by poset nodes, edges and sections is shown as a percent of the maximal memory allocated for such structures which was, in total, around 400 Mbytes of random access memory

This validates the usefulness of the multi-layered constrained substring poset model and suggests that, in the long run, the approach could account for considerable savings in memory that could make a significant difference in the application of the model for real world applications.

3.4.4 Similarity based retrieval

Arguably, similarity based retrieval is one of the most important functions of the DDAM model, a major objective of this research and of demonstrated significance to Medical Informatics. This function is also fully supported by the thesis of this dissertation which advocates a similarity based organization of representations and was already demonstrated, on a didactic, limited example, in Chapter 2 of this dissertation. The constrained substring poset structure defined in this chapter, which is the building block of the DDAM model, possesses a natural propensity to represent and retrieve information by similarity, demonstrated by images and examples. Therefore, as far as the

implementation of similarity based representation and retrieval are concerned, implementing the constrained substring poset structure provides a good perspective and a significant advancement toward the important end of similarity based retrieval.

In this dissertation, the similarity based representation is fully defined and formally described by the adaptive $\alpha\lambda\omega$ composition and decomposition algorithms introduced this chapter. However, the similarity based retrieval algorithm remains an experimental algorithm, introduced only informally and empirically and whose complete formal description and analysis are ongoing research.

3.4.5 Visualization techniques

Achieving aesthetic layouts of the directed graph structures depicted in this dissertation has been possible through techniques based on the force directed algorithms (Battista, Eades et al. 1994; Frick, Ludwig et al. 1995; Eades and Huang 2000; Friedrich and Eades 2002). Such depictions of substring posets structures have been possible only because of the relatively small sizes of the languages represented (e.g., names of weekdays or month names in Figure 52.

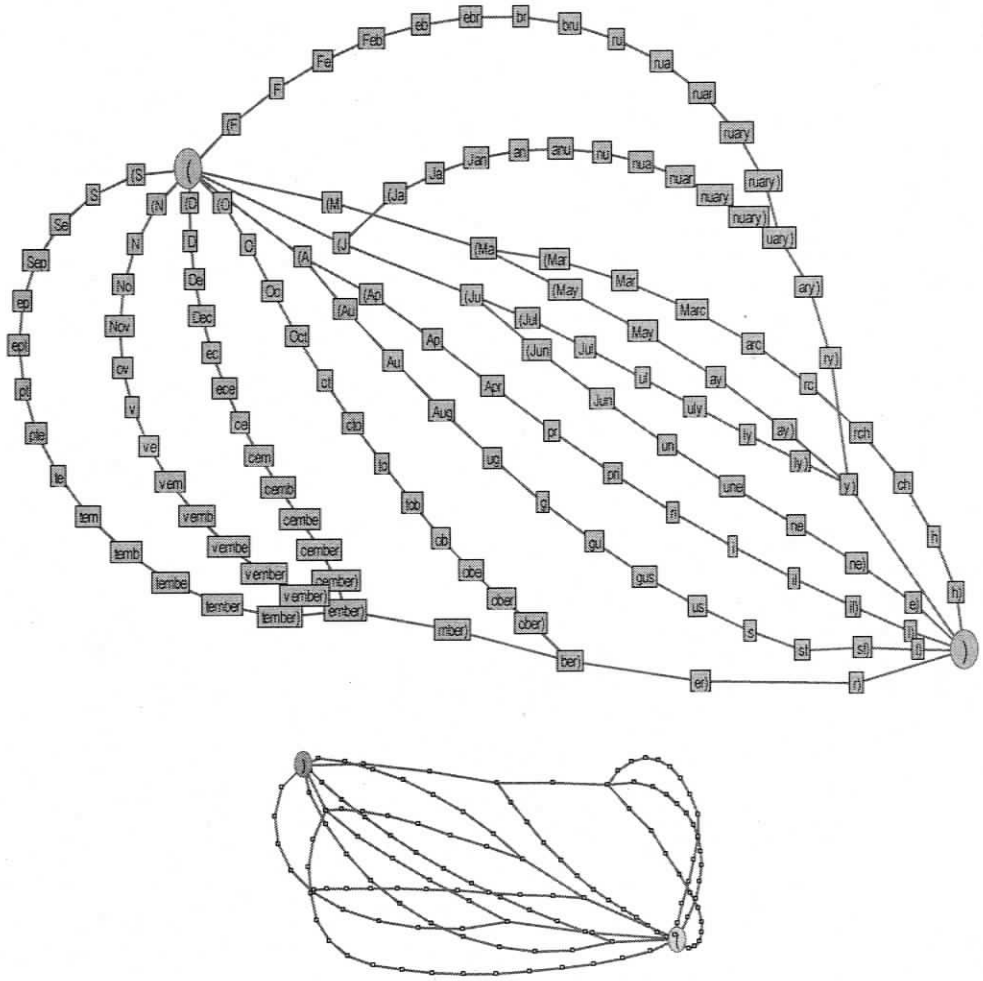


Figure 52. Depictions of compositions of the month names; only elements that make the compositions are shown

The recognized conceptual overlap of self-organizing maps and force directed graph layout optimization (Frick, Ludwig et al. 1995) suggests that such techniques used in conjunction with the natural similarity based organization of substring posets may achieve structural and functional equivalence to self-organizing maps (Kohonen 2001). The equivalence of the DDAM model with self-organizing maps was already discussed but a direct comparison remains out of the scope of this dissertation.

Chapter 4 CONCEPT SPACE REPRESENTATION EXPERIMENTS

This chapter contains the experimental results and evaluation of the DDAM model through information processing experiments and comparison with the results of existing models on similar tasks. The experiments range from artificial to natural sequence processing and consist of various pattern discovery, grammar induction and natural language processing tasks.

4.1 INTRODUCTION

This introductory section is a short description of the methodology and of the global focus of the evaluation approach. The models employed in the first three experiments are the two variants of DDAM model (the simple constrained substring poset – DDAM-1 and the constrained substring model – DDAM-2).

The two variants of DDAM model, in non-naïve, multi-layered implementations, will be referred to from now on by the acronyms DDAM-1 and DDAM-2 and will be evaluated in a series of experiments, most of which follow experimental setups found in literature, such as, for example (Elman 1990). Where possible, the evaluation methodology and criteria of success also involve existing models published in the literature, performing similar or identical tasks. The description of each experiment begins with presentations of data sources and criteria for success and continues with the pattern discovery and recognition tasks specific to a particular experiment.

The definition of a *significant pattern* or significant regularity given in Chapter 2 suggested that patterns with longer descriptions are more significant than shorter ones. This distinction is important since the focus of the evaluation will consist, among other things, of assessing the ability of the approaches to acquire significant regularities, i.e., whose lengths are as long as possible and whose probability to occur randomly is consequently as low as possible.

4.2 EXPERIMENTS WITH ARTIFICIAL SEQUENCES

This section comprises experiments conducted to explore the pattern discovery performances of DDAM models on artificial sequences, with a focus on the acquisition of significant regularities with long descriptions. The artificial sequences used in experiments are built by combining lexical items from artificial lexicons, in a manner similar to the descriptions available in (Elman 1990).

As described in Chapter 2 in the context of the review of word segmentation tasks, artificial data comprising nonsensical lexical items are oblivious to any form of semantic processing. This property could be thought of as an ability of this data to bring pattern discovery and recognition mechanisms down to a more primitive information-processing model. Primitive, simplified models could provide additional insights into processing mechanisms that could help with the development of pattern discovery and recognition algorithms suitable for lexical acquisition and associative information retrieval, such as the DDAM model. In addition, the fact that statistical properties of artificial texts are easy to control and the experimental results of such tasks are relatively easy to quantify, compare and discuss, makes artificial text useful for the evaluation of segmentation models and algorithms (Wolff 1975), (Elman 1990).

4.2.1 Experiment #1

4.2.1.1 Data sources

For the first experiment a restricted lexicon of 15 "words" of 3 characters each (Table 28) was used.

vir	san	huj	pos	lic	teg	zop	mor	lox	mar	vez	cow	fan	bac	kil

Table 28. The lexicon use to create the artificial data for experiment #1

A language of 15 sequences was created; each of them is 36 characters long and comprises 12 randomly selected "words" from the lexicon. No separators are used at the word boundary (Table 29).

virvezloxmorhujsantegvirzopsancowlic
poshujhujloxzopmarposmorcowfanvirlic
morfanbacsansanfanzopvirposkilposfan
loxlicfanloxhujlictegzopmarkilbacsan
licsanzoptegsanmorposzopsanmormorhuj
virloxzopfanlicsancowtegtegmorposlic
posloxzopkilvezhujcowhujsanlichujzop
posbacmormarfanzopmarloxvirhujvezsan
sanmarloxmormarvezbacsanbacsanfancow
fanmorsanfancowtegvezposhujtegkillox
markiltegloxvirloxmormormorcowfanhuj
tegposzoplicloxkilposlicmarmorteglox
backilcowloxposmarloxlicloxvirtegmar
virvirsancowmarlicbacmarmarposbacvir
morhujvirzoploxvirtegteglicvirsanhuj

Table 29. Artificial input sequences for experiment #1.

4.2.1.2 Criteria for evaluation

The purpose of creation of artificial training data and the expected outcomes are precisely set. The artificial lexica have been defined upfront, all lexical entries are known and one can therefore easily measure the performance of pattern discovery algorithms that attempt to discover lexical entries from artificial data. In addition, it has been possible to directly compare the results of the following experiment with those of the ADIOS (Automatic DIstillation Of Structure) (Solan, Horn et al. 2004; Edelman, Solan et al. 2005) and SEQUITUR (Nevill-Manning and Witten 1997), two relevant approaches to DDAM and which have been reviewed in detail in Chapter 2.

4.2.1.3 Lexical acquisition by DDAM-1

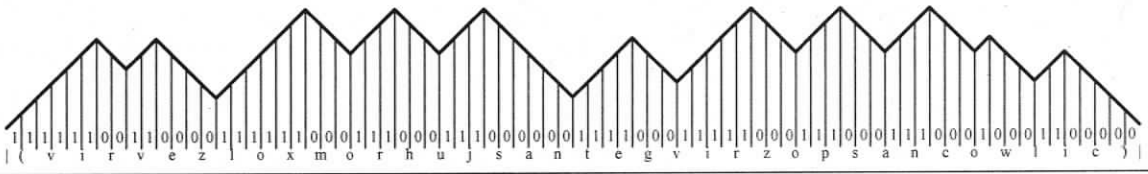
The sequences in Table 28 are fed into the DDAM-1 model, which builds a hierarchical representation of the input using normal constraints (see definition of normal constraints in Chapter 3). The model is not explicitly exposed to the lexicon. In the representation, the DDAM-1 model "discovers" the character patterns that make each word of the lexicon (unsupervised learning). The patterns discovered are shown in Table 30 together with their estimated counts. The second layer of the DDAM-1 shows all and nothing but the 15 "words" used to create the input. The reason the model did not "discover" the entire lexicon in the first layer has to do with the ambiguity in some patterns which contain identical symbols (e.g., *a* in *san* and *fan* or *o* in *cow*, *lox* and *zop*). Non-ambiguous patterns (e.g., *huj*) or patterns with little ambiguity (e.g., *teg*) are fully represented in the first layer. Constraint relaxation yields the expected result that DDAM-

1 is able to pick-up higher order, longer patterns that would intuitively correspond to phrases (e.g., *morhuj*, *zopmar*, *mormor*, *loxvir*, *morcow*, *bacsan*).

Layer 1	Pattern	z	w	f	b	k	ve	an	co	ac	il	vir	san	huj	pos	lic	teg	zop	mor	lox	mar
	Count	5	9	11	8	7	5	11	9	8	7	14	17	13	13	13	13	12	15	17	13
Layer 2	Pattern	vir	san	huj	pos	lic	teg	zop	mor	lox	mar	vez	cow	fan	bac	kil					
	Count	14	17	13	13	13	13	12	15	17	13	5	9	11	8	7					

Table 30. Patterns acquired in the first two layers of the DDAM-1 model; the second layer contains the 15 "words" lexicon.

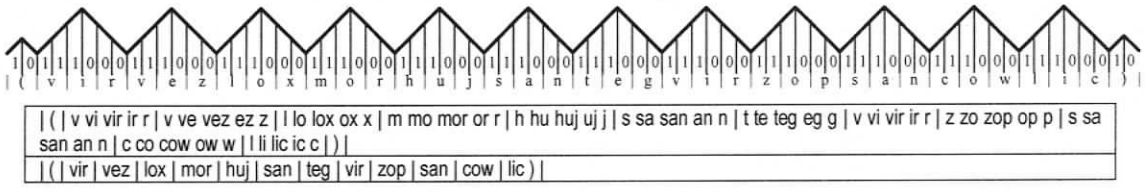
4.2.1.4 Lexical acquisition by DDAM-2



| ((v (vi (vir (virv (virve virve irve irvez irvezl rvezl vezl ezl zl zlo zlox zloxm zloxmo zloxmor zloxmorh loxmorh oxmorh xmorh xmorhu xmorhuj xmorhujs morhujs orhujs rhujs rhujsa rhujsan rhujsant hujsant ujsant jsant sant ant nt nte nteg ntegv ntegvi tegvi egvi gvi gvir gvirz gvirzo gvirzop gvirzops virzops irzops rzops rzopsa rzopsan rzopsanc zopsanc opsanc psanc psanco psancow psancowl sancowl ancowl ncowl ncowli cowli owli wli wlic wlic) lic) ic) c)) |

Figure 53. Dyck path of the optimal composition of the first string in Table 29; the grey dots mark the elements with high suffix-prefix ambiguity ratios at which non-overlapped compositions are likely to be derived

The DDAM-2 model is performing equally well on this data and is able to create optimal compositions (Figure 53) and to derive from them non-overlapped compositions with various, predefined threshold levels of ambiguity such as in Figure 54.



| (| v vi vir ir r | v ve vez ez z | l lo lox ox x | m mo mor or r | h hu huj uj j | s sa san an n | t te teg eg g | v vi vir ir r | z zo zop op p | s sa san an n | c co cow ow w | l li lic ic c |) |

| (| vir | vez | lox | mor | huj | san | teg | vir | zop | san | cow | lic) |

Figure 54. Dyck path of a non-overlapped composition (ambiguity thresholds $\underline{a} = 2$ bit, $\underline{a} = 2$ bit) of the first string in Table 29; the first row shows the actual composition while the second shows the "peaks and valleys" representation

Expectedly, for lower values of the ambiguity thresholds (e.g., 1), the acquisition shows higher-rank regularities that correspond to phrases, such as in Table 31.

vir | vez | lox | morhuj | san | teg | virzop | sancow | lic
poshuj | huj | loxzop | marpos | morcowfan | vir | lic
mor | fan | bacsan | sanfan | zop | vir | pos | kilpos | fan
loxlic | fan | lox | huj | lic | teg | zopmar | kil | bacsan
licsan | zop | teg | sanmor | poszop | san | mormor | huj
vir | loxzop | fan | licsan | cow | tegteg | morpos | lic
pos | loxzop | kil | vez | huj | cow | hujsan | lic | huj | zop
pos | bac | mormar | fan | zopmar | loxvir | huj | vez | san
san | marlox | mormar | vez | bacsan | bacsan | fancow
fan | mor | sanfancow | teg | vez | poshuj | teg | kil | lox
markil | teg | loxvir | lox | mormor | morcowfan | huj
teg | poszop | liclox | kilpos | lic | mar | mor | teglox
bac | kil | cow | lox | pos | marlox | lic | loxvirteg | mar
vir | vir | sancow | mar | lic | bac | mar | mar | posbac | vir
morhuj | virzop | loxvirteg | teg | lic | virsan | huj

Table 31. "Peaks only" representation of non-overlapped compositions (ambiguity thresholds $\underline{a} = 0$ bit, $\underline{a} = 0$ bit) of the strings in Table 29; the compositions contain phrases such as "morhuj", "virzop", etc.

The results of DDAM-2 are directly comparable to those of ADIOS and SEQUITUR and show improvements in the acquisition of multiword phrases (i.e., higher rank regularities).

Phrase	DDAM-2	ADIOS	SEQUITUR	Phrase	DDAM-2	ADIOS	SEQUITUR
bacsan	•(4)	•	•(4)	mormor	•(2)	•	
fancow	•(1)		•(2)	morpos	•(1)		•(2)
fanzop			•(2)	posbac	•(1)		•(2)
hujsan	•(1)			poshuj	•(2)		
kilpos	•(2)		•(2)	poszop	•(2)		
liclox	•(1)		•(2)	sancow	•(2)	•	
licsan	•(2)		•(2)	sanfan	•(1)		
loxlic	•(1)			sanfancow	•(1)		
loxvir	•(2)	•		sanmor	•(1)		
loxvirteg	•(2)			teglox	•(1)		•(2)
loxzop	•(3)		•(2)	tegteg	•(1)		
markil	•(1)			virsan	•(1)		•(2)
marlox	•(2)		•(3)	virteg			•(2)
marpos	•(1)			virzop	•(2)		
morcowfan	•(2)		•(2)	zopmar	•(2)	•	•(2)
morhuj	•(2)		•(3)	zopsan			•(2)
mormar	•(2)		•(2)				

Table 32. Cumulated list of multiword phrases discovered by DDAM-2, ADIOS and SEQUITUR showing dots and phrase counts for each phrase, as acquired by each model; ADIOS picks up very few multiword patterns and SEQUITUR misses on some significant chunks which would actually satisfy its "rule utility" constraint

The ADIOS model used in the MEX (Motif Extraction) mode with a relaxed p=1.0 value, on the same data but pre-processed appropriately, was able to form rules for all the 3-letter patterns words (Table 33), but picked up only five additional higher-rank regularities (*loxvir*, *zopmar*, *mormor*, *sancow*, *bacsan*).

ID	Pattern	Expansion
P26	(v,i,r)	vir
P27	(h,u,j)	huj
P28	(m,o,r)	mor
P29	(l,o,x)	lox
P30	(l,i,c)	lic
P31	(t,e,g)	teg
P32	(z,o)	zo
P33	(m,a,r)	mar
P34	(a,n)	an
P35	(p,o,s)	pos
P36	(c,o,w)	cow
P37	(P32,p)	zop
P38	(b,a,c)	bac
P39	(s,P34)	san
P40	(P29,P26)	loxvir
P41	(v,e,z)	vez
P42	(f,P34)	fan
P43	(k,i,l)	kil
P44	(P37,P33)	zopmar
P45	(P28,P28)	mormor
P46	(P39,P36)	sancow
P47	(P38,P39)	bacsan

Table 33. ADIOS table of extracted patterns showing rules for only 3 letter patterns

The algorithm failed to capture higher-rank regularities either purposely due to design constraints or as a fundamental limitation of the approach. Whatever the case, this significantly limits the application of the model, especially to tasks such as phrase recognition. It is also likely that ADIOS achieves pattern discovery by multiple passes through the entire data set, a feature that differentiates it from SEQUITUR which is a one-pass, linear complexity algorithm, as well as from DDAM-2 which, in its non-trivial implementation, performs a processing that is local to the composition of the current sequence and to its similarity neighbourhood (i.e., including only the sequences that share similarities) which a reduced neighbourhood, especially for sparse representation spaces such as those of natural sequences.

vir | vez | lox | morhuj | san | teg | vir | zopsan | cow | lic
pos | huj | huj | lox | zopmar | pos | morcowfan | vir | lic
mor | fan | bacsan | san | fanzop | vir | pos | kilpos | fan
lox | lic | fan | lox | huj | lic | teg | zopmar | kil | bacsan
licsan | zop | teg | san | morpos | zopsan | mor | morhuj
vir | loxzop | fan | licsan | cow | teg | teg | morpos | lic
pos | loxzop | kil | vez | huj | cow | huj | san | lic | huj | zop
posbac | mormar | fanzop | marlox | vir | huj | vez | san
san | marlox | mormar | vez | bacsan | bacsan | fancow
fan | mor | san | fancow | teg | vez | pos | huj | teg | kil | lox
mar | kil | teglox | vir | lox | mor | mor | morcowfan | huj
teg | pos | zop | liclox | kilpos | lic | mar | mor | teglox
bac | kil | cow | lox | pos | marlox | liclox | virteg | mar
vir | virsan | cow | mar | lic | bac | mar | mar | posbac | vir
morhuj | vir | zop | lox | virteg | teg | lic | virsan | huj

Table 34. SEQUITUR compositions of the strings in Table 29

The SEQUITUR model is able to acquire only certain multiword sequences, some of which seem to hinder the subsequent acquisition of other multiword sequences that would actually satisfy the "rule utility" constraint (e.g., *loxvir*, *loxvirteg*, *mormor*, *poshuj*, etc.) since they occur at least twice in the input. In fact all additional multiword phrases acquired by the DDAM-2 model appear at least twice in the input. By the early commitment to certain chunks (Hopkins 1999) and due to what appears to be again a local minimum problem, SEQUITUR is missing on the acquisition of some significant (i.e., long) chunks and arrives at a representation that may be locally appropriate but not optimal in the larger context. From this perspective the DDAM-2 model is superior in its ability to pickup many additional significant regularities.

4.2.2 Experiment #2

4.2.2.1 Data sources

The second experiment is very similar to the previous except that the lexicon was enriched with an additional 15 "words." The additional lexical items are 4-letter patterns derived from the original lexicon (e.g., *heuj* from *huj*) (Table 35), in order to increase the ambiguity of the input and hence the difficulty of the pattern acquisition task.

mar	lic	san	vir	teg	vez	huj	mor	lox	cow	kil	zop	pos	fan	bac
maer	lioc	saen	viur	teag	veez	heuj	meor	loix	cuow	kiel	zaop	pois	faon	baic

Table 35. The lexicon use to create the artificial data for the second experiment is derived from that of experiment #1

The input data created contains 30 sequences of variable length, each with 15 randomly selected words from the new lexicon written without separators at word boundary (Table 36).

```
liocvezposposcowlicmarliocmaerloixmarsaencuowliockil
baicposmarbaicmarmarkielpoissaenfanzaopcuowteagmorcow
virkilmortegfaonliczopkilmaercowvircuowzaophujteg
meorsanfanbaicfaonteagkielviurkielcuowteagbaiclicpoissaen
liczoploxvirzaopsancuowveezhujzaopposloixviurcuowmar
veezloixlicfanteagposmaercowbacloxveeztegzaoptegmeor
tegcowvirzopfaonvezvirsanfaonposheujteagheujsaenteag
faonsanviurmarkielmaerviurloxloixzaopbaicmeorsaensaenhuj
maermeorsaenmeorloixtegmarmeorfancuowfaonfaonkielkielhuj
sankillicvirmarlicmarmarviurteagfaonsansaenloxpos
saenzopzoploixcowkilvirsanliocpoistegvezpoiscuowvez
baictegkielfaonsanfanmaervezloixbaicpoismeorcuowmarbaic
veezkilcuowcuowheujkilmorzoploixkielloixheujsantegbaic
loixsanmaerliocbacvezposliocteagliccowcuowbaickielvez
vircowvezzaopbaicmarlicposmarmorsaenvirzopmaercuow
heujcowmorloxliocpoiscowviurbaicvirvezcowkielzaopmar
killiocsaenviurmeorsanloixcowfansaenbaiccuowteagmeorheuj
tegteaghujtegheujsaenfaonzaopmeorpoisheujbaicvirpoislioc
viurheujvirzaoptegvirsanliockielkilviurpossanmorfaon
tegsanteghujpoismarcuowtegtegteagmorbaclicloxcow
vezfaoncuowbachujcowliczopposzopsaenmarcuowzaopsaen
viurmeorloixkielfanlicmaervezbaiczopbaicsanpoisposmar
viurcowliockilfankielcuowvezliocviurfaonmarposbaicveez
viurheujhujlicliocfanfanmarfaonposmormorzopmeormaer
posviurmaerposkielcowpoiscowmeorkilzopzaopzaopkielmaer
viurvirsaensaenveezpossaenvezveezlicposkielmaerliochuj
viurbaiclicfanvezloixbacpoiszopcuowkilbacvirmeorvir
vezteagtegkielkielteagloixmarloixbaiczopliocvirloxcow
baicmaersancuowloixloxheujmaermarloxcuowcowmeorheujmeor
poisloixkilzaopsanveezmeorkilkielteagvirhujcowsaenlic
```

Table 36. Artificial input sequences for experiment #2.

4.2.2.2 Criteria of success

As for the first experiment, criteria for success are the complete discovery of lexical entries in Table 35 from the artificial data, as well as the discovery of additional higher-rank regularities. This is followed by a direct comparison of the results with those of the ADIOS and SEQUITUR.

4.2.2.3 Lexical acquisition by DDAM-1

Under normal constraints, the DDAM-1 model performs very well and discovers the entire lexicon in the third layer (Table 37). Slight constraint relaxation yields two-word phrases such as *liczop*, *faonsan*, *posmar*, *virsan*, *licfan*, *liockil*, *cuowteag*, *maervez*, *kielmaer*, etc. The fact that 3-word patterns are not found is also interesting and gives a hint on the properties of input data, which indeed, upon quick visual inspection, does not seem to contain 3-word phrases.

Layer 1	l	c	z	s	w	m	r	x	n	f	h	po	ow	lo	ba	ki
	an	op	or	vi	te	me	ag	ur	ve	ez	uj	he	mar	lic	mae	aop
	sae	fao	san	vir	poi	teg	loi	cuo	kie	bai	vez	lioc				
Layer 2	c	ba	mar	lic	san	vir	teg	vez	lioc	huj	pois	cuow	teag	mor	lox	cow
	kil	faon	zop	maer	zaop	viur	saen	pos	baic	loix	heuj	fan	meor	veez	kiel	
Layer 3	mar	lic	san	vir	teg	vez	lioc	huj	pois	cuow	teag	mor	lox	cow	kil	faon
	zop	maer	zaop	viur	saen	pos	baic	loix	heuj	fan	meor	veez	kiel	bac		

Table 37. Patterns acquired in the first three layers of the DDAM-1 model; the complete acquisition of the lexicon is attained in the third layer due to the increased ambiguity of input data

4.2.2.4 Lexical acquisition by DDAM-2

The new version of the model outperforms DDAM-1. Aside from the acquisition of additional significant regularities (Table 38), DDAM-2 indicates the existence of the 3-word pattern *virsanlioc* appearing twice in the input and which escaped our admittedly superficial "visual inspection" (Figure 55).

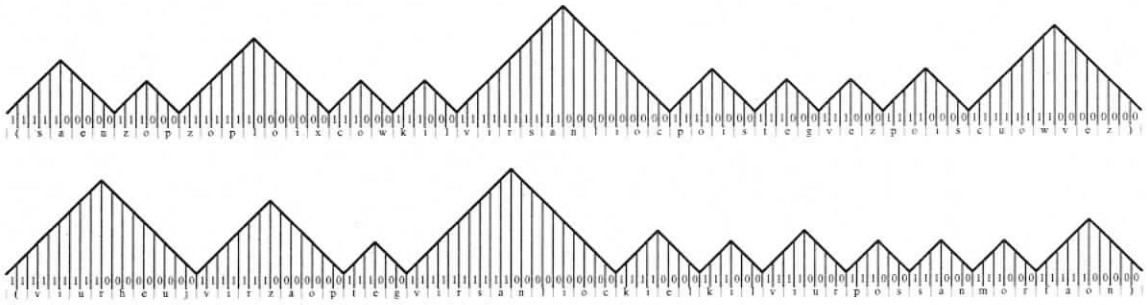


Figure 55. Dyck paths of the non-overlapping compositions (ambiguity thresholds $\underline{a} = 0$ bit, $\underline{a} = 0$ bit) of the two sequences containing the "elusive" 3-word pattern *virsanlioc*

lioc | vezpos | pos | cowlic | mar | lioc | maer | loixmar | saen | cuow | liockil
baic | posmar | baic | mar | markiel | poissaen | fan | zaop | cuowteag | mor | cow
vir | kilmor | teg | faon | liczop | kil | maercow | vir | cuowzaop | hujteg
meorsan | fan | baic | faon | teag | kiel | viur | kiel | cuowteag | baic | lic | poissaen
liczop | lox | vir | zaopsan | cuow | veez | huj | zaop | pos | loix | viur | cuowmar
veez | loix | licfan | teag | pos | maercow | bac | lox | veez | teg | zaopteg | meor
teg | cowvir | zop | faon | vez | vir | san | faonpos | heuj | teag | heujsaen | teag
faonsan | viur | mar | kielmaer | viur | lox | loix | zaopbaic | meor | saensaen | huj
maer | meorsaen | meorloix | teg | mar | meor | fan | cuow | faon | faon | kielkiel | huj
san | kil | licvir | marlic | marmar | viur | teag | faonsan | saen | lox | pos
saen | zop | zoploix | cow | kil | virsanlioc | pois | teg | vez | pois | cuowvez
baic | tegkiel | faonsan | fan | maer | vezloix | baic | pois | meor | cuowmar | baic
veez | kil | cuow | cuow | heuj | kilmor | zop | loixkiel | loix | heuj | santeg | baic
loix | san | maerlioc | bac | vezpos | lioc | teag | lic | cowcuow | baic | kiel | vez
vir | cow | vez | zaopbaic | marlic | posmar | mor | saen | virzop | maer | cuow
heuj | cow | mor | lox | liocpois | cow | viurbaic | vir | vez | cow | kiel | zaop | mar
kil | lioc | saen | viur | meorsan | loixcow | fan | saen | baic | cuowteag | meorheuj
tegteag | hujteg | heujsaen | faon | zaop | meor | pois | heuj | baicvir | pois | lioc
viurheuj | virzaop | teg | virsanlioc | kiel | kil | viur | pos | san | mor | faon
teg | santeg | huj | pois | mar | cuow | teg | tegteag | mor | bac | lic | loxcow
vez | faon | cuow | bac | hujcow | liczop | pos | zop | saen | mar | cuow | zaop | saen
viur | meor | loixkiel | fan | lic | maervez | baiczop | baic | san | pois | posmar
viur | cow | liockil | fan | kiel | cuowvez | lioc | viur | faon | mar | pos | baic | veez
viurheuj | huj | lic | lioc | fan | fan | mar | faonpos | mor | morzop | meor | maer
pos | viur | maer | poskiel | cow | poiscow | meorkil | zopzaop | zaop | kielmaer
viur | vir | saensaen | veez | pos | saen | vez | veez | lic | pos | kielmaer | lioc | huj
viur | baic | licfan | vezloix | bac | pois | zop | cuow | kil | bac | vir | meor | vir
vez | teag | teg | kiel | kielteag | loixmar | loixbaic | zop | lioc | vir | loxcow
baic | maer | sancuow | loix | lox | heuj | maer | mar | lox | cuow | cow | meorheuj | meor
pois | loix | kil | zaopsan | veez | meorkil | kielteag | vir | hujcow | saen | lic

Table 38. "Peaks only" representation of non-overlapped compositions (ambiguity thresholds $\underline{a} = 0$ bit, $\underline{a} = 0$ bit) of the strings in Table 36

DDAM-2 outperforms ADIOS and SEQUITUR in their ability to acquire multiword phrases. SEQUITUR however does better than ADIOS from this perspective.

Phrase	DDAM-2	ADIOS	SEQUITUR	Phrase	DDAM-2	ADIOS	SEQUITUR
baiclic			•(2)	marcuow			•(2)
baicmar	•(1)			markiel	•(1)		•(2)
baicvir	•(1)		•(2)	marlic	•(2)		•(2)
baiczop	•(1)			marmar	•(1)		
cowcuow	•(1)			meorheuj	•(2)		
cowvir	•(1)			meorkil	•(2)		
cowmeor	•(1)		•(2)	meorloix	•(1)		•(2)
cuowmar	•(2)		•(2)	meorsaen	•(1)		•(2)
cuowteag	•(3)	•	•(3)	meorsan	•(2)		•(2)
cuowvez	•(2)		•(2)	morzop	•(1)		
cuowzaop	•(1)			poiscow	•(1)		
faonpos	•(2)		•(2)	poissaen	•(2)		•(2)
faonsan	•(3)	•	•(3)	poskiel	•(1)		•(2)
heujsaen	•(2)		•(2)	posmar	•(3)	•	•(3)
hujcow	•(2)		•(2)	saensaen	•(2)		
hujteg	•(2)		•(2)	sancuow	•(1)		•(2)
kielkiel	•(1)			santeg	•(2)		•(2)
kielmaer	•(3)	•		tegkiel	•(1)		•(2)
kielteag	•(2)		•(2)	tegteag	•(2)		
kilmor	•(2)		•(2)	vezloix	•(2)		
licfan	•(2)			vezpos	•(2)		•(2)
licvir	•(1)			virsan		•	•(3)
liczop	•(3)	•	•(3)	virsanlioc	•(2)		
liockil	•(2)			virzaop	•(1)		•(2)
liocpois	•(1)		•(2)	virzop	•(1)		•(2)
loixbaic	•(2)		•(2)	viurbaic	•(1)		
loixcow	•(1)			viurheuj	•(2)		•(2)
loixkiel	•(2)			zaopbaic	•(2)		•(2)
loixmar	•(2)		•(2)	zaopsan	•(2)		
loxcow	•(2)		•(2)	zaopteg	•(1)		
maercow	•(2)		•(2)	zoploix	•(1)		•(2)
maerlioc	•(1)		•(2)	zopzaop	•(1)		
maervez	•(1)		•(2)				

Table 39. Cumulted list of multiword phrases in the DDAM-2 and SEQUITUR compositions showing dots and phrase counts as acquired by each model; SEQUITUR misses on eleven "important" chunks which would actually satisfy its "rule utility" constraint

The ADIOS model used in the MEX (motif extraction) mode with a relaxed p=1.0 value, on identical data but pre-processed appropriately, was able to form rules for the 3 and 4-letter patterns (Table 40) and picked up six additional higher-rank regularities. This again indicates that the model is either not supposed to attain such behaviour or a fundamental limitation of its algorithm is preventing it to discover higher-rank regularities. However, interestingly, the model was able to pick-up the higher rank patterns that have the highest counts which suggests that its heuristics include a sort of threshold for the pattern frequency.

ID	Pattern	Expansion
P26	(l,i,o,c)	lioc
P27	(b,a,i,c)	baic
P28	(f,a,o,n)	faon
P29	(t,e,a,g)	teag
P30	(z,a,o,p)	zaop
P31	(l,o,i,x)	loix
P32	(t,e,g)	teg
P33	(v,i,u,r)	viur
P34	(m,a,e,r)	maer
P35	(l,i,c)	lic
P36	(s,a,e,n)	saen
P37	(k,i,e,l)	kiel
P38	(k,i,l)	kil
P39	(c,u,o,w)	cuow
P40	(v,i,r)	vir
P41	(h,e,u,j)	heuj
P42	(m,e,o,r)	meor
P43	(p,o,i,s)	pois
P44	(p,o,s)	pos
P45	(h,u,j)	huj
P46	(c,o,w)	cow
P47	(z,o,p)	zop
P48	(m,a,r)	mar
P49	(m,o,r)	mor
P50	(e,z)	ez
P51	(a,n)	an
P52	(l,o,x)	lox
P53	(s,P51)	san
P54	(v,e,P50)	veez
P55	(v,P50)	vez
P56	(f,P51)	fan
P57	(P35,P47)	liczop
P58	(P39,P29)	cuowteag
P59	(b,a,c)	bac
P60	(P40,P53)	virsan
P61	(P28,P53)	faonsan
P62	(P44,P48)	posmar
P63	(P37,P34)	kielmaer

Table 40. ADIOS table of extracted patterns showing rules for only 3 and 4 letter patterns

It also seems that both DDAM-2 and ADIOS outperform SEQUITUR in the acquisition of a few simple, but ambiguous words. For example, in the first sequence (Table 41), SEQUITUR misses the appropriate acquisition and composition of the words *cow*, *lic*, *cuow* and *lioc* in the context of the patterns *cowlic* and *cuowlioc*.

lioc | vezpos | pos | c | owli | c | mar | lioc | maer | loixmar | saen | cu | owli | oc | kil |
baic | posmar | baic | mar | markiel | poissaen | fan | zaop | cuowteag | mor | cow |
vir | kilmor | teg | faon | liczop | kil | maercow | vir | cuow | zaop | hujteg
meorsan | fan | baic | faon | teag | kiel | viur | kiel | cuowteag | baiclic | poissaen |
liczop | lox | virzaop | sancuow | veez | huj | zaop | pos | loix | viur | cuowmar |
veez | loix | lic | fan | teag | pos | maercow | bac | lox | veez | teg | zaop | teg | meor |
teg | cow | virzop | faon | vez | virsan | faonpos | heuj | teag | heujsaen | teag |
faonsan | viur | markiel | maer | viur | lox | loix | zaopbaic | meorsaen | saen | huj
maer | meorsaen | meorloix | teg | mar | meor | fan | cuow | faon | faon | kiel | kiel | huj
san | kil | lic | vir | marlic | mar | mar | viur | teag | faonsan | saen | lox | pos |
saen | zop | zoploix | cow | kil | virsan | liocpois | teg | vez | pois | cuowvez |
baic | tegkiel | faonsan | fan | maervez | loixbaic | pois | meor | cuowmar | baic |
veez | kil | cuow | cuow | heuj | kilmor | zoploix | kiel | loix | heuj | santeg | baic |
loix | san | maerlioc | bac | vezpos | lioc | teag | lic | cow | cuow | baic | kiel | vez |
vir | cow | vez | zaopbaic | marlic | posmar | mor | saen | virzop | maer | cuow |
heuj | cow | mor | lox | liocpois | cow | viur | baicvir | vez | cow | kiel | zaop | mar |
kil | lioc | saen | viur | meorsan | loix | cow | fan | saen | baic | cuowteag | meor | heuj |
teg | teag | hujteg | heujsaen | faon | zaop | meor | pois | heuj | baicvir | pois | lioc |
viurheuj | virzaop | teg | virsan | lioc | kiel | kil | viur | pos | san | mor | faon |
teg | santeg | huj | pois | marcuow | teg | teg | teag | mor | bac | lic | loxcow |
vez | faon | cuow | bac | hujcow | liczop | pos | zop | saen | marcuow | zaop | saen |
viur | meorloix | kiel | fan | lic | maervez | baic | zop | baic | san | pois | posmar |
viur | cow | lioc | kil | fan | kiel | cuowvez | lioc | viur | faon | mar | pos | baic | veez |
viurheuj | huj | lic | lioc | fan | fan | mar | faonpos | mor | mor | zop | meor | maer
pos | viur | maer | poskiel | cow | pois | cowmeor | kil | zop | zaop | zaop | kiel | maer
viur | vir | saen | saen | veez | pos | saen | vez | veez | lic | poskiel | maerlioc | huj
viur | baiclic | fan | vez | loix | bac | pois | zop | cuow | kil | bac | vir | meor | vir |
vez | teag | tegkiel | kielteag | loixmar | loixbaic | zop | lioc | vir | loxcow |
baic | maer | sancuow | loix | lox | heuj | maer | mar | lox | cuow | cowmeor | heuj | meor |
pois | loix | kil | zaop | san | veez | meor | kil | kielteag | vir | hujcow | saen | lic

Table 41. SEQUITUR compositions of the strings in Table 36

4.2.2.5 Discussion

Though superior to ADIOS from the multiword acquisition perspective, SEQUITUR misses the acquisition of several significant chunks. The most likely reason for this behaviour is that the bottom-up chunking decisions of SEQUITUR, in the early stages of the learning procedure, lead to the commitment to certain chunks (Hopkins 1999) (e.g., *baiclic, virsan*) that hinder the subsequent acquisition of other ones (e.g., *licfan*, *kielmaer*, *liockil*, etc.) which would actually satisfy the algorithm's "rule utility" constraint. This appears to be a local minima problem, and is also likely to be improved to a certain extent by multiple passes through the input data. However multiple passes will increase the time complexity and remove the online learning capability of the SEQUITUR algorithm. Therefore, the acquisition of overlapping patterns (e.g., *licfan*, *liczop*) seems to hit a fundamental limitation of the SEQUITUR approach.

By contrast, instead of committing immediately to a non-overlapped composition, DDAM-2 first builds optimal, overlapping and redundant compositions of all the strings in the language in a bottom-up fashion. Due to its redundant compositional approach

based on the constrained substring poset model, DDAM-2 is able to account for all high-rank regularities, including those that overlap considerably. Subsequently DDAM-2 makes chunking decisions and commits to a particular non-overlapped representation from the optimal compositions, in a context-dependent, top-down manner. DDAM-2 is therefore more informed because it has access to more complete ambiguity information on which to make its chunking decisions. In addition, and probably most importantly, in this way DDAM-2 is also able to cope with the situation that the non-overlapping compositions of learned strings require to be dynamically reconsidered as a consequence of learning additional strings.

Finally it is fair to say that, for an online, one-pass algorithm with linear temporal and spatial complexity, SEQUITUR performs very well. The additional improvements in chunking and discovery of higher-rank regularities achieved by the DDAM-2 stem most likely from the added temporal and spatial complexity of the constrained substring poset model.

4.2.3 Experiment #3

4.2.3.1 Data sources

The third experiment has involved the creation of artificial input data using 1000 randomly selected words from a lexicon of only three words of different lengths: *ba*, *dii* and *guuu*. This has resulted in the 3013 characters sequence in Table 42.

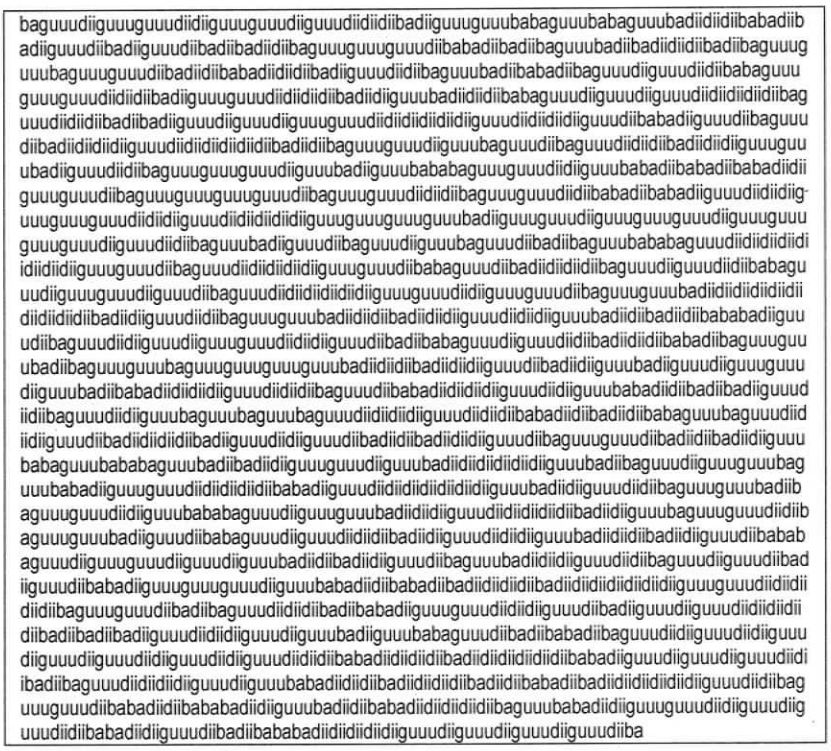


Table 42. 3013 character artificial input sequence for experiment #3

4.2.3.2 Criteria of success

The exact same set up was originally described in (Elman 1990) in order to test a model of simple recurrent network (SRN) particularly suitable for sequence processing, using a *supervised learning* paradigm. The task of the network was to predict the next letter in the sequence. The error signal was plotted and showed an increase at word boundary, i.e., before each of the consonants *b*, *d* and *g* (Elman 1990). Because in this particular experiment lexical items do not contain any morphological ambiguity, i.e., symbols which make individual words are specific to those words (e.g., *b* and *a* to *ba*, *g* and *u* to *guuu*, *d* and *i* to *dii*), the lexical acquisition problem was expected to be trivial. One would also expect high-rank regularities to be abundant and hence easy to acquire, a capability that, along with the deterministic nature of the models presented here, goes beyond the capabilities of the model described in (Elman 1990).

4.2.3.3 Lexical acquisition by DDAM-1

As expected, under normal constraints, the DDAM-1 acquires the lexical items *ba*, *dii* and *guuu*. However, in the same first layer, the model also discovers patterns such as *iba*,

idii and *diidi*. In subsequent layers, the model acquires 2-word "phrases" such as *baguuu*, *guuuba*, *baba*, *diiba*, *diiguuu*, *diidii* and *guuuguuu* as well as multi-word patterns such as *diidiidii*, *diidiiba*, *badiiguuu*, *guuudiidiidii*, *diidiiguuu* etc. As expected, constraint relaxation causes the DDAM-1 model to pick up "sentences" such as *diidiibaguuuguuu*, *badiibaguuuguuu*, *guuudiiguuudiidiiba*, and even *baguuudiiguuuguuudiiguuu*, a 7-word "sentence" that appears twice in the input. Therefore, in this experiment, the simple constrained substring poset model is able to not only discover the lexicon completely, but also to pick-up higher-rank regularities of the input data, i.e., phrases, propositions and sentences which comprise combinations of several phrases. In (Elman 1990) Elman refers to auto-associative models and briefly acknowledges their potential use for sequence processing. Aside from the more powerful *unsupervised learning* paradigm of the constrained substring poset model introduced in this dissertation, the results demonstrate superior sequence processing capabilities as well as provide a closer and clearer image of what exactly happens inside the model (i.e., how processing is achieved) through the deterministic, more elegant, poset formulation which goes beyond the non-deterministic, "black-box" type of model of recurrent artificial neural network used in (Elman 1990).

4.2.3.4 Lexical acquisition by DDAM-2

DDAM-2 shows comparable outcomes in this experiment. It also indicates that the 7-word sentence *baguuudiiguuuguuudiiguuu* exists actually in the form of the 8-word *baguuudiiguuuguuudiiguuudii* as well as the 9-word *babaguuudiiguuuguuudiiguuudii* in the non-overlapped composition in Table 43.

| (| baguuudiiguuuguuu | diidiiguuuguuudii | guuudiidiidiibadiiguuuguuu | babaguuuba | baguuubadii | diidii | babadiiba | diiguuudiibadiiguuudiiba | diibadiidiibaguuuguuu | guuudiibabadii | badiiba | guuubadiiba | diidiidiibadiiba | guuuguuu | baguuuguuudiibadiidiiba | badiidiidiibadii | guuudiidiibaguuubadii | babadiiba | guuudiiguuu | diidiiba | baguuuguuuguuudii | diidiibadii | guuuguuudiidiidii | badiidiiguuubadiidii | diibabaguuudiiguuudii | guuudiidiidiidiiba | guuudiidiidiibadiiba | diiguuudiiguuudiiguuu | guuudiidiidiidiidiiguuu | diidiidiidiiguuudii | babadiiguuudiiba | guuudiibadiidiidii | guuudiidiidiidiidii | badiidiiba | guuuguuudii | guuubaguuudiiba | guuudiidiidiibadii | diidiiguuuguuu | badiiguuudiidiiba | guuuguuuguuudii | guuubadiiguuubaba | baguuuguuudiidiiguuubaba | diibabadiibabadii | diiguuuguuudiibaguuuguuu | guuuguuudiibaguuuguuu | diidiidiibaguuuguuudii | diibabadiibabadii | guuudiidiidiiguuu | guuuguuudiidiidiiguuudii | diidiidiidiiguuuguuu | guuuguuubadiiguuu | guuudiiguuuguuu | guuudiiguuu | guuuguuuguuudiiguuu | diidiibaguuubadii | guuudiiba | guuudiiguuu | baguuudiibadiiba | guuubaba | baguuudiidiidiidiidii | diidiiguuuguuudiiba | guuudiidiidiidiidii | guuuguuudiiba | baguuudiibadiidii | diidiiba | guuudiiguuudiidiiba | baguuudiidiidiidiidii | baguuudiidiidiidiidii | guuuguuudiidiiguuu | guuudiibaguuuguuu | badiidiidiidiidii | diidiidi | diibadiidiiguuudiidii | baguuuguuubadiidiidi | badiidiidiiguuudiidii | baba | badiiguuudiiba | guuudiidiiguuudiiguuu | guuudiidiidiiguuudiibadii | babaguuudiiguuudii | diibadiidiidi | babadiiba | guuuguuubadiiba | guuuguuu | baguuuguuuguuuguuu | badiidiidiibadiidiidii | guuudiibadiidi | guuubadiiguuudii | guuuguuudiiguuubadii | babadiidiidiidi | guuudiidiidiiba | guuudiibabadii | diidiidiiguuudiidi | guuudiibabadii | badiidiibadiidii | diibaguuudii | diiguuu | baguuu | baguuu | baguuudiidiidiidiiguuudii | diidiibabadiidii | badiidiiba | baguuubaguuudiidiidi | guuudiibadiidi | diidiibadii | guuudiidiiguuudii | badiidiibadiidi | diiguuudiiba | guuuguuudiibadiidiiba | diidiiguuuba | baguuubaba | baguuubadiibadiidi | guuuguuudiiguuubadii | diidiidiidiidiiguuubadii | baguuudiiguuuguuu | baguuubabadii | guuuguuudiidiidiidiiba | badiiguuudiidi | diidiidiidi | diiguuubadiidii | guuudiidiibaguuuguuubadii | baguuuguuudiidiiguuubaba | baguuudiiguuuguuuba | diidiidiiguuudiidiidiidii | badiidiiguuu | baguuuguuudiidiiba | guuuguuubadiiguuudii | babaguuudiiguuudii | diidiibadii | diiguuudiidiidiiguuubadiidi | diibadiidiiguuudiiba | babaguuudiiguuuguuudiiguuudii | guuubadiidiibadiidi | guuudiiba | guuubadiidiidi | guuudiidiiba | guuudiiguuudii | badiiguuudiibaba | diiguuuguuuguuudiiguuu | babadiidi | babadiibadiidi | diidiibadiidi | diidiidiidiidiiguuuguuudii | diidiidiidiiba | guuuguuudiiba | diiba | guuudiidiidiidiibadiiba | badiiguuuguuu | diidiidiiguuudiibadii | guuudiiguuudiidiidiidiiba | diibadiibadiiguuudiidii | diiguuudiiguuubadii | guuuba | baguuudiibadiiba | badiiba | guuuudiidiiguuudii | diiguuudiiguuudiiguuudiidi | guuudiidiiguuudii | diidiibabadiidi | diidiibadiidiidiidiidi | babadiiguuudii | guuudiiguuudiidiiba | diiba | guuudiidiidiidiiguuudii | guuubabadiidi | diibadiidiidiibadiidi | babadiibadiidiidiidi | diidiiguuudiidiibaguuuguuu | diibabadiidi | baba | badiidiiguuu | badiidiibabadiidi | diidii | diibaguuubaba | diidiiguuuguuudiidiiguuu | diiguuudiidiiba | badiidii | guuudiibadiibaba | badiidiidiidiidi | guuudiiguuudiiguuudii | guuudiiba |) |

Table 43. "Peaks and valleys" representation of a non-overlapped composition of the sequence in Table 42; the two longest high-rank regularities are underlined

4.2.4 Context dependent grammar induction

Though categorized as lexical acquisition or chunking, previous experiments can also be regarded as a form of grammar induction that works at a word and phrase level. In effect one could easily write all chunking decisions of the algorithms as context free, formal grammars consisting of sets of rewrite rules. In the following experiment ADIOS, SEQUITUR or DDAM-2 will be used as grammar induction algorithms and tested against a common data set.

4.2.4.1 Data sources

The ADIOS demo evaluation kit includes among other things, a small, artificially generated corpus whose 500 sentences generated from a context free grammar very equivalent to that in Table 9.

Grammar rule	Example
NP1 → the {cat, dog, cow, bird, rabbit, horse}	the cat
NP2 → {Joe, Beth, Jim, Cindy, Pam, George}	Beth
NP → {NP1, NP2}	
VP1 → {believes, thinks}	believes
VP2 → {please, read}	read
P → NP2 VP1 that	Jim believes that
S1 → {that} NP is {easy, tough, eager} to VP2 {annoys, worries, disturbs, bothers} NP	that the cat is eager to please bothers the dog.
S2 → {Beth, Pam, Cindy} {believes, thinks} that P to VP2 is {easy, tough}	Beth thinks that Jim believes that to read is tough.
S → {S1,S2}	

Table 44. The rewrite rules of the context free grammar of the 500 sentence corpus included in the ADIOS demo evaluation package

Apart from the occasional semantic quirks which cause some sentences to logically contradict each other or to represent rather silly scenarios, this simple data source could be used in order to compare some of the grammar induction capabilities of ADIOS, SEQUITUR and DDAM-2.

#	Sentence	Symbol-encoded
1	Cindy thinks that George thinks that to read is tough	ABCDBCEFGH
2	that the bird is eager to read bothers the dog	CIJGKEFLIM
3	Pam thinks that Jim thinks that to read is tough	NBCOBCEFGH
4	Pam believes that Cindy thinks that to please is tough	NPCABCEQGH
5	that the cat is easy to read disturbs George	CIRGSEFTD
6	Cindy believes that George thinks that to read is easy	APCDBCEFGS
7	Pam believes that Joe thinks that to please is tough	NPCVBCEQGH
8	Cindy believes that Joe believes that to please is easy	APCVPCEQGS
9	that the cat is tough to please worries the bird	CIRGHEQUIJ
10	Cindy thinks that Joe thinks that to please is easy	ABCVBCEQGS

Table 45. The first 10 sentences taken out of the generated, 500 sentence corpus included in the ADIOS demo evaluation kit together with their symbol-encoding

The most important limitation of data sets such as the one in Table 45 is that representation space of the sentences is artificially compacted. The sentences exhibit a too great degree of structure which departs significantly from the properties of real natural language, whose syntax and semantics are much richer and cause representation spaces to be much more sparse.

As a matter of technicality, when applied to input data such in Table 45, SEQUITUR and DDAM-2 will attempt to represent the input at character level, and create grammar rules that involve sub-tokens as well as the blank character, unlike ADIOS which works at a token level. Therefore, in order to bring the analysis to a better common denominator, the

data set has been symbol-encoded by replacing each lexical item with a unique symbol according to the rewrite rules in Table 46.

Cindy → A	thinks → B	that → C	George → D	to → E	read → F	is → G
tough → H	the → I	bird → J	eager → K	bothers → L	dog → M	Pam → N
Jim → O	believes → P	please → Q	cat → R	easy → S	disturbs → T	worries → U
Joe → V	annoys → W	cow → X	Beth → Y	horse → Z	rabbit → @	

Table 46. Grammar rules used to symbol-encode the generated data source in order to bring the analysis to a common denominator

4.2.4.2 Criteria for evaluation

Because grammars can be formally specified in various ways and there are exponentially many grammars for a given language, there is a lack of universal criteria to compare various descriptions and decide which one is better. Hence grammar induction evaluation is not a straightforward matter. The alternative to compare grammars from a MDL perspective, as argued in the background chapter, seems inappropriate since a rich, potentially redundant grammar able to capture various, algorithmically significant (i.e., long) regularities in a dataset could be satisfactory and useful but not necessarily have the smallest description length. Comparing results based solely on the length of the acquired patterns, though significant in the context of algorithmic information theory, also does not do justice to the ADIOS model which seems to be unable to acquire long regularities. Therefore, only a limited aspect of the algorithms will be compared, namely the capacity to acquire equivalence sets, in particular the equivalence sets of terminal symbols.

Also a matter of technicality, unlike in the case of ADIOS where equivalence sets are explicit in the output, for DDAM-2 and SEQUITUR, retrieving an equivalence set from their induced grammars involves an additional step. The additional step consists of sorting the rewrite rules by their prefix/suffix content, followed by the selection of the rules that contain at least a terminal symbol. For example, given the sorted rules in Table 47, the sets {J, R, X, M, Z, @}, {L, T, U, W} and {H, K, S} are all terminal symbol equivalence sets whose elements possess identical prefix or suffix contents.

Rule head	Prefix context #1	Suffix context #2	Symbol-encoded expansion	Decoded expansion
11 →	12	J	(CI|J	(that the bird
33 →	12	R	(CI|R	(that the cat
74 →	12	X	(CI|X	(that the cow
95 →	12	M	(CI|M	(that the dog
161 →	12	Z	(CI|Z	(that the horse
170 →	12	@	(CI|@	(that the rabbit
564 →	L	135	L|IM)	bothers the dog)
220 →	T	135	T|IM)	disturbs the dog)
581 →	U	135	U|IM)	worries the dog)
566 →	W	135	W|IM)	annoys the dog)
9 →	G	H	G|H	is tough
17 →	G	K	G|K	Is eager
36 →	G	S	G|S	Is easy

Table 47. Example of sorted rewrite rules in the DDAM-2 induced grammar; because the prefix content of rules 11, 33, 74, 95, 161, 170 and rules 9,17,36 and the suffix content of rules 564, 220, 581, 566 are identical, {J, R, X, M, Z, @}, {L, T, U, W} and {H, K, S} are equivalence sets of terminal symbols

4.2.4.3 Terminal symbol equivalence set acquisition

All three models studied here seem to be able to satisfy, to various extents, the alternative definition of syntactic systematicity introduced in Chapter 2 and which consists of capability of a language processing model to assign a lexical item into the appropriate equivalence set, in an unsupervised manner. The acquisition results are similar and all models are fairly good at acquiring the equivalence sets of the six proper nouns (i.e., Beth, Cindy, George, Jim, Joe, Pam), of the six animal nouns (i.e., rabbit, bird, dog, cat, cow, horse) as well as the sets of verbs {believes, thinks}, {please, read} and of adverbs {eager, easy, tough}. The only exception seems to be the failure of ADIOS to account for the set of verbs {bothers, worries, annoys, disturbs} (Table 48) but which was captured by both SEQUITUR (though only partially) (Table 49) and by DDAM-2 (Table 50).

Cindy thinks that {Beth,Cindy,George,Jim,Joe,Pam}
{believes,thinks} that
the {bird,cat,cow,dog,horse,rabbit}
{Beth,Cindy,Pam}{believes,thinks} that {Beth,Cindy,George,Jim,Joe,Pam}{believes,thinks} that to
{Beth,Cindy,Pam}{believes,thinks} that {Beth,Cindy,George,Jim,Joe,Pam}{believes,thinks} that to {please,read} is
that the {bird,cat,cow,dog,horse,rabbit} is {eager,easy,tough} to {please,read}
Cindy thinks that {Beth,Cindy,George,Jim,Joe,Pam} {believes,thinks} that to {please,read} is
that {Beth,Cindy,George,Jim,Joe,Pam} is {eager,easy,tough} to {please,read}

Table 48. The results of acquisition of terminal symbol equivalence set by the ADIOS model

that the {cat, rabbit}
(that {Cindy, the, Pam, George, Jim, Joe}
{Cindy, George, Pam, Jim, Joe, Beth} believes that
{Cindy, George, Pam, Jim, Joe, Beth} thinks that
(that the {cat, horse, dog, bird, cow}
is easy to read { worries, bothers, annoys}
is eager to read { disturbs, annoys}
is easy to read bothers {Beth, the}
is tough to please {annoys, disturbs}
is easy to please {disturbs, worries, bothers}
{rabbit, George, dog, Joe, cow} is easy to read
{Joe, cow} is tough
{bird, cow} is eager
{thinks, believes} that
{disturbs, bothers, worries} Joe
to {read, please}
is {tough, eager, easy}
the {rabbit, bird, dog, cat, cow, horse}
bothers {Cindy, George, the, Pam, Joe}

Table 49. The results of acquisition of terminal symbol equivalence set by the SEQUITUR model

(that {the, George, Cindy, Joe, Pam, Jim, Beth}
(that the {bird, cat, cow, dog, horse, rabbit}
(that the cow is tough to read {disturbs, worries}
is eager to read {bothers, disturbs, annoys}
is easy to read {bothers, worries, annoys, disturbs}
is tough to read {bothers, worries, annoys, disturbs}
is tough to please {bothers, annoys, disturbs}
is eager to please {bothers, worries, annoys, disturbs}
is easy to please {bothers, worries, annoys, disturbs}
{Cindy, Pam, Beth} thinks that
{Cindy, Pam, Beth} believes that
{Cindy, George, Pam, Jim, Joe, Beth} thinks that to read is tough)
{Cindy, George, Pam, Jim, Joe, Beth} believes that to read is tough)
{Cindy, George, Pam, Jim, Joe, Beth} thinks that to please is tough)
{Cindy, George, Pam, Jim, Joe, Beth} thinks that to please is easy)
{Cindy, George, Pam, Jim, Joe, Beth} believes that to please is easy)
{Cindy, George, Pam, Jim, Joe, Beth} believes that to please is tough)
{Cindy, George, Jim, Joe, Beth} thinks that to read is easy)
{Cindy, George, Jim, Joe} believes that to read is easy)
{bothers, disturbs, worries, annoys} the dog)
{bothers, disturbs, worries, annoys} the cow)
{bothers, disturbs, worries} the cat)
{bothers, worries} the horse)
to {read, please}
is {though, eager, easy}

Table 50. The results of acquisition of terminal symbol equivalence set by the DDAM-2 model

In addition, certain technical aspects with regard to the runtime behaviour of the ADIOS and DDAM-2 models have also been studied. For this, an additional data set of about 2000 sentences was generated in a similar way but from a more complex context free grammar whose specification was also included in the ADIOS demo package. The experiment carried out on this larger data set has revealed that the run-time of the ADIOS was in the order of minutes (precisely 7 min 27 sec), under the CYGWIN environment. This run-time was also considerably shorter than the entire run-time, since the analysis

stopped automatically at about 25% of the task due to an imposed limitation of the demo version that prevented that number of discovered patterns to extend beyond 100. The run time of DDAM-2, on the same hardware and on the exact same data set was under 5 seconds, i.e., about 90 times faster than ADIOS. The drastic run time difference provides clear empirical evidence about two (possibly confounding) efficiency aspects of the two models: the algorithmic efficiency and the implementation efficiency. The potential differences in the latter aspect are difficult to assess, but those of the former aspect stem most likely from the spatio-temporal complexity trade-off of the DDAM-2 that is able to achieve a reduced time complexity (i.e., shorter runtime) at the expense of space complexity (i.e., memory requirements).

4.2.4.4 Discussion

Qualitatively, differences between the three machine generated grammars are also evident. ADIOS seems to be oriented towards creating compact grammars, while SEQUITUR and DDAM-2 attempt to capture as much regularity in the dataset as possible. As a result, and due to the relative compactness of this particular dataset, SEQUITUR and DDAM-2 end up with rich grammars comprising hundreds of rewrite rules. Qualitatively, it also appears that SEQUITUR results uncover yet again some local minima problems that prevent the model to achieve a better global coherence unlike those captured by DDAM-2.

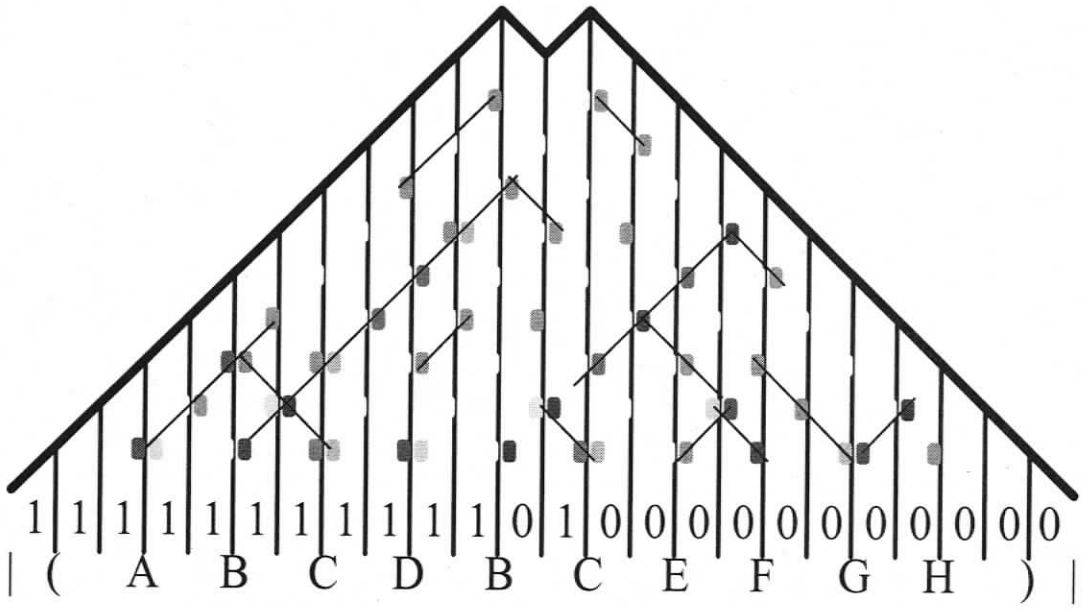


Figure 56. Optimal, overlapped representation of the first sentence in Table 45 which does not commit to a certain grammar but subsumes various alternative grammars, at the same time

This leads to probably the most important observation that the optimal, overlapped representations that DDAM-2 creates for each sentence in the data set are not committed to any particular grammar and hence can be thought of as being equivalent to multiple grammars, at the same time. For example, since it is almost a maximum rank composition that represents virtually all sub-sequences of ABCDEFGH, the representation in Figure 56 subsumes nearly all $2^{11} = 2048$ possible segmentations (or grammars) of the sentence ABCDEFGH. Though highly redundant when compared to the trivial composition of ABCDEFGH, this representation is very necessary in order to deal with the ambiguities of datasets, particularly compact, artificial ones. However, even in the case of natural language processing, redundant representations could be desirable and may resemble human representations, which are also known to be redundant. The fact that natural language often contains redundant specifications and syntax which do not seem to be always committed to a unique, parsimonious grammar that obeys minimum description length principles, is an additional argument that speaks for the cognitive validity of redundant models in general and of the DDAM in particular.

Therefore, from the point of view of language processing, be it natural or artificial, representing multiple grammars in a memory at the same time, seems to be an appropriate choice, providing one's memory is large enough to hold such redundant representations. Committing to a certain grammar would only imply an additional dynamic process of quick and efficient retrieval of the most appropriate grammar, when needed. Since memory seems to be plenty in the human brains and limited only by current technology in artificial processing agents, it looks like a redundant mechanism such as that of DDAM-2 that trades high space complexity in order to reduce time complexity could become both a useful and a usable approach, at the same time.

An additional related aspect that has a bearing on the *language generation* capabilities of the DDAM-2 model must also be recognized: the optimal, overlapped representations that DDAM-2 achieves fit the input data perfectly and allow no generalization and generation of new sentences. For example, in Figure 57, the representation of the first 10 sentences in Table 45 contains exactly the 10 paths corresponding precisely to the 10 compositions.

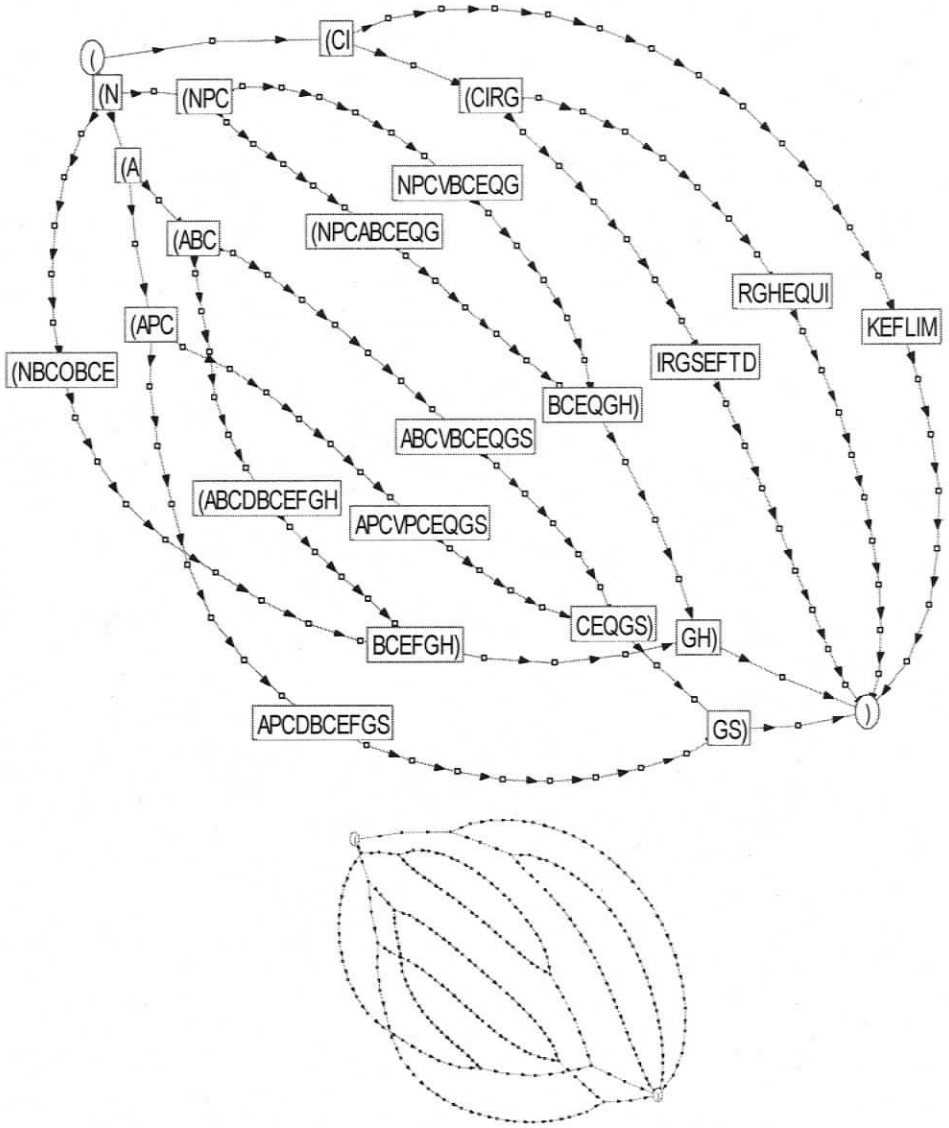


Figure 57. Optimal compositions of the 10 sentences in Table 45 (ADIOS demo dataset); all 10 possible paths are unique, do not contain loops and correspond to existing strings in the language

Even though some paths share certain sub-sequences, each sequence is unique and starts at the open round bracket (alpha, the start symbol) and ends at the closed round bracket (omega, the end symbol), unambiguously. Therefore, due to the lack of ambiguity, no deviation, invention or generalization to new strings is achievable with such compositions. In order to be able to do this, one would have to gradually increase the representation ambiguity. Only in this way can one create generalizations, which are natural, interesting, and possess the appropriate ratio between regularity and novelty.

In the context of the data set used in this particular experiment, increasing the ambiguity of representations has led to the structures in Figure 58 and Figure 59. From a constrained substring poset point of view, the ranks (and hence the redundancy) of the compositions have decreased, but at the same time, their ambiguity has increased. By instilling ambiguity in representations one has gained the capabilities of generalization and "inventiveness."

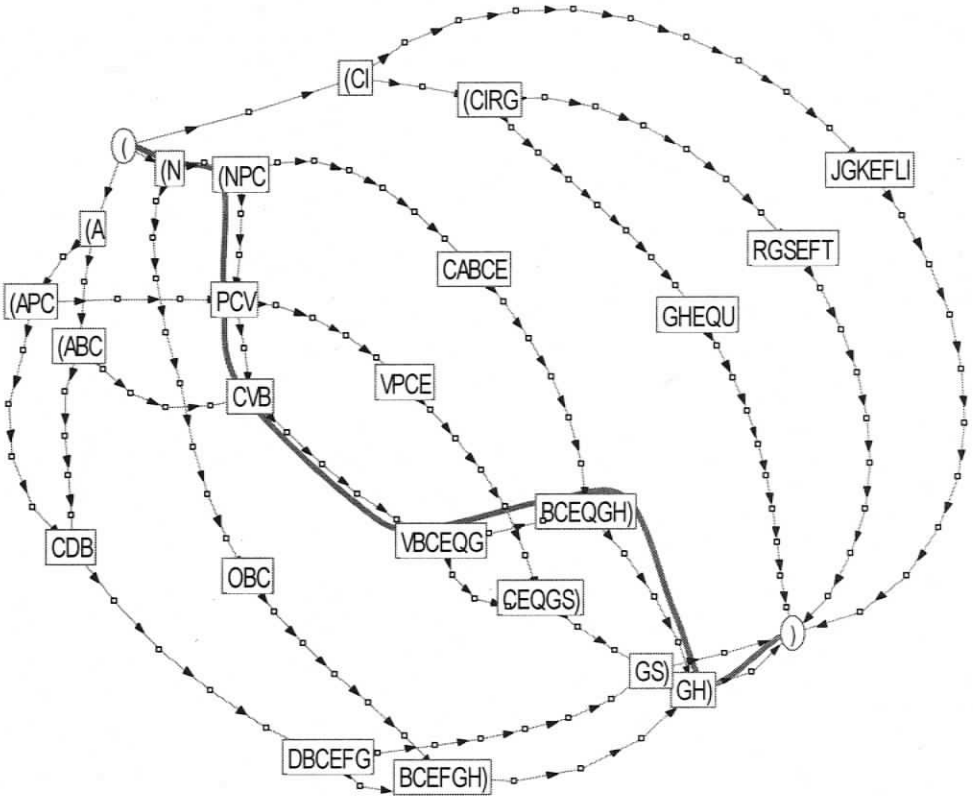


Figure 58. Ambiguous compositions (ambiguity thresholds equal to 3) of the 10 sentences in Table 45 (ADIOS demo dataset); there are more than 10 possible paths some of them not part of the data set (e.g., NPCVBCEQGH shown in green) but none contain loops

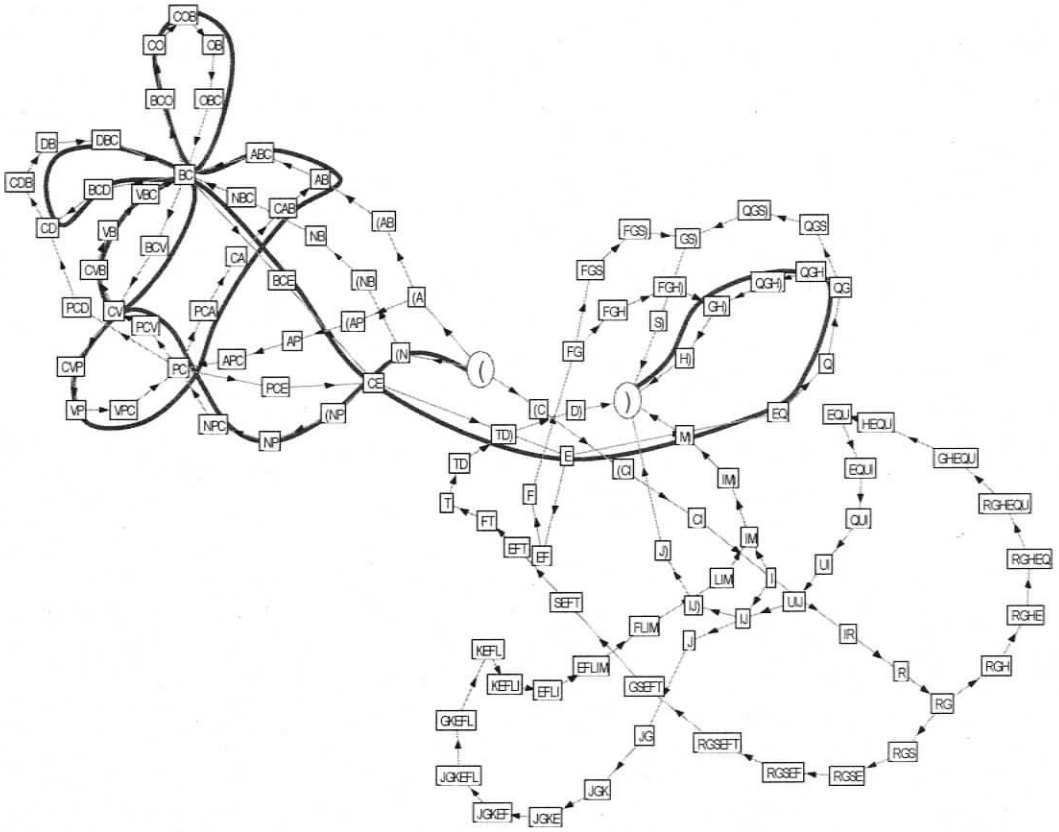


Figure 59. Ambiguous compositions (ambiguity thresholds equal to 7) of the 10 sentences in Table 45; there are infinitely many additional possible paths such as the one that shown in blue and that may contain loops

For example, "*Pam believes that Joe thinks that to please is tough*" (or NPCVBCEQGH in the symbol encoding) is a sentence that has a representation in Figure 58, but since it does not exist in the corpus is therefore a generalization. In addition, unlike the representation in Figure 58 in which the number of generalizations is limited, the structure in Figure 59 can account for infinitely many possible sentence generalizations due to the existence of loops. Tough admittedly of limited practical use for natural language processing, by allowing a variable numbers of loops in compositions, one could generate such sentences as "*Pam believes that [Joe believes that]* Cindy thinks that [Jim thinks that]* [George thinks that]* [Joe thinks that]* to please is tough*" (or NPC [VPC]* ABC [OBC]* [DBC]* [VBC]* EQGH) where the chunks in squared brackets may repeat an indefinite number of times (i.e., the star symbol stands for the iteration through a loop).

4.3 EXPERIMENTS WITH NATURAL SEQUENCES

This section comprises experiments that use the exact same processing principles, but on sequences that are natural (e.g., DNA sequences, text, etc.) rather than artificially constructed sequences as in previous experiments.

The major difference between experiments with artificial and natural sequences is that, in the case of the latter the results are not known in advance. This impacts directly the evaluation criteria for experiments in this section which will lack the precision of previous experiments on artificial sequences. In addition, given the additional complexity of natural sequences, the processing results of all models are expected to be inferior to those obtained for artificial sequences.

4.3.1 Genomic sequence processing

4.3.1.1 Data sources

DNA sequences of the SARS virus (e.g., TACCCAGGAAAAGCCAACCAACCTCGATCTCTTGTAGA…) (Balotta, Corvasce et al. 2004) as well as protein sequences derived from the SARS genome represented as sequences of aminoacids using a single letter encoding (e.g., MESLVLGVNEKTHVQLSLPVLQ…) were the data sources for the following experiment.

4.3.1.2 Criteria for evaluation

In the case of DNA sequences one would expect the models to be able to acquire the "words of genetics" (i.e., codons) corresponding to the Universal Genetic Code shared by all living organisms, as well as higher-rank regularities of such words. The evaluation consists of a quantitative assessment of the discovered patterns as valid codons or codon sequences. Because a codon is always a 3-nucleotide sequence, a simple criterion is that patterns are valid if their length is a multiple of 3. Comparing the results with those of

specialized programs for DNA and protein sequence alignment and motif detection, while an interesting and very relevant endeavour that may be pursued in the future, remains beyond the scope of the experiments in this dissertation.

4.3.1.3 Codon detection by DDAM-1

Normal constraints have caused the DDAM-1 model to acquire certain patterns in its first layer and to exhibit relatively slight differences of acquisition in higher layers. The knowledge that codons are always sequences of 3-nucleotides and the acquisition of some 4-letter patterns indicated that pattern acquisition constraints needed to be tightened. This intervention rendered the process of pattern discovery somewhat supervised and yielded the results in Table 51 which show acquisition of some codons but also the persistence of some four nucleotide patterns.

AGA	CTT	CAAA	TGTT	CTTT	ATG	CAG	TTG	CATT	CTAT	CTA	CAT	TGA	ACT	AAG	TTA
Arg	Leu				Met	Gln	Leu			Leu	His	Stop	Thr	Lys	Leu

Table 51. The most frequent, larger patterns acquired by the DDAM-1 model and their corresponding coded aminoacid

4.3.1.4 Codon detection by DDAM-2

The non-overlapped compositions derived from DDAM-2 optimal compositions show many of the codons and codons sequences but patterns whose lengths are not multiple of 3 can still be found (Table 52).

t	a	g	c	tt	aa	ca	ct	tg	ac
ta	gt	ag	at	tc	gc	ga	cc	tgt	ttg
atg	aca	cg	gg	ttt	tac	ctt	tca	tga	aaa
att	aag	aga	tta	tgc	ctg	caa	cta	gtg	gtt
tat	tgg	aat	act	agg	agt	gta	tct	tag	caaa
ttc	cat	gag	taa	aac	cac	tttctt	gac	gct	acc
gca	tagaa	agc	gtc	tttcaa	cca	cct	ctta	ggt	atc
caccaa	cag	ctc	gctca	tgcaa	ttattg	acgc	ata	atta	cacca
caga	cgt	ggaag	tagag	tcc	tgaga	tgata	ttct	ttga	tggcc
ggttg	ctact	aagca	aagcc	acac	agaaat	agcatt	agcttt	ataa	cagg

Table 52. The first 100, most frequent patterns in the non-overlapped compositions the DDAM-2 model; the 57 greyed cells correspond to codons or valid codon sequences

The outcome seems superior to that of the ADIOS model, in the MEX mode, on the same data (Table 53) even though admittedly, due to the lack of information about the ADIOS

model and its parameters, its heuristics could not be tuned easily in a way that maximizes the number of patterns whose lengths are multiples of 3.

tt	aa	ag	at	ct	gg	gt	ac	tgc	att
cac	ctt	caa	cc	gtt	gaa	agt	act	tgct	aat
cag	gag	cat	ggt	atg	ttt	gat	cg	acaa	catt
cct	cact	cttt	cgt	acac	gac	actt	ctac	atgc	aaaa
attt	gttt	caat	atgg	atgt	ctgt	acag	ctaa	gatt	ataa
gtgg	aggt	ctgc	ccag	ggag	aagc	aagg	agg	cctt	ccaa
cagt	agaa	tgctt	aaag	atgct	ctgg	ggtt	aag	ctgct	cactt
atgtt	gtgt	ctat	acct	gtg	acat	gcac	aatgt	cagg	agtt
caatt	cgg	gagt	ttgg	acact	gact	atgag	actac	gatg	cgag
acttt	catgg	gtttt	gatgc	gatgtt	gaggtt	ctttcg	gttgaat	ggtgat	aaat

Table 53. The first 100, most frequent patterns discovered by the ADIOS model in the MEX configuration; the 28 greyed cells correspond to codons or valid codon sequences

Comparing the two sets of results shows that 20 out of the 34 patterns discovered by both DDAM-2 and ADIOS, do indeed have the length which is a multiple of 3 (Table 54).

aa	aag	aat	ac	acac	act	ag	agg	agt	at
ataa	atg	att	caa	cac	cag	cagg	cat	cc	cct
cg	cgt	ct	ctt	gac	gag	gg	ggt	gt	gtg
gtt	tgc	tt	ttt						

Table 54. Patterns acquired by both the ADIOS and the DDAM-2 models; the 20 greyed cells correspond to codons or valid codon sequences

4.3.1.5 Protein pattern detection

Unexpectedly, the DDAM-1 model is hardly able to pick 2-letter patterns (e.g., LL, LS, AL) from the protein sequence. On the other hand DDAM-2 pattern detection is improved and the model is able to pick up some longer patterns showing results comparable to those of the ADIOS model. However, the fact that all models demonstrate that regularities are still scarce in protein sequences is a sign that the randomness present in this particular input data is greater than in the artificial data used in previous experiments and than in DNA sequences. This randomness would translate into the primary structure of SARS proteins. The potential implications of this insight for proteomics, if any, are beyond the scope of this dissertation. However this gives the opportunity to acknowledge some limitations of the models and, in general, of the representation of 1-dimensional sequences (primary structure) generated from objects (e.g., proteins) which are known to also possess additional (secondary, tertiary and quaternary) structural properties which determine their 3-dimensional configuration. In

other words, non-contiguous regularities built from distal pieces in a sequence and which are determined by particular dispositions of a sequence in higher dimensional space (i.e., a 2 or 3 dimensional space) are not going to be acquired by this model.

vp**ml**lvppsfmtbvm
ioooi**iooaaa**eaio
rsrxcrssnnrgcrr
vhflsllbmmkpkvh
euaiaoo**aa**oioiiu
zjncnxxc**rr**lslrj
lhbfzzz**ml**stzcsv
ou**aa**oo**oooa**eooai
xjcnppprxngpwnr
mlsltfkmmflllcz
ooaoeaiaoao**ioo**o
rxnxgnl**rr**nxcxwp
hzshslvfmcvlp**ml**
uoauaie**aa**o**ioo**ao
jpnjncznrwrxsrx
smflmshzvtlk**ml**v
aaaioauoeeoiaii
nrncrnjpzgxlrcr
tpztpccmbvmplbt
eooe**oooaa**e**oooa**e
gspgswwrczrsxcg
vmvzzthlsp**ml**lmt
io**ioo**euoaooiiae
rrrppgjxnsrccrg
zcpmstsvbhm**mlml**
oooaaeaiauoaoai
pwsrngnrcj**rr**xrc
sfkkm**ml**hstcmvpv
aai**ioo**iuaeooioi
nnll**rr**cjngw**rr**sr
cvpbmphvfkfttbs
oioaoo**ue**aiaee**aa**
wrscrsjznlnggcn
llfshlzsclhlmvh
ii**aa**uioaoouoaiu
ccnnjcpnwxjx**rr**j

Table 55. Artificial sequences from experiment #1 considered a transposed collection of 36 column sequences of 15 symbols rather than a collection of 15 row sequences of 36 symbols

In order to illustrate this better, an artificial sequence example is necessary. For example, the 2-dimensional configuration of the input sequences used in experiment #1 exhibits certain regularities (*ml* - 9 times, *rr* and *aa* – 8 times, *ioo* – 6 times, *oooa* – 4 times) when considered a transposed collection of 36 column sequences of 15 symbols (Table 55) rather than a collection of 15 row sequences of 36 symbols (Table 29). Such patterns will never be picked up by the associative memory models from the sequences in Table 29 except if they are transposed as in Table 55.

4.3.2 Automated lexical acquisition from text

Automated lexical acquisition is an important Natural Language Processing task which matches very well the capabilities of the sequence processing model developed in this dissertation. The following experiments use text from various sources as input. In order to maintain a valid evaluations methodology which involves a close as possible common denominator for comparison with published results, the following experiments have been explicitly designed around general text sources as opposed to text sources taken from a more specific professional medical discourse.

4.3.2.1 Data sources

WordNet (Miller 1995) is an electronic lexical database built in agreement with psycholinguistic principles which suggest that human lexical memory is organized in a hierarchical fashion, based on semantic relationships. Because natural language abounds in polysemous (i.e. multiple meaning) words, in WordNet, the 146,000 lexical entities (or word forms) are associated with their possible meanings through a many-to-many relation accounting for over 195,000 word-sense pairs. The meanings themselves are stored in 110,000 sets of synonyms (or synsets), each synset containing an entry with the role of explaining the synset meaning through a definition and usage examples. For example, in WordNet, the word form *drug* is associated with three different meanings as a noun and verb, all listed in Table 1.

Part of speech	Word forms	Definition	Example
Noun	drug	a substance that is used as a medicine or narcotic	
Verb	drug, dose	administer a drug to	"They drugged the kidnapped tourist"
	drug, do drugs	use recreational drugs	

Table 56. Senses of the word "drug" in WordNet

WordNet is also organized by semantic relations which are represented as pointers between the synsets and this makes it more of a knowledge base than a simple lexicon. Following the example, by searching through WordNet for the semantic relations of the meaning of the noun *drug* one can easily find that it is a *causal agent* which is a part of the pharmacopoeia which is a "collection or stock of drugs" and that *to drug, to dose* is one way *to medicate* and *to treat*. Although such semantic relations in Wordnet are very

structured and precise, the definitions and usage examples of the meanings are largely unstructured text that requires automated approaches in order to derive some of its structure. This text source is represented by some 80,000 noun synsets (i.e., sets of synonyms) and their definitions and usage examples (i.e., glosses) which equates to some 6 megabytes of textual data.

Another source of textual data is represented by general English literature. Lewis Carroll's "ALICE'S ADVENTURES IN WONDERLAND" (Carroll 2005) (AIW for short) has been chosen in order to bring our analyses to a common denominator which is closer to what has been reported in literature and in order to alleviate the potential problems created by the analysis of texts which are too technical in nature and/or exhibit a great deal of structure.

One of the important differences between natural language text and the sequences used in previous experiments is that in texts, separators (e.g., blanks, commas, periods, brackets, etc.) are normally used in order to separate lexical items. In order to eliminate the importance of separators in a lexical acquisition process, the text source has been unsegmented by removing all separators. For the purpose of the following experiments, the WordNet glosses and the Lewis Carroll text have been preprocessed into sequences such as in Table 57.

neuralnetworkneuralnetanynetworkofneuronsornucleithatfunctiontogethertoperformsomefunctioninthebody
domaindemesnelandterritoryoverwhichruleorcontrolisexercisedhisdomainextendedintoeuropehemadeitthelawoftheland
downtherabbithole
alicewasbeginningtogetverytiredofsittingbyhersister
onthebankandofhavingnothingtodoonceortwiceshehad
peepedintothebookhersisterwasreadingbutithadno
picturesorconversationsinitandwhatistheuseofabook
thoughtalicewithoutpicturesorconversation
soshewasconsideringinherownmindaswellasshecould
forthehotdaymadeherfeelverysleepyandstupidwhether
thepleasureofmakingadaisychainwouldbeworththetrouble
ofgettingupandpickingthedaisieswhensuddenlyawhite
rabbitwithpinkeyesranclosebyher

Table 57. Examples of two WordNet glosses and of a fragment from the Lewis Carroll text, preprocessed into unsegmented sequences

4.3.2.2 Criteria for evaluation

Since any human processor could easily act as a gold standard for this task, the evaluation criteria will be based on the assessment of the subjective appropriateness of the chunking in randomly or specifically selected outputs as well as by matching the algorithms' outputs with the original, segmented texts and by estimating their performance using well known precision and recall measures. In addition, when possible, comparisons have been made with published results on similar experiments, even though, in many cases the results have not been directly comparable due to different evaluation methodologies and fundamental differences in algorithms and data sets.

4.3.2.3 DDAM-1 automated text segmentation

The DDAM-1 model was subjected to processing the WordNet glosses, a source of textual data which cover many aspects of the English language but which also exhibits a great deal of structure imposed by the purpose of WordNet to serve as a lexical database.

the, and, ing, of, that, ofthe, for, in, ofa, to, or, with, ed, inthe, ina, sof, ment, ation, ly, genus, by, ness, unitedstates, having, man, pro, who, from, ings, witha, onthe, usually, water, large, tion, other, work, ish, plant, small, person, form, at, head, thatis, ated, wood, ring, light, long, king, used, into, ction, white, which, family, under, body, new, ingthe, order, softhe, usedto, part, high, where, black, some, way, edby, fromthe, ingof, off, nation, tothe, any, flowers, yellow, group, tionof, comp, place, read, people, hold, whose, found, back, northamerica, war, green, western, less, herb, hand, over, north, system, very, language, theactof, made, eastern, ward, day, soft, when, blood, play, northern, they, disease, round, mark, ground, withthe, common, hard, word, consistingofa, number, ship, shrub, color, branch, point, house, forma, inwhich, have, horse, good, cell, leavesand, flower, fruit, game, brown, great, bythe, especially, forthe, perennial, food, book, foot, rock, base, former, each, between, blue, fort, asia, human, count, edbya, short, power, usedfor, south, city, roman, board, them, bone, being, grass, public, make, side, government, heart, zation, without, like, ments, free, ession, wall, drug, term, operat, europe, ology, press, graph, lower, many, stand, name, river, leaf, american, come, particular, own, edwith, down

Table 58. A selection of 200 most frequent and interesting patterns discovered by DDAM-1

A selection (200 out of 6800) of some of the most interesting and frequent patterns discovered by DDAM-1 is available in Table 58. The model has acquired some interesting patterns such as *the,* known to be the most common English word, we well as very common determiner *that*, prepositions (e.g., *and*, *of*, *to*, *or*, etc.), common suffixes (e.g., -ed, *-ing*, *-ness*, *-tion*, *-ish*), prefixes (e.g., *pre-*) and other frequent patterns such as *inthe*, *ofthe*, *witha*, *fromthe*, *usedto*.

Some longer patterns have also been discovered but this was likely possible because of the specific formulations of concept definitions and because of the particular thematic

realms of the examples in WordNet lexical database (e.g., *the activity of*, *characterized by*, *the state of being*, *a member of a*, *north america*, *united states*, etc.).

thequalityofbeing, genusofthefamily, characterizedby, thestateofbeing, characteristic, ofsoutheastern, associatedwith, nessthequality, qualityofbeing, consistingofa, resultingfrom, westernunited, northamerican, thepropertyof, mediterranean, americahaving, theactivityof, unitedstates, northamerica, yellowflower, anyofseveral, monetaryunit, whiteflowers, northwestern, amemberofthe, southamerica, southeastern, stateshaving, specialistin, development, ofsomething, information, whiteflower, performance, havinglarge, formationof, ofthefamily, orsmalltree, theactionof, presidentof, planthaving, temperature, competition, personality, havingsmall, instruction, oftheunited, havingwhite, urpleflower, intheunited, especially, government, particular, betweenthe, especially, consisting, ofthegenus, someonewho, considered, containing, cultivated, california, apersonwho, hemisphere, photograph, inwhichthe, positionof, university, resembling, commercial, intendedto, individual, flowersand, electrical, amemberofa, mechanical, experience, genusofthe, revolution, presenceof, dependence, collective, thequality, usedtomake, characters, flowerhead, intheblood, triangular, treehaving, propertyof, inthenorth, blueflower, smallwhite, plantofthe, perennial, partofthe, personwho, something, political, condition

Table 59. A selection of the longest patterns discovered by the 3 layers DDAM-1

An increase in number of layers of the DDAM-1 has caused the acquisition of even longer patterns (shown with blanks for readability) such as *the academic department responsible*, *political party in the united states formed in*, *the basic unit of money in the*, *central nervous system*, *constellation in the southern hemisphere*, *the branch of medicine concerned with the* and the *th president of the united states* ("th" stands for the n^{th} where *n* is a number higher than 3), a pattern that occurs around 30 times in the input data.

rosion, peciallya, includ, humanbeings, dictionary, mbr, northeastern, pples, veral, forthe, elevator, pointo, give, declarationofindependence, 1896, treme, ounitedstates, engine, achi, noun, yment, chest, ith, confidence, unconscious, ussi, divisible, border, alesorder, endence, ngp, capillar, clustersof, ofbeings, container, classical, 1992, excret, active, assingthroug, weddinga, pathogen, pathology, iennial, child, al, vine, shr, mergen, notation, do, kingof, arac, traitof, characteriz, declarationof, gas, amilitar, oxygen, center, cali, americancivilwar1863, that, eae, emp, corp, amic, nannual, producedby, latingth, acolor, gh, catholicchurch, meta, hypertensive, yorkcity, mu, lef, 18thcentury, sabout, ingonlythe, phono, crystal, flavo, throw, queens, becamea, action, esto, year, efferson, license, painful, cluster, ars, ics, neigh, sg, ndf, fabrica,

Table 60. 100 randomly selected chunks from the DDAM-1 output; the 51 underlined chunks have been subjectively deemed appropriate

Based on 100 randomly chosen chunks from the DDAM-1 output, the subjective appropriateness of chunks was judged to have a 51% precision, i.e., 51 chunks out of 100 subjectively corresponded to appropriate words, phrases or collocations while the rest were composed of only partial words, many missing only small parts usually at the beginning or end.

4.3.2.4 DDAM-2 automated text segmentation

The DDAM-2 model was subjected to processing the AIW text. This is a task that could be considered more difficult from at least two perspectives. Firstly, the text source does not seem to exhibit a great deal of structure and, as many general natural language sources, it contains many instances of *hapax legomena* or words that occur only once

(e.g., *pink*, *daisies*, *daisy*) as well as occasionally frequent repetitions of long phrases (e.g., "*as well as she could*" which occurs five times) and idioms. The difficulty of lexical acquisition stems from both the lack of adequate information to define all words and from the apparent difficulty to effectively separate and commit to a segmentation that differentiates appropriately between patterns that exist at morphological, lexical and phrase levels, given the contextual and frequency inconsistencies of those patterns. Secondly, this text source is considerably smaller (i.e., about 40 times) than the WordNet glosses source and while this was beneficial for the computational complexity of analysis it was also detrimental to the capacity to completely bootstrap the word segmentation process.

The analysis was done using six different ambiguity levels in order to study the effect of the parameter values on the analysis and, at the same time to clarify the meaning of the ambiguity parameter itself. The evaluation criterion is simple and consists of measures of recall and precision, all applied in the context of the correct segmentation of the first paragraph in AIW, available in Table 61.

Text	Word count
down | the | rabbit | hole	4
alice | was | beginning | to | get | very | tired | of | sitting | by | her | sister	12
on | the | bank | and | of | having | nothing | to | do | once | or | twice | she | had	14
peeped | into | the | book | her | sister | was | reading | but | it | had | no	12
pictures | or | conversations | in | it | and | what | is | the | use | of | a | book	13
thought | alice | without | pictures | or | conversation	6
so | she | was | considering | in | her | own | mind | as | well | as | she | could	13
for | the | hot | day | made | her | feel | very | sleepy | and | stupid	11
whether | the | pleasure | of | making | a | daisy | chain | would | be | worth | the | trouble	13
of | getting | up | and | picking | the | daisies | when | suddenly | a | white	11
rabbit | with | pink | eyes | ran | close | by | her	8
Total words	**117**

Table 61. The paragraph of 117 words used to evaluate the word segmentations capabilities of DDAM-2; the word segmentation of the paragraph denoted by the 106 pipe symbols, is considered the "gold standard"

This methodology, though not perfect, is simple enough and suitable for a discussion in the context of other published results on similar tasks that could shed some light on the utility of the DDAM model. The paragraph contains 117 words written as 11 separate lines each with a variable number of words separated by " | " (blank-pipe-blank) for improved legibility. The general definitions of precision P and recall R are (TP- true positives, FP – false positive, FN – false negatives):

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}$$

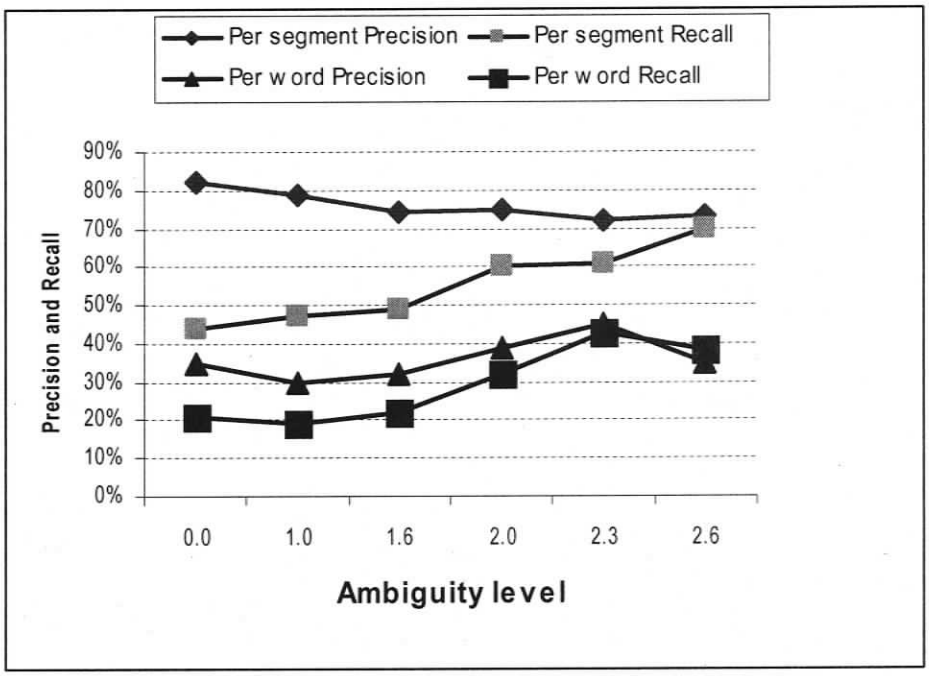
However, their use is dependent on the choices of evaluation explained in the context of the example analysis result in Table 62.

down | the / rabbit / hole
alice / was / beginning | to / get | very / tired /of | **sitting** | **by** | her / sister
on / the | **bank** | **and** | of / having | nothing / to | **do** | once / or / twice | she / had
peep \ ed / into / the | **book** | her / sister | was / read \ ing / but / it | **had** | **no**
pictures / or / conversation \ s / in | it / and | what / is / the / use / of | a / book
thought / alice | **without** | pictures / or / conversation
so | she / was | **considering** | in / her | own / m \ ind | as / well / as / she / could
for / the | ho \ t / day | made / her | feel / very | sleepy / and | **stupid** | **whether**
th \ e | **pleasure** | **of** | making / a / d \ ais \ y / chai \ n / would / be | **worth** | **the** | **trouble**
of | getting / up / and | picking / the | dais \ i \ es | when / suddenly / a | **white**
rabbit / with | **pink** | **eyes** | **ran** | close / by | **her**

Table 62. DDAM-2 result for the segmentation of the target paragraph with an ambiguity parameter equal to 0; the per-segment true positive (TP) are coded as pipes "|", the false positives (FP) as backslashes "\", the false negatives (FN) as forward slashes "/"; the correctly identified words (per word TP) are bold and underlined

For per-segment evaluation, the precision and recall of each pipe sign "|" is assessed. Since in Table 62 there are 50 pipe signs (TP), 11 backslashes (FP) and 56 forward slashes (FN), the per-segment precision is 50/61=82% and the per-segment recall is 50/106=44%. On the other hand, for per-word evaluation, the precision and recall of each exact word is assessed. Since in Table 62 there are 25 correctly identified words (TP), 46 incorrectly identified words (FP) and 117 words in the whole paragraph (TP+FN), the word precision is 25/71=35% and the word recall is 25/117=21%.

Increasing the ambiguity level has gradually decreased the per-segment precision but was beneficial for the per-segment recall and for both per-word precision and per-word recall which seem to peak at an ambiguity level equal to 2.32 bits. Increasing the ambiguity value above 2.32. bits causes precision and recall values to decrease due to the acquisition of additional subword patterns.



	Per-segment		Per-word	
Ambiguity	Precision	Recall	Precision	Recall
$\log_2(0)=0.00$	82%	44%	35%	21%
$\log_2(2)=1.00$	79%	47%	30%	19%
$\log_2(3)=1.58$	74%	49%	32%	22%
$\log_2(4)=2.00$	75%	60%	39%	32%
$\log_2(5)=2.32$	72%	61%	**45%**	**43%**
$\log_2(6)=2.58$	**73%**	**70%**	35%	38%

Table 63. DDAM-2 results for the segmentation of the target paragraph with various ambiguity parameter levels, showing an improvement in per-word precision and recall but a decrease in per-segment precision

Arguably, the evaluation is restrictive since it does not capture the fact that many false positives (e.g., *alicewas*, *therabit*, *hersister*, *onthe*, *init*, *aswellasshecould*, *wouldbe*) are valid collocations which, from pattern discovery, natural language processing and information retrieval points of view could be of interest. It also does not account for some of the false negatives (e.g., *conversation-*, *pick-*, *-ing*) which could also be of interest for morphological processing (Table 64).

down | the / rabbit | hole
alice / was | beginning | to / get | very | tired | of | sitting | by | her / sister
on / the | bank | and / of | having | nothing | to | do | once | or | t \ wice
p \ eep \ e \ d / into / the | book | her / sister | was | reading | but / it | had | no
pictures | or | conversation \ s | in / it | and | what / is | th \ e / use | of | a | book
thought / alice | without | pictures | or | conversation
so | she / was | consider \ ing | in / her | own | mind | as / well / as / she / could
for / the | ho \ t / da \ y | made / her | fe \ el / ve \ ry | sleepy | and | stupid | whether
t \ h \ e | plea \ sure | of | making | a / d \ ais \ y | ch \ a \ in | would / be | worth | the | trouble
of | getting | up / and | pick \ ing / the | da \ isi \ e \ s / when | suddenly | a | white
rabbit | with | pin \ k / ey \ e \ s / ra \ n | close | b \ y / he \ r

Table 64. DDAM-2 result for the segmentation of the target paragraph with an ambiguity parameter equal to 2.0 bits

4.3.2.5 SEQUITUR automated text segmentation

For comparison, SEQUITUR was applied to the segmentation of the very same data. Because previous experiments on artificial sequences have revealed that SEQUITUR exhibits some local minima problems which may cause it to fail to appropriately segment sequences situated at the beginning of the data set, the evaluation paragraph (i.e., the first paragraph of the AIW text) was cut and appended to the very end of the file in order to improve the chances for a correct segmentation.

down | the / rabbit | ho \ le
alice | was / beginning | to / get | very | tire \ d / of | sitting | by | her / sister
on | the / ban \ k / and | of / having | nothing | to / do | once / or / twice | she / had
peeped | into / the | book | her / sister | was | read \ ing | but / it | had / no
pictures / or / conversation \ s / in | it / and / what | is / the | use | of / a | book
though \ t / alice | with \ out | pictures / or / conversation
so | shewas / c \ onsider \ ing / in | her / own | mind | as / well / as / she / could
for | the | hot | day | mad \ e / her | feel | very / sleepy / and | stupid | whether
the | pleas \ ure | of | making | a / d \ a \ is \ y | ch \ ain | would / be | worth | the | trouble
of | getting / up / and | pick \ ing / the | da \ is \ ie \ s / when | suddenly | a / white
ra \ bb \ it / w \ ith | pin \ k / ey \ es | ran | close | by | her

Table 65. SEQUITUR result for the segmentation of the target paragraph

Per segment		Per-word	
Precision	Recall	Precision	Recall
79%	57%	35%	28%

Table 66. SEQUITUR precision and recall results for the segmentation of the target paragraph

As a result, for the target paragraph SEQUITUR has attained a per-segment precision and recall of 79% and 57% respectively (FP=24, TP=60, FN=46) and a per-word precision and recall of 46% and 28% respectively (FP=38, TP=33, FN=84) Table 66. These results are comparable to those of DDAM-2 and only slightly better for the per-segment precision.

4.3.2.6 Discussion

The ultimate purpose of the DDAM-2 model is not word segmentation but the more general goal of achieving a unified representation of sequential data that allows the implementation of advanced information processing algorithms. In this larger context, the specific task of word segmentation of textual data implies the commitment to certain non-overlapped, specific representations that contain controlled amounts of ambiguity. In case of limited input, this commitment is going to deviate from the expected output and no amount of algorithmic sophistication seems to be able make up for the lack of additional data. However, the fact that the model is able to attain modest levels of optimal word segmentation which exceed those of similar models (e.g., SEQUITUR), in a totally unsupervised fashion and using limited data, are positive aspects that warrant a discussion in the context of other published algorithms, on similar tasks.

Model	Corpus	Phonetic transcription	Per word precision	Per word recall
SEQUITUR (Nevill-Manning 1996)	AIW	No	35%	28%
DDAM-2	AIW	No	45%	43%
(Schone 2001)	TREC-IR	No	12%	25%
(Marcken 1996)	CHILDES	Yes	17%	-
SOM (Hammerton 2003)	CHILDES	Yes	18.9%	36.6%
DR (Brent and Cartwright 1996)	CHILDES	Yes	41%	47%
(Schone 2001)	Switchboard	Yes	54%	56%
DDAM-2.1	CHILDES	Yes	61%	65%
Bootlex (Batchelder 2002)	CHILDES	Yes	67.2%	68.2%
MBDP (Brent 1999)	CHILDES	Yes	80% (71% avg)	80% (72% avg)
MBDP variant (Venkataraman 2001)	CHILDES	Yes	80%	80%
DLG (Kit 2005)	CHILDES	No	75%	71%

Table 67. Published unsupervised word segmentation results for unsegmented English language, both transcribed phonetically or not

Model	Corpus	Phonetic transcription	Per word precision	Per word recall
SRN (Simple Recurrent network) (Christiansen, Allen et al. 1998)	CHILDES	Yes	42.71%	44.87%
USEG (Ponte and Croft 1996)	TIPSTER	No	93.6%	90%
MBDP (supervised) (Brent and Tao 2001)	Hansard	No	100%	100%
PPM (Partial Phrase Matching) (Teahan 1998)	Brown	No	100%	100%

Table 68. Published supervised word segmentation results for unsegmented English language, both transcribed phonetically or not

Notwithstanding other evaluation methodology differences and most importantly the fact that the segmentation capabilities of DDAM-2 and SEQUITUR have only been evaluated on a short 117 word paragraph, Table 67 and Table 68 show, as expected, that DDAM-2 results are comparable only with unsupervised approaches to segmentation of general English text (SEQUITUR, Schone, DR, Bootlex, MBDP, DLG) but not with supervised ones (USEG, MBDP-supervised, PPM). The approaches whose results are still clearly superior to DDAM-2 are either supervised approaches (e.g., PPM, MBDP-supervised, USEG) falling outside the scope of this dissertation, or unsupervised approaches based on top down search (e.g., MBDP, Bootlex and DLG).

The MBDP (Model Based Dynamic Programming) family of models whose estimated average word segmentation precision and recall is 71% and 72% respectively (Kit 2005) have been specifically used to model child language acquisition and are specifically optimized to work best on phonetic transcriptions of spoken language with very short utterances as those in the CHILDES corpus (MacWhinney 2000). In particular, MBDP is designed to "make very good used of sentence boundaries and other punctuation" (Brent and Tao 2001) and may be significantly impaired when applied to longer utterances with limited repetition (Schone 2001)[21].

For example, the Bernstein-Ratner87 data set (Bernstein-Ratner 1987) on which some of the MBDP models have been evaluated, is a set of about 10,000 phonetically transcribed spoken utterances between children and their mothers. However, only about 60% of the utterances are unique, the rest being duplicates. In addition, this dataset also contains over 2,000 instances of single word utterances, with some words appearing extremely frequently: e.g., *look* 132 times, *okay* – 241 times, *pekaboo* 52, *yeah* 189, etc. It is clear therefore that particular statistical properties of such data sets may influence the evaluation results of models that specifically take advantage of them. The high frequencies of certain features and most importantly the existence of single word utterances can only improve the chances for the perfect segmentation of those single

---

[21] Page 58

words. In addition, such highly frequent single word utterances will also become functionally similar to text separators such as commas, periods and stop words and again help the overall segmentation process. Finally, the single word utterances, accounting for about 350 unique words and over 20% of all utterances in the dataset, can also be regarded as a lexicon that is provided upfront to the segmentation model and hence may cast serious doubts over the unsupervised nature of the segmentation process.

English	Correct segmentation
yeah he's holding the balloon	y& | hiz | holdIN | D6 | b6lun
paul can put his finger through mommy's ring	pOl | k&n | pUt | hIz | flNgR | Tru | mamiz | rlN
you want me to talk on the telephone	yu | want | mi | tu | tOk | an | D6 | tEl6fon
push it in	pUS | It | In
a bat	6 | b&t
judy can feel daddy's scratchy face	Gudi | k&n | fil | d&diz | skr&ci | fes
balloons all gone	b6lunz | Ol | gOn
it's not a cow	Its | nat | 6 | kQ
smells good	smElz | gUd
will you comb my hair	wIl | yu | kom | m9 | h*
i don't want just little bits of it	9 | dont | want | GAst | lItL | bIts | 6v | It
can you do the zipper	k&n | yu | du | D6 | zIpR
wanna get the meat out of his mouth	wan6 | gEt | D6 | mit | Qt | 6v | hIz | mQT
eat	it
is it a girl	Iz | It | 6 | g3l
we didn't see three	wi | dId!t | si | Tri
no that's not a dog	no | D&ts | nat | 6 | dOg
in a little while	In | 6 | lItL | W9l
here's a kitty's nose	h(z | 6 | kItiz | noz
don't chew on the phone say hello	dont | cu | an | D6 | fon | se | hElo
let's see if we can find some food	lEts | si | If | wi | k&n | f9nd | s6m | fud
i don't want any more toast	9 | dont | want | Eni | m% | tost

DDAM-2 segmentation	DDAM-2.1 segmentation
y& | hi \ z | hol \ dlN | D \ 6 / b6lun	y& / hi \ z | hold \ lN | D \ 6 | **b6lun**
pOl | **k&n** | **pUt** | **hIz** | **flNgR** | **Tru** | **mamiz** | **rlN**	**pOl** | **k&n** | **pUt** | **hIz** | flN \ gR | **Tru** | mami \ z | **rlN**
yu / want / mi / tu | **tOk** | an / D6 / tEl6fon	**yu** | **want** | mi / tu | **tOk** | **an** | **D6** | **tEl6fon**
pUS | **It** | **In**	**pUS** | **It** | **In**
6 | b \ &t	**6** | **b&t**
Gudi / k&n | **fil** | **d&diz** | **skr&ci** | **fes**	Gud \ I / k&n | **fil** | d&di \ z | skr&c \ i | **fes**
b6lunz | Ol / gOn	b6lun \ z | Ol / gOn
Its / nat | **6** | k \ Q	Its / nat | **6** | k \ Q
smEl \ z | **gUd**	smEl \ z | **gUd**
wIl / yu | **kom** | **m9** | **h***	**wIl** | **yu** | **kom** | m \ 9 | **h***
9 | dont / want | Gast / I \ ltL | b \ It \ s | **6v** | **It**	**9** | **dont** | **want** | **GAst** | I \ ltL / b \ It \ s | **6v** | **It**
k&n / yu / du | D6 / zIpR	k&n / yu | **du** | D6 / zIpR
wan6 | **gEt** | D \ 6 / mi \ t | Qt / 6v | hIz / mQT	**wan6** | **gEt** | D6 / m \ it | Qt / 6v | **hIz** | **mQT**
i \ t	**it**
Iz / It | 6 / g3l	**Iz** | **It** | **6** | **g3l**
wi | **dId!t** | **si** | **Tri**	**wi** | **dId!t** | **si** | **Tri**
no | D&t \ s | **nat** | **6** | **dOg**	n \ o | D&t \ s | **nat** | **6** | **dOg**
In | 6 / lItL | W9 \ l	**In** | 6 / lItL | **W9l**
h(z / 6 | **kItiz** | **noz**	**h(z** | **6** | k \ lti \ z | **noz**
dont | c \ u | an / D6 / fon | se / hElo	**dont** | **cu** | **an** | D6 / fon | **se** | **hElo**
lets / si | **If** | wi / k&n / f9nd | **s6m** | **fud**	**lEts** | **si** | **If** | wi / k&n | **f9nd** | **s6m** | **fud**
9 | dont / want | **Eni** | m \ % | **tost**	**9** | **dont** | **want** | **Eni** | **m%** | **tost**

Table 69. DDAM-2 and DDAM-2.1 (slightly modified algorithm) segmentations for a randomly selected set of 22 utterances (about 110 words) from the Bernstein-Ratner87 CHILDES dataset; the per-segment true positive (TP) are coded as pipes "|", the false positives (FP) as backslashes "\", the false negatives (FN) as forward slashes "/"; the correctly identified words (per word TP) are bold and underlined

The importance of the specific nature of such datasets, is also suggested by the fact that when applied to the Bernstein-Ratner87 data set (Bernstein-Ratner 1987) (Table 69, Table 71), DDAM-2 shows a significant improvement of the per-segment precision and recall (79% and 76%) though not for the per-word evaluation which remains at 48% precision and 43% recall, due to the limitations of per-word evaluation. On the other hand, DDAM-2.1, a slightly modified algorithm that aims specifically at choosing segmentations with a maximal sum of the prefix/suffix ratios of the compositional elements, is able to attain a significantly improved per segment precision and recall of 79% and 85% and per-word precision and recall of 61% and 65% respectively. This could be explained by the specific nature of the data set alone.

y& | hi \ z | hold \ IN | **D6** | **b6lun**
pOl | **k&n** | **pUt** | **hIz** | **flNgR** | **Tru** | mami \ z / r \ IN
yu | **want** | **mi** | **tu** | **tOk** | an / D6 | **tEl6fon**
pUS | **It** | **In**
6 / b \ &t
Gudi | **k&n** | **fil** | d&di \ z / skr&ci | **fes**
b6lun \ z | **Ol** | **gOn**
It \ s / nat | **6** | **kQ**
smEl \ z | **gUd**
wIl | **yu** | **kom** | **m9** | **h***
9 / dont | **want** | **GAst** | **lItL** | **blts** | **6v** | **It**
k&n / yu | **du** | **D6** | **zIpR**
wan6 | gEt / D6 | mi \ t | **Qt** | **6v** | **hIz** | **mQT**
it
Iz | **It** | 6 / g3l
wi | dId \ !t | **si** | **Tri**
no | D&t \ s / nat | 6 / dOg
In | 6 / lItL | **W9l**
h(\ z | 6 / klti \ z | no \ z
dont | **cu** | an / D6 | **fon** | se / hElo
lEts / si | If / wi | **k&n** | **f9nd** | **s6m** | **fud**
9 / dont | **want** | **Eni** | **m%** | **tost**

Table 70. MBDP segmentation on the evaluation utterances

Though not favourable for MDBP which is an incremental algorithm that attains the best segmentation capabilities towards the end of the dataset, the segmentation of the randomly selected utterances is shown in Table 70. The per-segment and per word precisions and recall are shown in Table 71 and are comparable to those of DDAM-2.1.

	Per-segment		Per-word	
	Precision	Recall	Precision	Recall
DDAM-2	79%	67%	48%	43%
DDAM-2.1	79%	85%	61%	65%
MBDP	82%	80%	62%	60%

Table 71. DDAM-2 and DDAM-2.1 (slightly modified algorithm) segmentations results DDAM-2 and DDAM-2.1 per-segment and per-word precision and recall are 79% and 67% respectively, and 48% and 43% respectively; per-segment and per-word precision and recall are 79% and 85% respectively, and 61% and 65% respectively

However, the fact that the precision and recall still do not come close to the highest levels of MBDP model (i.e., 80%), suggests that an even deeper underlying problem may be the cause. Despite various additional attempts to alleviate this situation through slight variations of the non-overlapped decomposition algorithm, word segmentation at precision and recall levels comparable to MBDP seems to be unattainable by the DDAM-2 model. The fact that DDAM-2 was able to pick up some long patterns but leave them unsegmented (e.g., *youwantmeto*, *onthetelephone*, *dontwant*, *wecanfind*) while, at the same time, was able to correctly segment single words such as *daddys*, *finger*, *scratchy*, *balloons* etc., indicates that the limited local information on which the DDAM-2 algorithm makes segmentation decisions is just not enough to attain the segmentation of longer patterns while maintaining the correctly segmented single words. These two apparently contradicting objectives (i.e., breaking up longer patterns while not splitting up single words) seem to be achievable only by using a more informed, more global cost function, an approach which was specifically avoided in order to keep the complexity of the model as low possible and the model itself as biologically plausible as possible. The slight modifications in DDAM-2.1 (i.e., choosing segmentations with maximum prefix/suffix ratios of the elements in the composition) have yielded significant improvements but per-word evaluation remains still limited due to the occasional segmentation at sub-word level (e.g., *mami-s*, *dady-s*, *balloon-s*, etc).

To generalize on these observations, which suggest that word segmentation is unattainable by approaches based on limited local information only, one could regard the word segmentation task from the most general perspective, i.e., from the perspective of all possible segmentations. Because, for a sequence of length n, there are 2^{n-1} possible segmentations, a top-down exhaustive search for the best segmentation of a string

(according to some cost function) is unfeasible for long strings (e.g., for $n > 32$ the number of possible segmentations is over 4 billions) since one has to iterate through exponentially many possibilities. From this perspective, the unsupervised word segmentation capabilities of the models in Table 67 applied to sequences which are thousands of characters long are nothing but an amazing feat from the perspective of a top-down exhaustive segmentation search applied to long sequences. Without any apriori information (i.e., completely unsupervised approach), since all possible segmentations have equal probability, the exponential number of segmentations also dismisses any discussion about the possibility to randomly choose the correct one, for long sequences.

One reasonable way to approach a top-down processing is the Viterbi search employed by MBDP and Bootlex models which sacrifices their biological plausibility (Hammerton 2003). A second one, used initially by De Marcken and which is also employed in the DDAM-2 memory model, is to construct a bottom-up, hierarchical representation of input data form which to derive a segmentation. In case of DDAM-2, the bottom-up representation is also overlapping and redundant and the subsequent non-overlapped representation (or segmentation) is derived from it in a more informed, top-down manner. Contingent on the sparseness of the pattern space which is often extreme in case of long, real sequences, this top-down step will start much further from the one-segment per sequence situation.

Finally, a specific feature of DDAM-2 alone consists of the fact that the commitment to a non-overlapped representation (i.e., segmentation) does not imply that the internal representation of a string composition needs to be changed, since the segmentation can be done only on a working memory copy (or clone) of the internal, optimal (and overlapped) representation. Formally, this step equates to decreasing the generality of a composition from that of generalized combinatorial composition, which allows non-zero negative terms, to that of regular combinatorial composition in which all negative terms are zero. As any associative memory with both read/write capabilities, the advantage of this feature is that the original representations in the AM model can be dynamically updated (i.e., upon incorporating additional data in the model) in an online manner. As a result, future

segmentation choices will immediately reflect the integration of additional evidence into the model. This feature brings the model closer to a biologically plausible model of human information processing, which are also known to be very dynamic.

4.3.3 Grammar induction

The following experiment follows the same principles as context dependent grammar induction on artificially generated languages and is meant to test the generality of the DDAM-2 model, which, using the very same principles, is also able to tackle the task of grammar induction from natural sequences. In addition, the discussion of the results will provide an opportunity for extrapolation to more advanced approaches, even though they may be speculative in nature and beyond the scope of this dissertation.

4.3.3.1 Data sources

The source of natural sequences in this experiment is represented by the segmented version of the AIW text, containing one full paragraph per line of text, and written in lower case in order to simplify the induction task, to increase the yield of pattern discovery and to reduce the number of rules in the resulting grammar.

4.3.3.2 Criteria for evaluation

As in the previous unsupervised grammar induction experiments on artificial sequences presented in this chapter, there is a lack of evaluation criteria. This has led to subjective evaluation approaches consisting only of the qualitative estimation of the usefulness and usability of syntactic patterns and equivalence classes discovered by the model. A comparison with other grammar induction models (e.g., ADIOS, SEQUITUR) is not performed due to both limitations of the available demo versions (e.g., ADIOS) which would need significant modification in order to accommodate new data. In addition, the evidence already gathered from the results obtained for synthetic sequences does not warrant the amount of work required for a rerun of experiments on natural sequences given that little or non-significant differences are expected.

4.3.3.3 DDAM-2 grammar induction results

Subjected to the AIW text, DDAM-2 has created optimal representations of all paragraphs in the text from which it has derived two grammars. The morphological grammar comprised around 6000 rewrite rules while the lexical grammar had around 10,000 rules.

know	ledge | s | n
dream	y | ing | ed
shar	ply | p | k | ks | e | ed
carr	oll | ier | ied | y

Table 72. Examples of prefix based morphological equivalence sets discovered by DDAM-2 in the AIW text

ra | ha | gi | e	ven
eng | encour | man	aged
bran | stea | custo	dy
no | some	body
b | g | n | al | some | t	one
argu | parch | mo | oint	ment
viol | sil | cont | s | impati | inv | ev | anci | l | m | r	ent
wal | li | as | ba | mar | in | cho | remar | loo | than | shrie	ked
hope | truth | delight | beauty | wonder | row | doubt | dread | mourn	ful

Table 73. Examples of suffix based morphological equivalence sets discovered by DDAM-2 in the AIW text

As it can be easily seen in Table 72 and Table 73, the morphological pattern discovery is of limited value for a text such as AIW. Besides the acquisition of some useful prefixes (e.g., *-y*, *-ing* and *–ed* attached to the stem *dream*) and lists of morphemes (e.g., the ones that end in *–ful*), the many false positives (e.g., *shar*, *carr* which are not stems), render the approach limited. One remarkable result, and typical of DDAM-2 which is not making segmentation decisions greedily, is the differentiation between the sets of words *brandy*, *steady*, *custody* and the words *nobody* and *somebody*, even though they all end up in *–dy*.

From the perspective of lexical equivalence class acquisition, results are more interesting. This is probably due to the sparser nature of the lexico-syntactic patterns and to their closeness to the linguistic realm of semantics. This property of being able to dynamically define equivalence sets of items sharing a particular context could form the basis of advanced semantic acquisition and processing algorithms.

I	_should_ | 'm | 'll_ | _don't_ | _can_ | _know_ | _must_be_ | 've_ | _beg_your_ | _was_going_to_ | 'll | 'd_ | _can't_ | _think_i_can_ | _was_ | _shall_ | _had_ | 've | 'd | _could_not_ | _gave_her_
she	had_ | thought | was_ | found | fell | saw_ | began_ | put | came_ | might_ | got | could_ | scolded | remembered | would_ | heard | heard_ | felt_ | went_ | should_ | went_on,_ | felt | had | stood | walked_ | came | looked_ | went | wanted | made_ | got_ | is | sentenced | decided | might
_out_of_the_	house, room, wood
,'_said_the_	mouse | duck | dodo | mouse,_ | youth | caterpillar | pigeon | footman | duchess | cat | march_hare.__ | hatter | dormouse | king | queen | gryphon | mock_turtle | cook | white_rabbit
,_she_	ran_ | tried | came_upon_a_ | found | kept_ | ran | found_ | thought | went_on,_` | came | made | pictured

Table 74. Examples of lexical equivalence classes discovered by the DDAM-2 model in the AIW text; (blanks are replaced by underscores)

The most eloquent example of an equivalence class in Table 74 is the one that shares the prefix context *said the ...* which was used to dynamically define the equivalence set of the many characters in the AIW text (e.g., *hatter*, *mouse*, *dodo*, etc.). Though clearly affected by the existence of various separators in the text which leads to the creation of multiple equivalence classes for very similar but different contexts (e.g., "_she_" and , "_she_" differ only by a comma) the method seems applicable.

Prior context(s)	Discovered pattern	Next context(s)
	the little	{crocodile, busy bee}
	she's such a capital one for catching mice	
{silent}	for a minute or two	{she stood, looking, sobs}
	, when suddenly a	{white rabbit, footman in livery}
{watch, tongue hanging, took the hookah}	out of its	{waistcoat-pocket, mouth}
	three legged	{table, stool}
{tried, forgotten, took up, taking}	the little golden key	{in the lock, and hurried, and unlocking the door}
{it}	wasn't very civil of you to	{offer, sit }
	alice had no idea what	{latitude was, to do}
{very, quite, getting}	tired of	{sitting, being all alone, swimming, this}
	the white rabbit blew three blasts on the trumpet	
	and the moral of that is	– {"..."}
{who's to go, you're to go, got to come, as far}	down the chimney	{as she could}
{I don't, she doesn't}	believe there's an atom of meaning in	{it}

Table 75. Examples of word patterns discovered by the DDAM-2 model in the AIW text; the contexts (prior and next) of the patterns are also included

Additional examples, some of which are included in Table 75, could also be of interest if larger contexts are taken into account. For example, the set {*watch*, *tongue*, *hookah*} of objects which share the property of being able to be "taken out of their" containers (i.e., *waistcoat-pocket* and *mouth* respectively) which, in turn, are objects which share the property of being containers.

Semantically similar contexts		Aligned contexts			
shutting people up like telescopes		people		shut up	like telescope
I could shut up like a telescope		I	could	shut up	like telescope
I must be shutting up like a telescope		I	must be	shut up	like telescope
I'm opening out like the largest telescope		I	am	open out	like telescope

Table 76. Examples of semantically equivalent contexts whose alignments requires more sophisticated algorithms

Finally, a discussion of the transition to even higher processing levels, i.e., semantic processing, is warranted. In this context, one has to recall that the deterministic nature of the DDAM-2 model allows input data to be unambiguously represented and retrievable. These are positive aspects. However, the same deterministic nature could also be regarded as a fundamental limitation which prevents the exact alignment of patterns which may be very similar semantically but may differ slightly syntactically. Achieving the alignment of contexts such as those in Table 76 calls for additional processing, a natural extension of the DDAM-2 model and beyond the scope of this dissertation. Such an approach could take the exact, deterministic syntactic pattern discovery to a level which could account for the syntactic and semantic similarity between contexts such as those in Table 76. This would lead to the creation of the dynamic equivalency sets {*people*, *I*}, {*could*, *must be*, *am*}, {*shut up*, *open out*} despite the small inadvertencies in alignment and could be regarded as a primitive form of lossy compression or *automatic summarization* and a necessary step towards more advanced semantic processing. However, because it removes the deterministic nature of the DDAM-2 model, the approach effectively becomes a *non-deterministic sequence alignment* that aims at creating *lossy or abstracted representations* in which certain non-important features for the task at hand may be "ignored" (e.g., the *ing* in *shutting* and *opening*, the *a* in *like a telescope*, the adjective *largest*). The ability of "learning to ignore non-salient features" seems to be the mechanism by which a model could automatically simplify the algorithmic properties of sequences, align them and derive dynamically interesting and useful equivalence sets. This opens the possibility of unsupervised acquisition of lexico-semantic categories which together with *context dependent grammar induction* seem to be both prerequisites of advanced semantic processing.

4.3.4 Medical Natural Language Processing (NLP)

One could argue that general NLP tasks such as the ones presented so far are more difficult and complex than specialized, medical NLP. The reduction in difficulty appears to be caused by the higher degree of structure exhibited by textual artefacts from a specialized, professional discourse, such as the medical discourse. This has formed the rationale for not focusing the evaluation solely on medical language processing.

However, in the restricted, domain-specific context of medical language processing, one could also argue that, at the same time, processing still remains very complex, a situation that has to do with the overall complexity of human body and with the issues of high precision requirements of processing. Textual medical descriptions, while occasionally semi-structured (e.g., in a Subjective-Objective-Assessment-Plan or SOAP format) can be also highly complex and may draw from a terminological base comprising hundreds of thousands of terms. This being said, it is reasonable to consider that the fundamental processing principles of medical NLP should be largely similar to those of general sequence processing. This is why this section of experiments follows previous experiments on artificial sequences and general text.

4.3.4.1 Data sources

The data sources used in the following experiments are the MedTest collection (Hersh, Hickam et al. 1994) and the International Classification of Diseases, revision 10 (ICD10) (1992).

MedTest is a collection of 75 queries by 2,344 documents used for evaluation of the SAPHIRE information retrieval system (Hersh, Hickam et al. 1994). Each document (see Appendix 3 for two examples) contains an abstract and metadata (title, authors, journal, and MeSH index terms). The collection was originally created for the evaluation of the MEDLINE system in clinical settings, and was later adapted for the evaluation of retrieval systems in biomedicine.

ICD10 is a mono-axial classification of diseases which comes as a multiple language resource (French, English, German) in the common form of a relational database. The

subset of data used for experiments consisted of the English entries in the hierarchy, which accounted for some 30,000 individual strings representing biomedical concepts.

4.3.4.2 Criteria for evaluation

The evaluation of unsupervised models for information processing, particularly for grammar induction, is notoriously difficult due to the lack of the gold standard of what a good segmentation is, or a good grammar must look like. Therefore, clear criteria for the evaluation of the medical language processing capabilities of the DDAM model have only been possible in the case of the segmentation of compound biomedical terms and in the case of the somewhat artificial task of text segmentation of MedText abstracts for which the separators have been removed. The rest of the evaluation was based on the qualitative assessment of the "interestingness" of the results, which, indeed, in a particular case have exceeded any expectation: the DDAM model was able to pick up an *extremely significant regularity* present in two distinct MedTest abstracts.

4.3.4.3 Unsupervised morpho-segmentation of compound medical terms

The morpho-segmentation of compound medical terms aims at segmenting the terms into their compositional building blocks. In the terms of the DDAM model, this equates to reducing overlapped compositions of terms to non-overlapped compositions, or to deepening the valleys in the corresponding Dyck paths until they reach a depth equal to zero, such as in Figure 60.

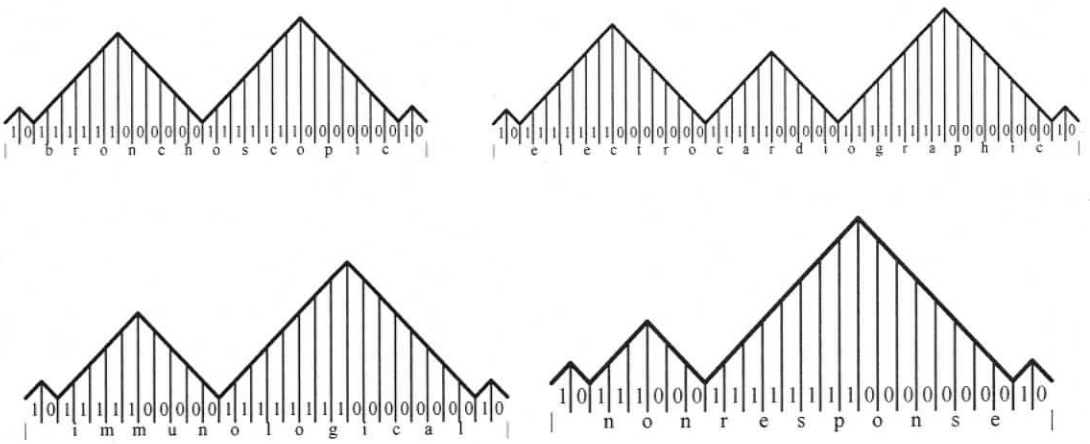


Figure 60. Examples of non-overlapped compositions which correspond to the morpho-segmentation of the compound terms bronchoscopic, electrocardiographic, immunological and nonresponse

Due to the need for high accuracy, all morpho-segmentation approaches for medical language processing are supervised and based on significant manual work. This makes it difficult to compare them with the completely unsupervised morpho-segmentation of DDAM model which would be clearly unable to reach too high precisions and recall scores. Though integrating explicit morphological knowledge in supervised approaches based on the DDAM model is possible and may certainly lead to an improved morpho-segmentation precision, this line of research was left for the future.

For the time being, the assessment of DDAM morpho-segmentation is based only on limited evidence and examples such as those in Table 77. An additional selection of morpho-segmentation examples is available in Appendix 4.

antigen | ic	angi | ocardiographic	cine | angiography
apha | g | ic	pho | n | ocardiographic	chol | angiography
di | phas | ic	angi | ograph | ical	ang | iography
hepato | tr | oph | ic	mye | l | ograph | ically	arter | iography
hypo | b | ar | ic	angi | ographic	autorad | iography
hypoe | cho | ic	arteri | ographic	mict | iography
immuno | blast | ic	chro | mat | ographic	disc | ography
multimer | ic	de | m | ographic	radi | ography
myelo | blast | ic	el | ectr | oencephal | ographic	lymph | ography
myoclon | ic	electro | my | ographic	arthr | ography
nephro | path | ic	encephal | ographic	neur | ography
neuro | path | ic	esophag | ographic	electroencephal | ography
osteo | path | ic	ge | ographic	myel | ography
pre | diagn | ost | ic	mye | l | ographic	echocardi | ography
uretero | pe | lv | ic	neur | ographic	ultrason | ography
	angi | ographic	roentgen | ography
	r | oent | gen | ographic	son | ography
		electromy | ography
		phleborhe | ography
		cardio | graphy
		poly | graph
		scinti | graph

Table 77. Excerpt from the DDAM morpho-segmentation output

Calculating the precision and recall at the morphological segment level is not a valid approach due to the lack of precise rules to define good morphological segments. For example the appropriateness of segmentations such as "angi-ographic", "angio-graphic" "angi-o-graphic" is difficult to assess with precision and recall measures alone. Based on 100 randomly chosen segmentations from the DDAM output, the unsupervised segmentation was therefore judged subjectively to have an approximate precision of 60%, i.e., 60 out of 100 compound words were deemed to be segmented appropriately.

4.3.4.4 Text segmentation

The text segmentation for medical text works on similar principles as text segmentation for general language. This task is somewhat artificial in nature since there are relatively few instances when text segmentation at word level is useful in practice (e.g., one of them is Optical Character Recognition). The MedTest collection was preprocessed so that all separators (blanks, punctuation, parentheses, etc.) were completely removed. Out of the 2344 documents, the document with id 1,000 was selected for evaluating the per-segment precision and retrieval. The per-word evaluation of was omitted due to the expected low values of precision and recall.

the / anti | emetic | efficacy / of | methylprednisolone | compared / with | metoclopramide | in | out \ patients / receiving | adjuvant / cmf | chemotherapy / for | breast / cancer | a / randomized / trial
a / randomized / trial | was / performed | comparing / the | antiemetic | efficacy / of | methylprednisolone | m \ pn | and | metoclopramide | mcp | in / 6 \ 0 | breast / cancer / patients | eligible | for | outpatient | adjuvant / chemotherapy | with | cyclophosphamide / methotrexate | a \ nd | 5 / f \ u | cmf | at / the / time / of / the \ ir | first | chemotherapy | course | patients / were / randomized / to / receive | either | m \ pn | 37 \ 5 / mg | or | mcp | 1 / mg / kg | bo \ t \ h | administered / in | 3 / e \ qu \ al / doses | iv | jus \ t / prior / to | chemotherapy / and | the \ n | im | 6 / and | 12 / hour \ s / after / treatment | patients / receiving / m \ pn | experienced | significantly / less | naus \ e \ a | p / less / than / 00005 | and / vomiting | p / less / than / 00005 | and | antiemetic | protection | was | maintain \ ed | in / patients / receiving | multiple | chemotherapy | courses | complete / protection | 0 / e \ me \ si \ s / was / observed / in | 5 \ 8 / of | patients / receiving / m \ p \ n | as / compared / with | 20 | of / patients / treated / with | m \ c \ p | p / less / than / 0005 | the / most / frequent | side / effects / were | facial | f \ lus \ h / in | 3 \ 8 | of / patients / and | somnolence | in / 15 | of | patients / receiving / m \ pn / a \ nd / m \ cp | respectively | complete / protection | from / c \ m \ f \ induced | gastrointestinal | side / effects | was / observed / in | two | third \ s | o \ f / our | patients / receiving / a \ ntiemetic | mp \ n | treatment / in | these / patients | administration / of | the | maximum | cumulative | cmf | dose | w \ as / possible | without / i \ m \ p \ air \ ing / their | quality / of | life | m \ p \ n / at | the / dose | and | schedule | reported | is | an | affective | anti \ eme \ tic / drug / s \ uitable / for | use | in / breast / cancer | out \ patients / receiving | adjuvant / cmf | therapy

Table 78. Result for the segmentation of the target paragraph with an ambiguity parameter equal to 1 bit; there are 124 true positives, 51 false positives and 105 false negatives which account for a segmentation precision of TP/(TP+FP)=124/175=71% and segmentation recall of TP/(TP+FN)=124/229=54%

The results in Table 78 are comparable to that of general text segmentation in terms of precision (i.e., 71%) but show a lower recall (i.e., 54%) due to the existence of many long regularities that have not been split (e.g., *comparedwith*, *efficacyof*, *patientsreceiving*, *adjuvantcmf*, *assideeffects*, *arandomizedtrial*, *thedose*, etc).

The evaluation using precision and recall of word level segments disregards the existence such longer regularities whose usefulness is due to the fact that they have lower frequencies in document collections. This may lead to a better efficiency of retrieval systems, which are commonly making use of inverse frequency measures to calculate the relevance of a document to a query (i.e., the rarer a term the higher its relevance to a document).

4.3.4.5 Lexical equivalence set induction

The induction of lexical equivalence sets is a natural by-product of the grammar induction and segmentation capabilities of the DDAM model. The equivalence set induction process is based on contextual (prefix, suffix, diafix) similarities of various terms and results in equivalence sets which, in some instances, may correspond to lexico-semantic classes. This discussion will only highlight some interesting examples that shed light on the utility of pursuing this kind of functionality.

The complete representation of the titles and abstracts of the 2344 documents in the MedTest collection followed by a grammar induction procedure with an ambiguity

parameter of 1.0 bit, has yielded a grammar with 85,276 rules. Metadata such as author names, journal names and MESH index terms were not processed. A comprehensive selection of suffix and prefix based equivalence sets is available in Appendix 5.

Though some equivalence sets may appear of limited use due to their abstract nature (e.g., #2,#3, #7 in Table 79) others provide a quick summary of the MedTest collection by collecting together pathological entities that are referred to in the documents (e.g., #2,#3, #7 in Table 79 and #2, #3, #4, #5, #6, #7 in Table 80).

1	_in_patients_with_	{a_ | acute | acute_ | acute_myocardial_infarction_ | advanced_ | advanced_breast_cancer | advanced_breast_cancer_ | aids | alcoholic | aplastic_anemia | ascites_due_to_ | barrett | barrett's_esophagus | barrett's_esophagus_ | barrett's_syndrome | cardiac | chronic | chronic_ | cirrhosis | cirrhosis_and_ | cirrhosis_and_ascites | cirrhosis_of_the_liver | complex | complex_ | congestive_heart_failure | crohn | crohn's | crohn's_disease | extensive | heart_ | hemophilia | hepatic | hepatic_ | impaired | inducible | inducible_ | liver_ | liver_cirrhosis_and_ | malignant_ | mastocytosis | mitral_valve_prolapse. | myocardial_infarction | or_without_ | other_ | previously | prior | severe_ | symptoms_of_ | symptoms_of_gastroesophageal_reflux | systemic_ | thalassemia | thalassemia_major | transient | ulcerative_colitis}
2	_studies_	{comparing_ | demonstrated_ | examining | included_ | indicate | indicate_that_ | investigating_ | performed | reported | reported_ | showed | suggest_that_}
3	_study_was_	{carried_out_on_ | conducted | designed | designed_to_ | performed | undertaken | undertaken_in_ | undertaken_to_}
4	_was_used_to_	{assess | evaluate | examine | measure}
5	_were_significantly_	{better | better_than_the_ | different | different_ | elevated | greater | higher | improved | increased_in_ | less_ | lower | more_ | reduced | smaller | smaller_in_}
6	blood_	{bank | centers | clots | components | flow | gas | gases | loss | pressure | pressures | products | supply | transfusion | transfusions | vessels | volume}
7	cerebral	{_and_ | _arterial_ | _arterial_spasm | _blood_flow | _blood_flow,_ | _blood_flow_ | _blood_flow_(| _blood_flow_and_ | _blood_flow_was_ | _blood_vessels | _cortical_ | _energy_ | _ischemia._ | _ischemia_and_ | _perfusion_pressure | _protection._ | _vascular_ | _vasospasm_ | _vasospasm_in_}
8	-year-old_	{boy | female | girl | male | man | woman}

Table 79. Selected lexical equivalence sets from the MedTest collection induced by common prefix patterns

Other sets simply bring together words that share lexico-semantic properties. For example, the set #8 in Table 79, contains references to various age groups of people. The sets #8, #9 in Table 80 contain mostly temporal concepts while the set #1 in Table 80 collects the various types of medical care. Sets such as #10, #11, #12, in Table 80 group mostly correct anatomical and physiological concepts, the set #13 in Table 80 lists various types of transplants, and sets #14, #15 refer to various types of therapies and

chemotherapies. Finally, the set #16, in Table 80 amalgamates various characteristics of patients including their number or their pathological or physiological status.

1	{coronary | critical | extended | health | intensive | medical | patient | primary | supportive}	_care_
2	{congestive | idiopathic}	_cardiomyopathy,_
3	{_(guillain-barr:e_ | _bowel_ | _guillain-barr:e_ | _malignant_ | _nephrotic_ | _patients_with_barrett's_ | _polycystic_ovary_ | _reiter's_ | _this_ | _with_barrett's_ | acquired_immune_deficiency_ | acquired_immunodeficiency_ | bannwarth's_ | barrett's_ | bartter_ | bartter's_ | fisher's_ | hellp_ | hepatorenal_ | immunodeficiency_ | -like_ | polycystic_ovary_ | respiratory_distress_ | reye-like_ | 's_ | sezary_ | this_ | uremic_}	syndrome
4	{_and/or_ | _and_viral_ | _bacterial_ | _chronic_ | _of_bacterial_ | _tuberculous_ | _with_listeria_monocytogenes_ | bacterial_ | purulent_ | tuberculous_ | tuberculous_}	meningitis
5	{_thrombocytopenic_ | anaphylactoid_ | henoch-sch:onlein_ | sch:onlein-henoch_ | thrombocytopenic_}	purpura
6	{_proliferative_ | membranous_ | necrotizing_ | proliferative_}	glomerulonephritis
7	{breast | cervical | lung | metastatic | national institute | ovarian}	_cancer_
8	{_during_the_ | _during_the_exposure_ | _during_the_follow-up_ | _during_the_ischemic_ | _during_this_ | _follow-up_ | _incubation_ | _latency_ | _month_follow-up_ | _over_a_four-year_ | 4-week_ | -day_treatment_ | during_this_ | follow-up_ | ischemic_ | month_ | -month_ | post-operative_ | six-month_ | study_ | treatment_ | week_ | -week_ | year_ | -year_}	period
9	{_days_ | _five_years_ | _h_ | _months_ | _weeks_ | _years_ | days_ | h_ | months_ | weeks_ | years_}	later
10	{_basilar_ | _carotid_ | _coronary_ | _for_coronary_ | _middle_cerebral_ | _on_hepatic_ | basilar_ | coronary_ | femoral_ | pulmonary_ | vertebral_}	artery
11	{affected | autonomic | cranial | median | parasympathetic | peripheral | phrenic | redundant}	_nerve_
12	{_ascitic | _extracellular | ascitic | body | cerebrospinal | extracellular | transformer pyrolysate}	_fluid_
13	{_after_ | _cadaveric_renal_ | _cardiac_ | _in_bone_marrow_ | _in_the_ | _renal_ | bone_marrow_ | marrow_ | renal_}	transplant
14	{_non-cross-resistant | adjuvant | cancer | combination | initial | maintenance}	_chemotherapy_
15	{_adjuvant | _endocrine | adjuvant | antibiotic | chemohormonal | endocrine | induction | pacemaker}	_therapy_in_
16	{2 | 3 | 5 | 6 | 7 | 10 | 11 | 12 | 14 | 15 | 17 | 20 | 22 | 24 | 25 | 28 | 30 | 31 | 32 | 36 | 40 | 42 | 45 | 47 | 48 | 50 | 57 | 58 | 68 | 72 | 73 | 80 | 84 | 89 | 105 | 120 | 131 | 164 | 200 | 298 |). | ,_all | ,_nine | ._many | ._most | ._three | ._two | ._we_conclude_that | ._when | _12 | _20 | _among | _and_two | _breast_cancer | _cirrhotic | _consecutive | _control | _diabetic | _evaluable | _female | _from_two | _hospitalized | _in_12 | _in_28 | _in_57 | _in_diabetic | _in_eight | _in_many | _in_selected | _in_some | _in_the_5 | _in_two | _male | _nine | _of_11 | _of_15 | _of_18 | _of_200 | _of_21 | _of_25 | _of_50 | _of_all | _of_eight | _of_five | _of_some | _of_those | _patients. | _postmenopausal | _premenopausal | _selected | _stroke | _symptomatic | _ten | _therapy_for | _to_treat | _transfused | _treated | _treatment_for | _unselected | adult | all | almost_all | among | between | cirrhotic | consecutive | eight | er-positive | euthyroid | fifteen | five | for | four | from | identifying | in | include | many | of | -one | postmenopausal | seven | -seven | seventy-four | six | -six | stroke | ten | those | three | -three | to | transfused | transplant | twenty | twenty-four | two | -two | when}	_patients_with_

Table 80. Selected lexical equivalence sets from the MedTest collection induced by common suffix patterns

The source of all equivalence classes in Table 79 and Table 80 is exclusively the text in the MedTest collection. While this exercise may appear trivial and being an unsupervised task may lack clear evaluation methodologies, the capability to discover regularities in data in a totally unsupervised fashion is very important and speaks to the usefulness of the DDAM model. The importance is underlined and demonstrated by the acquisition of one equivalence set whose significance exceeds that of any sets in Table 79 and Table 80. This equivalence set is formed from two distinct grammar rules whose commonality is the full paragraph "*patients with short duration of disease were especially prone to be antibody negative in serum but positive in csf. significant rise in serum antibody titers was seldom demonstrated in patients treated with*". The sheer length of this regularity implies that its frequency to appear by chance in two distinct documents is extremely low. Its occurrence is therefore an extraordinary event which is highly indicative of only one, virtually unequivocal scenario: the text must have been copied from one document into the other in some way. A simple search on this paragraph in the MedTest collection turned out two distinct abstracts with the identification number 803 and 1972, respectively (see Appendix 2). The inspection of the metadata revealed that the documents have been written by the exact same authors (i.e., Stiernstedt, G., Granstrom M., Hederstedt, B., Skoldenberg, B.) but published in different years (i.e., 1985 and 1987) in two different journals (i.e., J Clin Microbiol, 21(5):819-25 and Zentralbl Bakteriol Mikrobiol Hyg; 263(3):420-4).

Though this particular discovery is of limited usefulness in hindsight, the conclusions and analogies that can be drawn from this experiment are important, and extend beyound the obvious application to the approach to the discovery of plagiarism in textual artefacts. Firstly, achieving the same feat with a trivial approach which searches the entire MedTest collection for all substrings up to length n from the collection itself would have taken very long time compared to the couple of minutes in the DDAM model. Secondly, the potential to discover significant (i.e., long) regularities, in various circumstances, is a useful tool for decision making, particularly in time constrained conditions, where filtering of what is significant from what is not, is of importance. Though human capabilities in this area most likely work at a different, high conceptual level, where

patterns with long descriptions are rich and multi-sensorial in nature, the basic mechanism seems to be similar. What was demonstrated by this experiment is only a starting point towards a more advanced emulation of human pattern acquisition capabilities.

4.3.4.6 Similarity based retrieval

It has been argued in Chapter 2 that similarity based search and retrieval is one of the most important goals of this research. A glimpse of how it might work was offered through the analogy of the task with a hyperspace telescope that allows us to visualize a multidimensional conceptual space through an objective with a given bit radius. The possibilities and limitations of similarity based retrieval will be explored mostly qualitatively in two distinct experiments: one working at a morphological level and one which works at a morpholexical level.

Upon representing a dataset of 1,700 compound medical terms gleaned from the ICD10 and MedTest data sources, the DDAM model was queried with the term "tachycardia," which was among the terms in the dataset. The retrieved strings, shown in Table 10, are placed in columns corresponding to the concentric hyperspheres having bit radii ranging from 0.0 bits to 2.0 bits.

0.0	1.0	1.6	2.0
tachycardia	brachychronic	endomyocardial	anthracosilicosis
tachypnea	brachyphalangia	perimyocardial	asepsis
tachypnoea	brachycephalic	postmyocardial	candidiasis
tachyarrhythmias	bradypnea	intramyocardial	diathesis
tachyphylaxis	psittacosis	extracardial	diagnosis
bradycardia	anthracosis	paracardiac	prediagnostic
		intracarpal	endocardium
		intercarpal	epicardium
		carcinoma	myocardium
		methacholine	pericardium
		hyperdense	hyperglycaemia
		lordosis	hypoglycaemia
			hypoglycaemic
			infracostal
			intracostal
			chorditis
			subordinate
			arachnoid

Table 81. The results of the DDAM associative recall on the query "tachycardia" on a collection of 1,700 medical compound terms within a bit radius ranging from 0.0 to 2.0 bits; the columns correspond to concentric hyperspheres with increasingly larger bit radii which, besides the elements in the corresponding column, also include the results in previous columns (i.e., at lower radii)

Given that the query is the center of the concentric hyperspheres, the terms that begin with *tachy-* and end in *–cardia* are expected within a low bit radius. However, at higher bit radii, the associative properties of DDAM combine with the relative high sparseness of the pattern space (only 1,700 term in the dataset) and allow for results such as those that begin with *brachy-* and *brady-* due to the strong association with *bradycardia*. The associative properties of DDAM and the sparseness of the representation space are also responsible for unexpected terms such as *psittacosis,* which, in turn, is clearly associated with *anthracosis* by their common suffix. Further, at 1.6 bits bit radius, entries containing the patterns *-cardial*, *-cardiac* and *-carpal* are retrieved, followed by more distant ones such as *metacholine*, *hyperdense* and *lordosis*, the latter most likely being the result of the association with *anthracosis* and *psittacosis*. At 2.0 bits, though some entries share similarities with the original query, a substantial number of them are the results of indirect associations (e.g., those that end in *–sis*).

In a second example on the same dataset, using the query *hematoma*, the similarity based retrieval mechanism shows the same characteristics, with the distinction that, in this case, there are less spurious hits for low bit radii. At 2.0 bits however, indirect association causes terms such as *hydroxymyristoyl* which appears in the results set likely because of the similarity with *peristalsis* which has obvious similarities with a close match to the original query, namely ***peristomal***.

0.0	1.0	1.6	2.0
hematoma	hematomas	angioedema	hemangiomas
	cephalhaematoma		hemangioma
	hematuria		angiofibriomas
	lymphohematopoietic		hematopoietic
	hepatoma		manometric
	hepatomegaly		prematurity
	peristomal		premature
			perisplenitis
			peristalsis
			hydroxymyristoyl

Table 82. The results of the DDAM associative recall on the query "hematoma" on a collection of 1,700 medical compound terms within a bit radius ranging from 0.0 to 2.0 bits

By now it is clear that this approach to similarity based retrieval is bound to yield results sets that contain spurious hits caused by the combination of associative recall and the potentially significant sparseness of the pattern space. Therefore, a discussion of whether

embarking on a journey to create a technology "bound to make mistakes" is a good idea, is very appropriate at this point.

The first argument to help the case begins with remembering that even humans, especially in the novice stage, make mistakes. The reason is most likely the lack of information and knowledge, which essentially translates in a markedly sparse representation of a problem space which is filled with significant knowledge gaps. As with any novice, this results in unexpected associations and mismatches such as those common in children (e.g., mismatching the bull for the cow). With this in mind, mistaking *hepatoma* or *hemangioma* for *hematoma* does not appear too big a mistake, especially in the case of a novice.

The obvious solution to this problem is to overcome the lack of information and evolve the system from novice to expert. Though the potential for mistakes will always be there, many spurious hits will be weeded out by adding relevant information in the system. Essentially, this implies gradually filling the knowledge gap between a given representation and the ones that are marginally similar to it and that cause the spurious associations and results. To use the hyperspace telescope analogy, adding additional information in the memory effectively "knocks out" spurious elements further from the centre of the telescope objective and makes them retrievable only at higher bit radii. To prove this, the second experiment was run on the same data set but to which the following terms that share similarities with the spurious result *peristomal*, have been added: *stomach*, *periost*, *stomatitis*, *anastomosis*. As a result, on the very same query *hematoma*, the term *peristomal* which was initially retrieved within a 1.0 bit radius, was effectively "pushed" outwards from the query in the conceptual space and was retrievable only at 2.0 bit radius. What this experiment has proved was that, by adding relevant additional information into the system, we were able to overcome problematic associations and weed out spurious results. To generalize, what this proves is the possibility that similarity retrieval can be improved incrementally and continuously by dynamically adding relevant information into a system, a property which is in perfect agreement with the functional principles of CBR and of significant importance to Medical Informatics.

The second argument in favour of this kind of retrieval is based on the commonsense observation that associative recall does not have to happen in a vacuum. The full integration of context sources at phonological, morphological, lexical, syntactic, semantic and pragmatic levels, could provide the basis of a robust mechanism for eliminating spurious hits. Though such integration is unattainable currently and its effectiveness is far more difficult to prove than in the previous example, it is very tempting to conjecture that integrating information from different levels (i.e., increasing the amount of context) is only a matter of scale that leaves processing principles and mechanisms largely the same. Empirical evidence for this second argument could be built by stepping up a linguistic level and attempting similarity based search and retrieval based not only on morphological similarity but also on lexical similarities between textual items.

For this experiment the ICD10 strings have been represented in a DDAM model with two layers: a morphological layer that takes care of token similarities, and a lexical layer that accounts for similarities at lexical level. The query (i.e., the center of the conceptual space) was set to the string "shigella," which represents the name of a micro-organism responsible for a group of human infectious diseases. In ICD10, the entries that refer to these diseases fall under the code A03, Shigellosis, and are shown in Table 83.

ICD10 code	ICD10 string
A03.0	Shigellosis due to Shigella dysenteriae
	Group A shigellosis [Shiga-Kruse dysentery]
A03.1	Shigellosis due to Shigella flexneri
	Group B shigellosis
A03.2	Shigellosis due to Shigella boydii
	Group C shigellosis
A03.3	Shigellosis due to Shigella sonnei
	Group D shigellosis
A03.8	Other shigellosis
A03.9	Shigellosis, unspecified
	Bacillary dysentery NOS

Table 83. Entries in ICD10 referring to the diseases caused by various types of the Shigella micro-organism

Upon querying the DDAM model with the string "shigella" a series of tokens was retrieved within a 3.6 bit radius. In turn, the tokens have caused the retrieval of various ICD10 strings within bit radii ranging from 1.0 to 3.6 bits. The most natural way to display the retrieved items appears to be as a multiple hierarchy where tokens form the

first level and the ICD10 strings form the second level (Table 84). Because the hierarchy is multiple, the ICD10 strings may potentially repeat and fall under multiple categories.

Token radius (bits)	String radius (bits)	Category, ICD10 code	ICD10 string
0.0		**shigella**	
	2.0	A03.0	shigellosis due to **shigella** dysenteriae
	2.0	A03.2	shigellosis due to **shigella** boydii
	2.0	A03.3	shigellosis due to **shigella** sonnei
	2.0	A03.1	shigellosis due to **shigella** flexneri
0.0		**shigellosis**	
	1.0	A03.0	group a **shigellosis** [shiga-kruse dysentery]
	1.6	A03	**shigellosis**
	1.6	A03.9	**shigellosis**, unspecified
	2.0	A03.1	**shigellosis** due to shigella flexneri
	2.0	A03.0	**shigellosis** due to shigella dysenteriae
	2.0	A03.3	**shigellosis** due to shigella sonnei
	2.0	A03.2	**shigellosis** due to shigella boydii
	3.3	A03.2	group c **shigellosis**
	3.3	A03.1	group b **shigellosis**
	3.3	A03.3	group d **shigellosis**
	3.3	A03.8	other **shigellosis**
1.0		**shiga**	
	1.0	A03.0	group a shigellosis [**shiga**-kruse dysentery]
1.0		**kruse**	
	1.0	A03.0	group a shigellosis [shiga-**kruse** dysentery]
1.0		**dysentery**	
	1.0	A03.0	group a shigellosis [shiga-kruse **dysentery**]
	2.6	A03.9	bacillary **dysentery** nos
	3.3	A06.0	acute amoebic **dysentery**
	3.3	A07.0	balantidial **dysentery**
	3.3	A07.9	protozoal **dysentery**
1.0		**group**	
	1.0	A03.0	**group** a shigellosis [shiga-kruse dysentery]
	2.6	T80.3	reaction to blood-**group** incompatibility in infusion and transfusion
	2.6	B95.0	streptococcus, **group** a, as the cause of diseases classified to other chapters
	3.0	E78.0	hyperlipidaemia, **group** a
	3.3	A03.2	**group** c shigellosis
	3.3	A03.1	**group** b shigellosis
	3.3	A03.3	**group** d shigellosis
	3.3	F91.2	**group** delinquency
	3.6	Z63.9	problem related to primary support **group**, unspecified
	3.6	Z63.8	other specified problems related to primary support **group**
	3.6	Z63	other problems related to primary support **group**, including family circumstances
	3.6	A40.0	septicaemia due to streptococcus, **group** a
2.0		**dysenteriae**	
	2.0	A03.0	shigellosis due to shigella **dysenteriae**
2.0		**boydii**	
	2.0	A03.2	shigellosis due to shigella **boydii**
	3.0	B48.2	infection due to pseudallescheria **boydii**
2.0		**sonnei**	
	2.0	A03.3	shigellosis due to shigella **sonnei**
2.0		**flexneri**	
	2.0	A03.1	shigellosis due to shigella **flexneri**
2.6		**reaction**	
	2.6	T80.3	**reaction** to blood-group incompatibility in infusion and transfusion
	3.6	T80.4	**reaction** due to rh factor in infusion and transfusion
	3.6	D50.0	iron deficiency anaemia secondary to blood loss (chronic)
2.6		**support**	
	3.6	Z63.8	other specified problems related to primary **support** group
	3.6	Z63	other problems related to primary **support** group, including family circumstances
	3.6	Z65.3	child custody or **support** proceedings
	3.6	Z63.9	problem related to primary **support** group, unspecified
3.0		**pseudallescheria**	
	3.0	B48.2	infection due to **pseudallescheria** boydii
3.3		**delinquency**	
	3.3	F91.2	group **delinquency**
3.3		**balantidial**	
	3.3	A07.0	**balantidial** dysentery
3.3		**bacillary**	

	2.6	A03.9	**bacillary** dysentery nos
3.6		**custody**	
	3.6	Z65.3	child **custody** or support proceedings

Table 84. The results of the DDAM associative recall on the query "shigella" on the complete collection of about 30,000 ICD10 strings, using an increasingly large bit radius, from 0.0 bit to 3.6 bit

Expectedly, most entries in Table 83 can be found among those retrieved within 0.0 bit radius in Table 84. The only exception is *bacillary disentery NOS* which is retrieved within a bit radius of 2.6 bits, through indirect association, given its obvious lack of similarity with the original query. However, in Table 84 some entries can also be found in multiple categories while others are additions (some relevant, some not so much). For example, triggered by the relevant ICD10 string *bacillary disentery NOS* additional types of dysentery (i.e., *acute amoebic*, *balantidial*, *protozoal*) are also retrieved within a 3.3 bit radius. Another such example is the entry *B48.2, infection due to pseudallescheria boydii* which is retrieved due to its similarity to A*03.2, shigellosis due to shigella boydii.* Advancing towards higher bit radii than 3.3 bits results in obviously spurious results such as those triggered by the token *group* which is associated with *support* and *reaction* and which cause entries such as *Z65.3, child custody or support proceedings*, *F91.2, group delinquency*, *T80.3, reaction to blood-group incompatibility in infusion and transfusion*, etc. to become part of the result set. Preventing them would probably require integration of additional sources such as semantic and pragmatic knowledge that have the potential to indicate the lack of relevance of entries about child support and group delinquency for a set of items whose common theme is infections diseases.

The associative recall results can also be displayed using a graphical paradigm. Though individual items may not be as visible as in the multiple hierarchy in Table 84, the graphical display of results, in conjunction with force-directed automatic layout optimization algorithms, attains what was argued in Chapter 2 to be the functional and structural equivalence of self organizing maps. Upon visually inspecting the resulting map in Figure 61 three distinct clusters are detected.

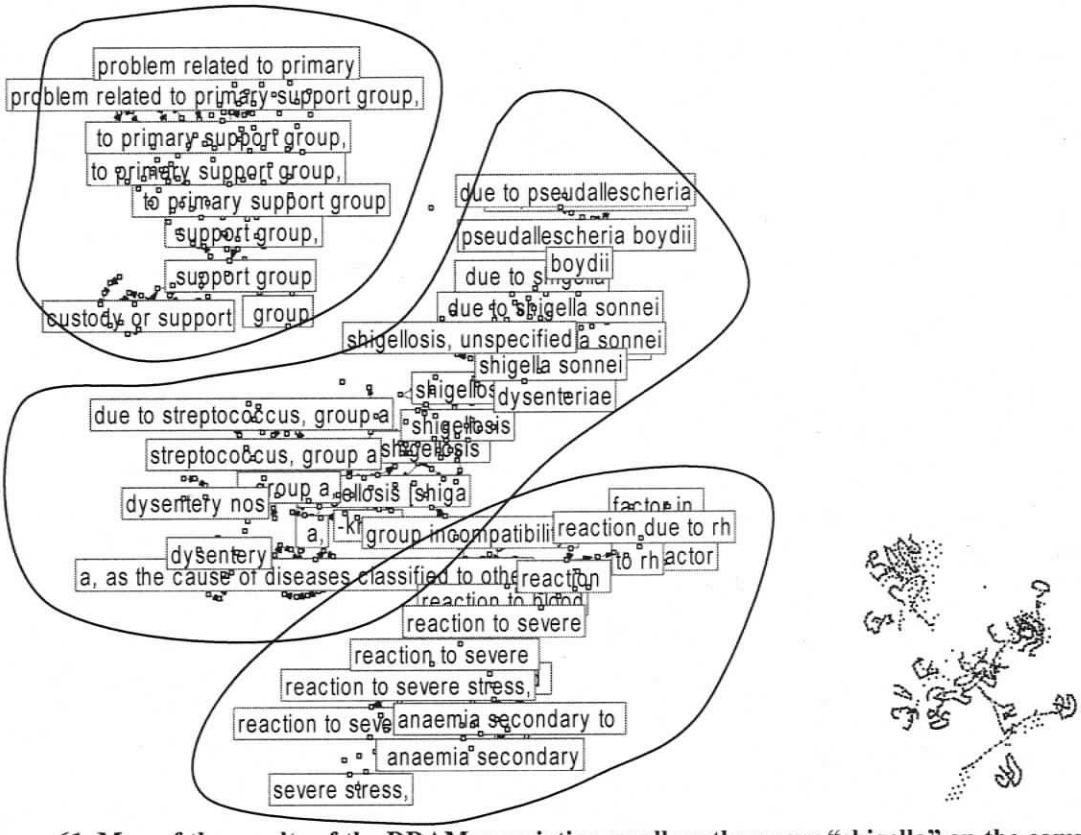


Figure 61. Map of the results of the DDAM associative recall on the query "shigella" on the complete collection of about 30,000 ICD10 strings showing three distinct clusters and demonstrating functional and structural equivalence to self organizing maps

The three clusters correspond clearly to three "themes" that characterize the items in the retrieval set:

- the middle cluster which corresponds to the main theme that has to do with various micro organisms and infectious diseases; this cluster seems to contain some additional partial strings such as "*due to streptococcus group a*" even though such strings were not among the entries in result set;
- the top cluster which seems to be centered on the phrase *support group*, and
- the bottom cluster which contains tokens such as *reaction*, *anemia* and which appears to be responsible for ICD10 entries such as *reaction to blood-group incompatibility in infusion and transfusion, reaction due to rh factor in infusion*

and transfusion, and *iron deficiency anaemia secondary to blood loss (chronic)* in the results set.

The fact that the clusters are distinct indicates that it may actually be possible to objectively weed out less relevant items, besides restricting the recall bit radius.

To conclude, the associative properties of the DDAM model appear to be useful for context-modeling experiments and in languages where the degree of structure is relatively high, such as in the case of medical language. Though covering the whole spectrum of contextual information sources, from morphological to sentential and pragmatic contexts was not attained yet, a glimpse of this functionality was provided by the limited experiments. The full integration of information sources was already envisioned theoretically (Pantazi and Moehr 2004) and what was achieved here is a pragmatic step towards it. The integration remains an important goal which will be pursued in the future.

4.4 SUMMARY OF EXPERIMENTAL RESULTS

A summary of results of all experiments, categorized on the type of data (artificial or natural) is available in Table 85 and Table 86.

Experiment name	Data source	Task	Most significant DDAM performance	performance by comparison with DDAM model		
				ADIOS	SEQUITUR	SRN - Elman
Experiment #1	constant length strings generated from a 15 words lexicon	lexical acquisition	complete acquisition of lexicon as well as of many significant patterns	worse	slightly worse than DDAM but better than ADIOS	N/A
Experiment #2	variable length strings generated from a 30 words lexicon	lexical acquisition		worse	slightly worse than DDAM but better than ADIOS	N/A
Experiment #3	generated sequence	lexical acquisition		N/A	N/A	comparable performance for lexical acquisition but no significant pattern acquisition
Context dependent grammar induction	500 sentence corpus generated from a context free grammar	terminal symbol equivalence set acquisition	complete acquisition of all terminal symbol equivalence sets	slightly worse performance, compact grammar, very slow algorithm	slightly worse performance	N/A

Table 85. Summary of results from experiments with artificial sequences

The summary of results from experiments with artificial sequences in Table 85 shows clearly that the DDAM model is superior to existing approaches whose results are either worse or at best, slightly worse or comparable. In many instances, the pattern acquisition superiority of DDAM could be attributed to the increased computational complexity of the algorithm and to the apparent robustness to the local minima problem that seems to be affecting other models (e.g., SEQUITUR). The results also demonstrate how performance may actually be impaired by the sophistication of some approaches which are driven by a set of parameters whose semantics are unclear and which require a careful tune-up in order to achieve desired as opposed to unpredictable results (e.g., ADIOS).

Experiment name	Data source	Task	Most significant DDAM performance	performance by comparison with DDAM model		
				ADIOS	SEQUITUR	unsupervised MBDP-like models
Genomic sequence processing	SARS virus genome and proteome sequence	codon detection	moderate results	slightly worse performance	N/A	N/A
		protein pattern detection	moderate results	slightly worse performance	N/A	N/A
Automated lexical acquisition from text	WordNet glosses	lexical acquisition	51% precision	N/A	N/A	N/A
	Alice's Adventures in Wonderland	text segmentation	73%, 70% for per segment and 45%, 43% for per word precision and recall respectively	N/A	comparable performance, slightly better for per segment precision but worse for word segmentation	N/A
	CHILDES, Bernstein-Ratner87	text segmentation	79%, 85% for per segment and 61%, 65% for per word precision and recall respectively	N/A	N/A	better word segmentation performance
Grammar induction	Alice's Adventures in Wonderland	grammar induction	modest acquisition of morphological equivalence classes, interesting results for lexical class acquisition	N/A	N/A	N/A
Medical Natural Language Processing (NLP)	MedTest collection	text segmentation	71%, 54% per segment precision and recall respectively	N/A	N/A	N/A
		lexical equivalence set induction	acquisition of an extremely significant equivalence set	N/A	N/A	N/A
	ICD10 data	unsupervised morpho-segmentation of compound medical terms	60% precision	N/A	N/A	N/A
		similarity based retrieval	demonstrated similarity based retrieval capabilities and clustering; inherent mechanism for eliminating spurious results	N/A	N/A	N/A

Table 86. Summary of results from experiments with natural sequences

Unlike the previous, ascertaining the performance of the DDAM model from the summary of results of experiments with natural sequences shown in Table 86 was not straightforward due to a lack of clear evaluation criteria which is common for unsupervised approaches to processing natural data. However the DDAM model was shown to be comparable with existing approaches. The only instance where the model performed worse was the case of text segmentation, precisely the per-word segmentation task on a very specific type of data, i.e., the CHILDES data. This particular word segmentation task, though clearly not in the spirit of acquisition of patterns which are as significant as possible (i.e., as long as possible), was evaluated in order to enable comparison with existing, state of the art approaches (e.g., MBDP-like models). Yet, the

fact that the DDAM results were still competitive with those of MBDP-like models can only be a positive sign.

Finally, an indication of the degree of generality of the DDAM model is offered by the observation that the unchanged version of the very same information processing approach has demonstrated consistent performance across a wide range of experimental setups, data sources and processing tasks, unlike any of the other existing models considered in this dissertation and which appear specific to a certain type of task. Partly due to the difficulty to modify them and partly due to some fundamental limitations of existing alternative models, this appears in Table 86 in the form of the multitude of N/A (i.e., not available) which suggests the unavailability of means for performance comparison. The best example for such a task is that of similarity based retrieval for which none of the existing models reviewed in this dissertation offered a viable basis for comparison.

Chapter 5 Conclusions, Outlook, and Future Work

This chapter contains the conclusions, the contribution to knowledge, a to-do list of future work and a discussion of the possibilities for application of the proposed associative memory model to richer representations of information such as images, sounds and simulations.

5.1 SUMMARY AND CONCLUSIONS

This section contains a short summary the work and conclusions.

The "axiom of information systems" (i.e., the need to build systems that are both usable and useful at the same time) has led naturally to the fundamental problems of informatics, i.e., the problem of representation, as well as to a vision for its solution: creating information systems which have the capacity to acquire, with as high degree of autonomy as possible, useful, relatively complete, problem specific representation of complex problems that need to be solved. In this context, the most fundamental conclusion of this dissertation is that human-like, context-dependent representation of information is of extreme importance to applied sciences such as Medical Informatics.

Further, it has been demonstrated that the usefulness of context-independent, general knowledge is compromised by the "frame problem," a fundamental issue of artificial intelligence that underlines the extreme importance of *representing change*, through *dynamic approaches to representation*. It has also been shown that representing and processing context-dependent, individual context knowledge are prerequisites for case-based reasoning (CBR), i.e., the method of knowledge processing that aims at solving new problems based on the solutions of similar past problems (i.e., cases). It has also been shown that CBR requires rich, *high dimensional representations* of cases, as well as the *similarity-based organization* of representations to facilitate their similarity-based retrieval. As a result, the exploration of the nature and the fundamental properties of context-dependent representations has been recast and unified around the notion of *associative concept representation spaces* characterized by four fundamental properties: *high dimensionality*, *sparseness*, *dynamicity,* and, *similarity-based organization*.

In the first half of this dissertation, it has been shown that, in order to achieve context-dependent representation and processing in the most appropriate way for Medical Informatics applications, information processing models and approaches need to address specifically every one of the four fundamental properties of *associative concept spaces*.

Concretely, the solution proposed entailed a particular spatio-temporal complexity trade off which used spatial complexity in order to minimize temporal complexity of the proposed models. Throughout this dissertation it has also been assumed and implied that the difficulties to implement such approaches arise only from their inherent complexity and from the amount of memory and processing power they require.

The theoretical accounts and the thesis of this dissertation have been fully supported by theoretical and empirical work on the *model of deterministic dynamic associative memory* (DDAM). Concretely, the DDAM model has demonstrated excellent capabilities of context-dependent representation and processing of artificial sequences in unsupervised lexical acquisition and grammar induction tasks, equalling or surpassing the performance of existing models.

In experiments on natural sequences, though performance was affected by the increased complexity of data and by the lack of clear evaluation criteria which is common to unsupervised tasks, the DDAM model performed comparably well or better than existing unsupervised models in tasks such as DNA codon detection, protein pattern detection, lexical acquisition from text, word segmentation and grammar induction. Experiments specific to medical language processing have been more difficult to assess due to the complete lack of existing unsupervised information processing models for medical data and whose performance on identical tasks could have served as evaluation criterion. However, the DDAM model has shown unsupervised morpho-segmentation of compound medical terms and text segmentation capabilities with precisions in the range of 60-70%, which is comparable to earlier experiments as well as with published results for similar tasks but on general text. For the unsupervised lexical equivalence set induction of medical terminology, a form of grammar induction applied to medical text, the DDAM model has shown interesting results that demonstrate the potential to apply it to unsupervised categorization tasks. Despite the lack of clear evaluation criteria, the usefulness of the DDAM model was demonstrated by the discovery and acquisition of an extraordinarily long pattern which has determined an equivalence class of significant

importance, a feat that current technology and even human processors are largely incapable of.

Finally, the DDAM model has demonstrated, without doubt, advanced capabilities for the difficult task of dynamic, deterministic, completely unsupervised similarity based retrieval and clustering of variable length sequences. Though still limited and part of ongoing research, these capabilities appear to be currently unmatched by alternative models that share the deterministic and dynamic nature of the DDAM model. In this context, and perhaps most importantly, it was demonstrated empirically how the amount of spurious matches that affect the similarity retrieval capabilities of the DDAM model can be dynamically reduced by simply providing additional data to the model which, as a consequence, requires only additional memory and computing power. This is first hand evidence for the major assumption in this dissertation namely that difficulties to implement the proposed technology arise solely from the inherent complexity and from the amount of memory and processing power required.

5.2 CONTRIBUTION TO KNOWLEDGE

This section summarizes the contribution to knowledge is which is threefold: methodological, theoretical (unifying traditionally distinct theories and concepts) and practical (the proposal, implementation and evaluation of the deterministic model of dynamic associative memory). It also comprises a list of publications that contain some of the material included in this dissertation.

The contribution to knowledge of this dissertation is threefold.

First, a *research methodology* based on *prototyping* was proposed and used extensively in order to pursue the theoretical and empirical work on the DDAM model.

Second, the contribution comprises theoretical accounts which unify theories and research topics which, traditionally, have formed the object of distinct fields of research. The introduction of the *knowledge spectrum framework* has allowed the clear distinction between *general knowledge* and *individual context knowledge* as well as their logical and conceptual connection with the *frame problem* and *case based reasoning*. The strong conceptual relationships between case-based reasoning, individual context knowledge processing, medical decision making, algorithmic information theory, information retrieval, have been clarified and unified around the theme of context-dependent information processing. From this perspective, a new *definition of the field of Medical Informatics* was proposed as "context-dependent medical information processing." Also on the theoretical side, an alternative approach to information processing models was proposed in the form of the *trade-off of space complexity for time complexity* (memory-based processing). This has allowed a generalization over existing information processing models (e.g., trie memory models) and led to the proposal of an alternative to minimum description length (MDL), namely *minimum description work* (MDW) that could cater to memory-based information processing models which are slow but possess plenty of memory.

Third, the contribution to knowledge has materialized in the proposal, implementation and evaluation of the new model of deterministic, dynamic associative memory (DDAM)

which emerges naturally from the theoretical findings. The proposed model generalizes existing n-gram and Markov model approaches and is based on the theoretically sound and elegant mathematical models of unconstrained substring poset and constrained substring poset which connect, extend and generalize existing mathematical concepts such as the Dyck path and the combinatorial integer composition. The model also proposes two novel algorithms for string composition and decomposition whose evaluation in grammar induction, general sequence processing and natural language processing speak for their usefulness. In this context, a novel possible mechanism for inventiveness and for CBR case adaptation was proposed as a new avenue for research.

Finally the DDAM model has allowed for an unequalled breadth of unsupervised information processing experiments which culminated with empirically demonstrated capabilities for similarity based retrieval. This offered a glimpse of the processing possibilities of the DDAM model, in particular of the important information processing function of similarity based retrieval which forms the object of future research.

Some of the material in this dissertation has been published previously:

- Pantazi SV, Kushniruk A, Moehr JR **The Usability Axiom of Medical Information Systems**, International Journal of Medical Informatics (accepted) (peer reviewed)

- Pantazi SV, Moehr JR. **An Associative Memory Model For Unsupervised Sequence Processing**, In Proceedings of PacRim2005 (p. 233-236), Victoria, Canada, August 2005 (peer reviewed)

- Pantazi, S.V., J.F. Arocha, and J.R. Moehr, **Case-based Medical Informatics**, in Intelligent Paradigms in Healthcare Enterprises, B.G. Silverman, et al., Editors. 2005, Springer-Verlag: Berlin, Heidelberg (p. 31-65)

- Pantazi SV, Arocha JF, Moehr JR. **Case-based medical informatics**. BMC Journal of Medical Informatics and Decision Making; 2004, 4:19 (peer reviewed)

- Pantazi SV, Moehr JR., **Automated Knowledge Acquisition by Inductive Generalization**, e-Health 2004; Victoria, Canada.

- Pantazi SV, Kagolovsky Y, Moehr JR. **Cluster analysis of Wisconsin breast cancer dataset using self-organizing maps.** In proceedings of MIE'2002: The XVIIth International Congress of the European Federation for Medical Informatics (p. 431-436), Budapest, August 2002 (peer reviewed)

5.3 OUTLOOK AND FUTURE WORK

This section contains a list of to do items as well as some ideas which are highly speculative and far reaching.

In addition to the accomplishments achieved so far, in a limited period of time, there are multiple significant avenues opened by this research and that could complement the work. Some of these are:

- The appropriate review and comparison of the DDAM model with other existing information processing models and existing search algorithms (e.g., Viterbi search, dynamic programming),
- The development of a solid theoretical foundation of the DDAM model (e.g., tighter bounds on complexity, formal description of multi-layered models),
- The removal of some processing limitations of the DDAM model (e.g., a generalized definition of ambiguity that involves the frequencies of patterns),
- The application of the DDAM model to data compression in the MDL sense and comparison in compression benchmarks with state of the art compression approaches such as PPM,
- The pursuit of the possibilities to apply the DDAM model, particularly the mechanism to simulate "inventiveness," to case adaptation in CBR contexts,
- The development of a robust similarity based retrieval algorithm and its application in CBR contexts, particularly in the biomedical field,
- The improvement in design and implementation of a parallel and distributed version of the associative memory model based on recent advanced in grid computing, and

- The application of the models to new information processing tasks and real world problems such those from the bioinformatics area.

5.3.1 Advanced similarity based retrieval

The processing approaches described in this dissertation are general enough to accommodate many types of sequential information. The knowledge free approach of the DDAM model which has remained, throughout the experiments in this dissertation, largely uncommitted to any particular apriori knowledge about the input data and indicates clearly the possibility for similarity based retrieval applications which are little dependent on a particular language. Therefore the avenue whose exploration must continue is that of advanced medical information retrieval that works on conceptual principles and which shows a dependence on a particular language which remains as little as possible. The similarity based retrieval approach also has the potential to allow for human-like query dialogs such as "retrieve all patient descriptions which are 80% similar/relevant to the following description, in the following context/scenario" a capability of extreme importance in the context of the emerging Electronic Health Records.

So far, the dissertation has demonstrated the usefulness of the associative memory model only in limited processing tasks on general sequences, a necessary endeavor that allowed the comparison with existing models. Technical issues, such as finding alternatives to deal with separators in textual data and extending the models to make further use of the existing hierarchical structure of textual discourse, are matters that need to be addressed in the future. So far, what it was gained are only insights and glimpses of a better design and implementation of a solution to these important problems.

5.3.2 Music composition

It has been argued in the context of the constrained substring poset and associative memory model experiments that instilling controlled amounts of ambiguity into the model could lead to the creation of artefacts that are slight variations of the original

representations. It has also been argued that this could be regarded as a mechanism for the possible invention of new representations that bear compositional resemblance to existing ones. This insight leads naturally to a possible application where a multitude of existing sequences, such as musical scores, could be represented and, given the addition of controlled amounts of ambiguity, could be morphed into new compositions which remain recognizable due to their resemblance to known pieces but which, at the same time, are entirely new as a whole. This is an interesting application to follow up on, if not academically at least as a hobby, even though technical difficulties are expected from the need to representing melodies, chords and rhythms simultaneously and from the difficulties to capture and represent correlations between potentially distal elements in musical scores.

5.3.3 Speech processing and recognition

In the early days of this research, much of the thrust behind the development of the DDAM model consisted of the need to find a solution to the problem of speech recognition. The difficulty of the task has caused work on this problem to be temporarily postponed but never abandoned. Speech data is eminently sequential, albeit a special type of sequential data that seems to involve several parallel data streams. The DDAM model might have the potential to advance processing and recognition of speech data but it could also be extended beyond, to any data which is multidimensional in nature.

5.3.4 Multidimensional representations

From a dimensionality standpoint, the DDAM model proposed in this dissertation is limited to the processing of 1-dimensional informational objects (i.e., sequences). However, in order to be able to achieve this, the model itself had to be extended to an additional dimension. As a consequence, the DDAM representations of 1-dimensional sequences are necessarily 2-dimensional and exist in the form of Dyck paths with peaks and valleys that transcend the 1-dimensionality of the original data. Expectedly, the processing complexity of the extended representations has been estimated to be, in the

worst case, $O(n^2)$ in the length n of a sequence, i.e., increased by one order of magnitude, but almost linear in the case of highly sparse, natural sequences. The general rule that can be inferred here is that in order to process a representation of a specific dimensionality one has to devise models that transcend the dimensionality of the original data and create representations whose dimensionality and complexity are necessarily higher but could remain in the same order of complexity if data were sufficiently sparse. Consequently, there appears to be no obvious reason not to believe that the information processing models developed in this dissertation could be extended to processing representations whose dimensionality is higher than that of 1-dimensional sequences, by simply stepping up into a higher dimensional representation space. For example, in order to process 2-dimensional images, the models could be extended to allow 3-dimensional representations of images. By analogy with the *Dyck paths* for sequences, higher dimensional representations for images could be named *Dyck surfaces* and will expectedly have a processing complexity which would be $O(n^3)$ in the worst case but potentially only $O(n^2)$, if data were sufficiently sparse.

By the same token, there appears to be no obvious reason to believe that the information processing models developed in this dissertation could not be extended to multidimensional data signals, 3-dimensional objects and full-fledged 4-dimensional spatio-temporal simulations. Providing the additional increase in the order of the processing complexity can be overcome, the following question arises: would there ever be a need to extend the models beyond the representation of 4-dimensional spatio-temporal simulations or would this be just enough to allow for full-fledged artificial intelligence? If the answer were no, this could mean that such a status could be attainable by a processing complexity of "only" $O(n^5)$ in the worst case but probably lower in reality, due to the immense sparseness of 4-dimensional spatio-temporal natural data.

5.3.4.1 Summary

To summarize, the discussion whether information processing models introduced in this dissertation can be applied to multidimensional representations leads naturally to

inquiring into the feasibility of *signal processing* applications such as synthesis and recognition of human speech. Given the voice variability of multiple speakers, noise and other factors that so far have limited the success, these areas of application could benefit from approaches that are able to discover, represent, process patterns and integrate contextual information from different linguistic levels in order to achieve higher accuracy levels while remaining computationally feasible. The next logical step consists of the application of context-dependent processing to increasingly complex representations such as images and 3-dimensional, static and dynamic models, followed by the improvement of processing, segmentation and retrieval of such representations. Though currently the realm of machine vision research, these types of applications could be of immense interest to biomedical sciences where visual representations abound. Yet more importantly, if successful, the feasibility of such applications will prove that it would be possible to attain the convergence of context-dependent approaches of information processing into a unified model. Contingent to the realization of parallel and distributed, grid computing implementations and possibly VLSI (very large scale integration), on-chip solutions (e.g., such as (Murty, Reghu Raj et al. 2003)) that could allow efficient, hardware based multidimensional pattern discovery and processing, such a unified model has the potential to significantly advance the fields of medical informatics, robotics and artificial intelligence.

Bibliography

(1992). International Statistical Classification of Diseases and Related Health Problems. 10th Revision. Geneva, World Health Organization (WHO).

Aamodt, A. and E. Plaza (1994). "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches." AICom - Artificial Intelligence Communications 7(1): 39-59.

Adriaans, P. W. (1992). Language learning from a categorical perspective, University of Amsterdam. PhD.

Adriaans, P. W. and M. M. v. Zaanen (2004). "Computational Grammar Induction for Linguists." Grammars 7: 57-68.

Anwar, A. and S. Franklin (2003). "Sparse distributed memory for 'conscious' software agents." Cognitive Systems Research 4(4): 339-354.

Armengol, E., A. Palaudàries, et al. (2000). "Individual Prognosis of Diabetes Long-Term Risks: A CBR Approach." Methods of Information in Medicine Journal 5: 46-51.

Armengol, E. and E. Plaza (2003). "Relational Case-based Reasoning for Carcinogenic Activity Prediction." Artificial Intelligence Review 20(1-2): 121.

Baddeley, A. (2003). "Working memory and language: an overview." Journal of Communication Disorders 36(3): 189-208.

Balotta, C., S. Corvasce, et al. (2004). "SARS coronavirus AS, complete genome." Retrieved May 27, 2004, from http://www.ncbi.nlm.nih.gov/entrez/viewer.fcgi?val=AY427439.

Barlow, H. B. (1989). "Unsupervised learning." Neural Computation 1: 295-311.

Barzilay, R. and L. Lee (2002). Bootstrapping Lexical Choice via Multiple-Sequence Alignment. Conf on Empirical Methods in NLP.

Barzilay, R. and L. Lee (2003). Learning to Paraphrase: An Unsupervised Approach Using Multiple-Sequence Alignment. Proceedings of HLT-NAACL, Edmonton, Canada.

Bassi, A. (1999). An associative semantic model for text processing. String Processing and Information Retrieval Symposium, 1999 and International Workshop on Groupware.

Bassi, A. (2000). A dynamic associative semantic model for natural language processing based on a spreading activation network. Computer Science Society, 2000. SCCC '00. Proceedings. XX International Conference of the Chilean.

Batchelder, E. (2002). "Bootstrapping the lexicon: A computational model of infant speech segmentation." Cognition 83: 167-206.

Batchelder, E. O. (1998). Can a Computer Really Model Cognition? A study of Six Computational Models of Infant Word Discovery. Proceedings of the 20th Annual Conference of the Cognitive Science Society, University of Wisconsin-Madison, Lawrence Erlbaum Assocs.

Battista, G. D., P. Eades, et al. (1994). Algorithms for Drawing Graphs: an Annotated Bibliography, ftp://wilma.cs.brown.edu/pub/papers/compgeo/gdbiblio.ps.Z: 44.

Baud, R. H., C. Lovis, et al. (2001). Conceptual Search in Electronic Patient Record. MEDINFO 2001, London, UK, Amsterdam: IOS Press.

Baud, R. H., A.-M. Rassinoux, et al. (1999). The Power and Limits of a Rule-based Morpho-Semantic Parser. AMIA Annual Symposium. Proceedings., Philadelphia: Hanley & Belfus, Inc.

Becker, P. and J. H. Correia (2005). The ToscanaJ Suite for Implementing Conceptual Information Systems.

Belèn, D.-A., G. Pablo, et al. (2003). Adaptation Guided Retrieval Based on Formal Concept Analysis. Proceedings of ICCBR 03, Springer-Verlag.

Belèn, D.-A. and G.-C. Pedro (2001). Classification Based Retrieval Using Formal Concept Analysis. Proceedings of ICCBR 01, Springer-Verlag.

Bell, A. J. and T. J. Sejnowski (1997). "The `Independent Components' of Natural Scenes are Edge Filters." Vision Research 37(23): 3327--3338.

Bernstein-Ratner, N. (1987). The phonology of parent child speech. Children's Language. K. Nelson and A. vanKleeck. Hillsdale, NJ., Lawrence Erlbaum Associates. 6.

Bichindaritz, I. (2005). Memory Organization As the Missing Link Between Case Based Reasoning and Information Retrieval in Biomedicine. ICCBR-05 Workshop on CBR in the Health Sciences.

Blake, C. and C. Merz. "UCI Repository of machine learning databases." Retrieved July 23, 2004, from http://www.ics.uci.edu/~mlearn/MLRepository.html.

Blois, M. S. (1984). Information and medicine: the nature of medical descriptions. Berkeley, University of California Press.

Bolle, D., D. R. C. Dominguez, et al. (2000). "Mutual information of sparsely coded associative memory with self-control and ternary neurons." Neural Networks 13(4-5): 455-462.

Bosch, A. v. d. and W. Daelemans (1998). Do not Forget: Full Memory in Memory-Based Learning of Word Pronunciation. Proceeding of NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning, Sydney, Australia, Association of Computational Linguistics.

Bourtchouladze, R. (2002). How Many Memory Systems Are There? Memories are made of this: how memory works in humans and animals. S. Rose. New York, Columbia University Press: viii, 199.

Bourtchouladze, R. (2002). Wax, Theatres and Nonsense Syllables: A Brief Overview of the History of Memory. Memories are made of this: how memory works in humans and animals. S. Rose. New York, Columbia University Press: viii, 199.

Brants, T., F. Chen, et al. (2002). Topic-based document segmentation with probabilistic latent semantic analysis. Proceedings of the eleventh international conference on Information and knowledge management, McLean, Virginia, USA, ACM Press.

Brent, M. R. (1999). "An efficient, probabilistically sound algorithm for segmentation and word discovery." Machine Learning Journal 34: 71-106.

Brent, M. R. and T. A. Cartwright (1996). "Distributional regularity and phonological constraints are useful for segmentation." Cognition 61: 93□125.

Brent, M. R. and X. Tao (2001). Chinese text segmentation with MDPB-1: Making the most of training corpora. Proceedings of the Annual Meeting of the Association for Computational Linguistics, Hong Kong.

Brusilovsky, P. and M. T. Maybury (2002). "From adaptive hypermedia to the adaptive Web." Communications of the ACM 45(5): 31-33.

Carpineto, C. and G. Romano (2005). Using Concept Lattices for Text Retrieval and Mining.

Carroll, L. (2005). ALICE'S ADVENTURES IN WONDERLAND, http://www.gutenberg.org/etext/11.

Cerveri, P., M. Masseroli, et al. (2000). Remote access to anatomical information: an integration between semantic knowledge and visual data. Proc. AMIA Symp.

Chaitin, G. J. (1970). "To a mathematical definition of "life"." ACM SICACT News 4: 12-18.

Chaitin, G. J. (1974). "Information-Theoretic Computational Complexity." IEEE Transactions on Information Theory 20: 10-15.

Chaitin, G. J. (1975). "Randomness and Mathematical Proof." Scientific American 232(No. 5 (May 1975)): 47-52.

Chaitin, G. J. (1982). "Gödel's Theorem and Information." International Journal of Theoretical Physics 22: 941-954.

Christiansen, M. H., J. Allen, et al. (1998). "Learning to segment speech using multiplue cues: A connectionist model." Language and Cognitive Processes 13: 221□268.

Cigarran, J. M., A. Peitas, et al. (2005). Automatic Selection of Noun Phrases as Document Descriptors in an FCA-Based Information Retrieval System.

Cimino, J. J. (1998). "Desiderata for controlled medical vocabularies for twenty-first century." Methods Inf. Med. 37: 394-403.

Clark, A. S. (2001). Unsupervised Language Acquisition: Theory and Practice, University of Sussex. PhD.

Cleary, J. G. and W. J. Teahan (1998). "Unbounded length contexts for PPM." The Computer Journal 36(5): 1-9.

Cole, R. and P. Becker (2005). Navigation Spaces for the Conceptual Analysis of Software Structure.

Cormen, T. H., C. E. Leiserson, et al. (2001). Introduction to Algorithms. Cambridge, Massachusetts, The MIT Press.

Creutz, M. and K. Lagus (2002). Unsupervised discovery of morphemes. Workshop on Morphological and Phonological Learning of ACL'02, Philadelphia, Pennsylvania, USA.

Daelemans, W. (1998). Abstraction is Harmful in Language Learning. Proceedings of NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning, Sydney, Australia, Association of Computational Linguistics.

Dennett, D. (1984). Cognitive Wheels: The Frame Problem in AI. Minds, Machines, and Evolution. C. Hookway, Cambridge University Press: 128-151.

Deutsch, E. (1998). "A bijection on Dyck paths and its consequences." Discrete Mathematics 179: 253-256.

Dittenbach, M., A. Rauber, et al. (2002). "Uncovering hierarchical structure in data using the growing hierarchical self-organizing map." Neurocomputing 48(1-4): 199-216.

Ducrou, J. and P. Eklund (2005). Combining Spatial and Lattice-Based Information Landscapes.

Eades, P. and M. L. Huang (2000). "Navigating Clustered Graphs using Force-Directed Methods." Journal of Graph Algorithms and Applications 4(3): 157-181.

Edelman, S., Z. Solan, et al. (2005). Learning Syntactic Constructions from Raw Corpora. 29th Boston University Conference on Language Development, Cascadilla Press.

Eklund, P. and B. Wormuth (2005). Restructuring Help Systems Using Formal Concept Analysis.

Ellard, D. and P. Ellard. (2003). "S-Q Course Book." Retrieved Jan 24, 2006, from http://www.eecs.harvard.edu/~ellard/Courses/.

Elman, J. L. (1990). "Finding Structure in Time." Cognitive Science 14(2): 179-211.

Engel, K. (1997). Sperner theory. Cambridge; New York, Cambridge University Press.

Erdos, P. L., P. Sziklai, et al. (2001). "A finite word poset." The Electronic Journal of Combinatorics 8(2): 1-10.

Fan, K.-C. and Y.-K. Wang (1997). "A genetic sparse distributed memory approach to the application of handwritten character recognition." Pattern Recognition 30(12): 2015-2022.

Federici, D. (2003). Implicant network: an associative memory model. Neural Networks, 2003. Proceedings of the International Joint Conference on.

Feigenbaum, E. A. (2003). "Some challenges and grand challenges for computational intelligence." Journal of the ACM (JACM) 50(1): 32-40.

Field, D. J. (1994). "What is the goal of sensory coding?" Neural Computation 6: 559-601.

Fierz, W. (2004). "Challenge of personalized health care:To what extent is medicine already individualized and what are the future trends?" Med Sci Monit 10(5): 111-123.

Fine, S., Y. Singer, et al. (1998). "The hierarchical hidden Markov Model: Analysis and applications." Machine Learning 32: 41.

Flachs, B. and M. Flynn (1994). Sparse adaptive memory and handwritten digit recognition.

Fodor, J. and Z. W. Pylyshyn (1988). "Connectionism and Cognitive Architecture: a Critical Analysis." Cognition 28: 3-71.

Foldiak, P. (1990). "Forming sparse representations by local anti-Hebbian learning." Biol. Cybern. 64: 165-170.

French, R. M. (2002). "The Computational Modeling of Analogy -Making." Trends in Cognitive Sciences 6(5): 200-205.

French, R. M. and C. Labiouse (2002). Four Problems with Extracting Human Semantics from Large Text Corpora. Proceedings of the 24th Annual Conference of the Cognitive Science Society, NJ.

Frick, A., A. Ludwig, et al. (1995). A Fast Adaptive Layout Algorithm for Undirected Graphs. DIMACS Workshop on Graph Drawing, Springer Verlag.

Friedrich, C. and P. Eades (2002). "Graph Drawing in Motion." Journal of Graph Algorithms and Applications 6(3): 353–370.

Fritsche, L., A. Schlaefer, et al. (2002). "Recognition of Critical Situations from Time Series of Laboratory Results by Case-Based Reasoning." J Am Med Inform Assoc 9(5): 520-528.

Giancoli, D. C. (1980). Physics. Englewood Cliffs, NJ, Prentice-Hall, Inc., 1980.

Godin, R. and P. Valtchev (2005). Formal Concept Analysis-Based Class Hierarchy Design in Object-Oriented Software Development.

Gold, M. E. (1967). "Language identification in the limit." Information and Control 10: 447-474.

Goldsmith, J. (2001). "Unsupervised Learning of the Morphology of a Natural Language." Computational Linguistics 27(2): 153-198.

Gough, M. P. (1997). "Associative List Memory." 10(6): 1117.

Grant, A., A. Kushniruk, et al. (2004). An informatics perspective on decision support and the process of decision-making in health care. Using Knowledge and Evidence in Health Care. L. Lemieux-Charles and F. Champagne. Toronto, University of Toronto Press.

Greene, R. L. (1994). "Efficient retrieval from sparse associative memory." 66(2): 395.

Greene, R. L. and A. A. Tussing (2001). "Similarity and Associative Recognition." Journal of Memory and Language 45(4): 573-584.

Greene, W., C.-y. Hsu, et al. (1996). "Case Records of the Massachusetts General Hospital: A Home-Court Advantage?" N Engl J Med 334(3): 197-198.

Grimaldi, R. P. (2004). Discrete and Combinatorial Mathematics: An Applied Introduction. Boston, San Francisco, New York, Addison Wesley.

Guha, R. and D. Lenat (1990). "Cyc: A Midterm Report." AI Magazine 11(3): 32-59.

Guyon, I. and F. Pereira (1995). Design of a linguistic post processor using Variable Memory Length Markov Models. Proc. 3rd Int'l Conf. Document Analysis and Recogition, Montreal, Canada.

Hadley, R. F. (1994). "Systematicity in connectionist language learning." Mind and Language(9): 247-272.

Hadley, R. F., A. Rotaru-Varga, et al. (2001). "Syntactic systematicity arising from semantic predictions in a Hebbian-competitive network." Connection Science 13: 73-94.

Hahn, U., M. Honeck, et al. (2001). Subword Segmentation - Levelling out Morphological Varieties for Medical Document Retrieval. Proceedings of the AMIA Annual Symposium.

Hamalainen, T., H. Klapuri, et al. (1997). "Parallel realizations of Kanerva□s sparse distributed memory on a tree-shaped computer." CONCURRENCY: PRACTICE AND EXPERIENCE 9(9): 877-896.

Hammerton, J. (2003). Learning to segment speech with self-organising maps. Computational Linguistics in the Netherlands 2002. Selected Papers from the Thirteenth CLIN Meeting. T. GAUSTAD. Amsterdam/New York, NY. VII: 159.

Harris, Z. S. (1955). "From phoneme to morpheme." Language 31(2): 190-222.

Hart, E. and P. Ross (2003). "Exploiting the Analogy between the Immune System and Sparse Distributed Memories." Genetic Programming and Evolvable Machines 4: 333-358.

Haykin, S. S. (1994). Neural networks: a comprehensive foundation. New York; Toronto, Macmillan.

Hely, T. A., D. J. Willshaw, et al. (1999). "A New Approach to Kanerva's Sparse Distributed Memory." IEEE TRANSACTIONS ON NEURAL NETWORKS XX(Y): 101-105.

Hersh, W. R., D. H. Hickam, et al. (1994). "A performance and failure analysis with a MEDLINE test collection." J. Am. Med. Inform. Assoc. 1: 51-60.

Hesse, W. and T. Tilley (2005). Formal Concept Analysis Used for Software Analysis and Modelling.

Hewitt, P. G. (1987). Conceptual Physics, Addison-Wesley Publishing Company, Inc.

Hirahara, M., N. Oka, et al. (2000). "A cascade associative memory model with a hierarchical memory structure." Neural Networks 13(1): 41-50.

Hirahara, M., N. Oka, et al. (2000). "Cascade associative memory storing hierarchically correlated patterns with various correlations." Neural Networks 13(1): 51-61.

Hodge, V. J. and J. Austin (2002). "Hierarchical word clustering -- automatic thesaurus generation." Neurocomputing 48(1-4): 819-846.

Hofmann, T. (2001). "Unsupervised learning by probabilistic latent semantic analysis." MACHINE LEARNING 42: 177-196.

Holyoak, K. J. and P. Thagard (1995). Mental leaps: analogy in creative thought. Cambridge, Mass., MIT Press.

Honigwachs, J. (2006). "About BIG NUMBERS (http://pages.prodigy.net/jhonig/bignum/qauniver.html)." Retrieved Feb 11, 2006.

Hopkins, M. W. (1999, 15 Feb 23:32:05 -0500). "Re: How to extract grammar from a program?" Retrieved Dec 6, 2005, from http://compilers.iecc.com/comparch/article/99-02-077.

Hutchens, J. L. (1994). Natural language grammatical inference, University of Western Australia. PhD.

Janlert, L.-E. (1987). Modeling Change - The Frame Problem. The Robot's dilemma: the frame problem in artificial intelligence. Z. W. Pylyshyn. Norwood, N.J., Ablex: xi, 156.

Johnson-Laird, P. N. (1993). Human and machine thinking. Hillsdale, N.J., Lawrence Erlbaum Associates.

Joslyn, C. (2004). Poset Ontologies and Concept Lattices as Semantic Hierarchies.

Joslyn, C. A. and S. Mniszewski (2004). Combinatorial Approaches to Bio-Ontology Management with Large Partially Ordered Sets. SIAM Workshop on Combinatorial Scientific Computing (CSC04).

Kanerva, P. (1988). Sparse distributed memory. Cambridge, Mass., MIT Press.

Kanerva, P. (1993). Sparse Distributed Memory and Related Models. Associative Neural Memories: Theory and Implementation. M. H. Hassoun. New York: Oxford University Press: 50-76.

Kanerva, P. (1997). Fully distributed representation. Proceedings of 1997 Real World Computing Symposium, Tokyo, Japan.

Kaski, S. (1997). Data Exploration Using Self-Organizing Maps. Neural Networks Research Centre. Helsinki, Helsinki University of Technology: 57.

Keane, M. T. (1988). Analogical problem solving. Chichester, West Sussex, England New York, E. Horwood; Halsted Press.

Kintsch, W. (2001). "Predication." Cognitive Science 25(2): 173-202.

Kintsch, W. (2002). "The potential of latent semantic analysis for machine grading of clinical case summaries." Journal of Biomedical Informatics 35(1): 3-7.

Kit, C. (2000). Unsupervised Lexical Learning as Inductive Inference, University of Shefield.

Kit, C. (2005). Unsupervised Lexical Learning As Inductive Inference via Compression. Language Acquisition, Change and Emergence. J. W. Minett and W. S. Y. Wang. CityU of HK Press: 251-296.

Klein, G. A. (1993). Decision making in action: models and methods. Norwood, N.J., Ablex Pub.

Klein, G. A. (1993). A Recognition-Primed Decision (RPD) Model of Rapid Decision Making. Decision making in action: models and methods. Norwood, N.J., Ablex Pub.: 138-148.

Klein, G. A. (1999). Sources of power: how people make decisions. Cambridge, Mass.; London, MIT Press.

Knuth, D. E. (1997). The art of computer programming: Sorting and Searching. Reading, Mass., Addison-Wesley.

Knuth, D. E. (1997). Retrieval on Secondary Keys. The art of computer programming: Sorting and Searching. Reading, Mass., Addison-Wesley. 3: 392-559.

Koberle, R. (1989). "Neural networks as content addressable memories and learning machines." Computer Physics Communications 56(1): 43-50.

Koestler, A. (1967). The Ghost in the Machine. London, Pan Books LTD.

Kohonen, T. (2001). Self-Organizing Maps. Berlin Heidelberg New York, Springer-Verlag.

Kolodner, J. (1993). Case-Based Reasoning. San Mateo, CA, Morgan Kaufmann Publishers.

Koppel, R., J. P. Metlay, et al. (2005). "Role of Computerized Physician Order Entry Systems in Facilitating Medication Errors." JAMA 293(10): 1197-1203.

Kovac, C. (2003). "Computing in the Age of the Genome." The Computer Journal 46(6): 593-597.

Krikelis, A. and C. C. Weems (1994). "Associative processing and processors." Computer 27(11): 12-17.

Kuncel, N. R., S. A. Hezlett, et al. (2001). "A Comprehensive Meta-Analysis of the Predictive Validity of the Graduate Record Examination's Implications for Graduate Student Selection and Performance." Psychological Bulletin 127(1): 162-181.

Lakhal, L. and G. Stumme (2005). Efficient Mining of Association Rules Based on Formal Concept Analysis.

Landauer, T. and S. Dumais (1997). "A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge." Psychological Review 104(2): 211-240.

Lenat, D. (1995). "CYC: A Large-scale Investment in Knowledge Infrastructure." Communications of the ACM 38(11): 33-38.

Lenat, D. B. and R. V. Guha (1989). Building large knowledge-based systems: representation and inference in the Cyc project. Reading, Mass., Addison-Wesley Pub. Co.

Li, M. and P. M. B. Vitányi (1997). An introduction to Kolmogorov complexity and its applications. New York, Springer.

Lovis, C., R. Baud, et al. (1997). Morphosemantems Decomposition and Semantic Representation to allow Fast and Efficient Natural Language Recognition. AMIA Annual Fall Symposium.

Luger, G. F. (2002). Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Addison-Wesley.

Ma, Q. and H. Isahara (2000). "Semantic networks represented by adaptive associative memories." Neurocomputing 34(1-4): 207-225.

Machler, M. and P. Buhlmann (2002). Variable Length Markov Chains: Methodology, computing and software. Seminar for Statistics Report 104, ETH Zurich.

Macura, R. T. and K. Macura (1997) "Case-based reasoning: opportunities and applications in health care (editorial)." Artificial Intelligence in Medicine Volume, 1-4 DOI:

MacWhinney, B. (2000). The CHILDES project: Tools for analyzing talk. Mahwah, NJ, Lawrence Erlbaum Associates.

Maher, M. L., M. B. Balachandran, et al. (1995). Case-Based Reasoning in Design, Lawrence Erlbaum Associates.

Maille, N., I. C. Statler, et al. (2005). An Application of FCA to the Analysis of Aeronautical Incidents.

Manevitz, L. M. and Y. Zemach (1997). "Assigning meaning to data: Using sparse distributed memory for multilevel cognitive tasks." Neurocomputing 14(1): 15-39.

Mangasarian, O. L. and W. H. Wolberg (1990). "Cancer diagnosis via linear programming." SIAM News 23(5): 1-18.

Manning, C. D. and H. Schütze (1999). Foundations of Statistical Natural Language Processing. Cambridge, Mass., MIT Press.

Marcken, C. G. d. (1996). Unsupervised Language Acquisition. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. PhD: 133.

Martin, A. and L. L. Chao (2001). "Semantic memory and the brain: structure and processes." Current Opinion in Neurobiology 11: 194–201.

McCray, A. T., A. M. Razi, et al. (1996). The UMLS knowledge source server: A versatile Internet-based research tool. JAMIA Symposium, Hanley and Belfus.

McElree, B., S. Foraker, et al. (2003). "Memory structures that subserve sentence comprehension." Journal of Memory and Language 48(1): 67-91.

Mcnamara, T. P. and V. A. Diwadkar (1996). "The Context of Memory Retrieval." Journal of Memory and Language 35(6): 877-892.

Miller, G. (1956). "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information." The Psychological Review 63: 81-97.

Miller, G. (1995). "WordNet: A Lexical Database for English." Communications of the ACM 38(11): 49-51.

Miller, R. A. (1994). "Medical diagnostic decision support systems--past, present, and future: a threaded bibliography and brief commentary." J Am Med Inform Assoc 1(1): 8-27.

Miller, R. A. and F. E. Masarie (1990). "The demise of the 'Greek Oracle' model for medical diagnostic systems." Methods of Information in Medicine 29: 1-2.

Moehr, J. R. (1989). "Teaching Medical Informatics - Teaching on the Seams of Disciplines, Cultures, Traditions." Medical Informatics and Education Special Issue, Meth. Inform. Med. 28: 273-280.

Moehr, J. R. (1994). "Health Informatics - A Scientific Challenge?" Meth. Inform. Med. 33: 250-253.

Moehr, J. R., F. J. Leven, et al. (1982). "Formal Education in Medical Informatics. - Review of Ten Years' Experience with a Specialized University Curriculum." Meth. Inform. Med. 21: 169-180.

Montani, S. and R. Bellazzi (1999). Integrating Case Based and Rule Based Reasoning in a Decision Support System: Evaluation with Simulated Patients. JAMIA Symposium supplement.

Montani, S., R. Bellazzi, et al. (1998). A CBR System for Diabetic Patient Theraphy. Proc. Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP 98), Wiley & Sons Publ.

Moor, G. J. E. D., S. Norager, et al. (2005). "The Role of the Grid in a Future Global Health Information Space." Methods of Information in Medicine 44(2): 137-139.

Murdock, B. Memory Models: Quantitative. International Encyclopedia of the Social & Behavioral Sciences.

Murdock, B., D. Smith, et al. (2001). "Judgments of Frequency and Recency in a Distributed Memory Model." Journal of Mathematical Psychology 45(4): 564-602.

Murdock, B. B. (1993). "Derivations for the Chunking Model." Journal of Mathematical Psychology 37(3): 421-445.

Murdock, B. B. (1993). "TODAM2: A Model for the Storage and Retrieval of Item, Associative, and Serial-Order Information." Psychological Review 100(2): 183-203.

Murphy, K. and M. Paskin (2001). Linear Time Inference in Hierarchical HMMs. Proceedings of Neural Information Processing Systems (NIPS2001).

Murphy, K. P. (2002). Dynamic Bayesian Networks: Representation, Inference and Learning, University of California, Berkeley. PhD.

Murphy, R. C. (2003). Phrase detection and the associative memory neural network. Neural Networks, 2003. Proceedings of the International Joint Conference on.

Murty, V. S., P. C. Reghu Raj, et al. (2003). Design of a high speed string matching co-processor for NLP. VLSI Design, 2003. Proceedings. 16th International Conference on.

Musen, M. A. (2002). "Medical informatics: searching for underlying components." Methods Inf Med 41(1): 12-19.

Negishi, M., D. Bullock, et al. (1999). A self-organizing two-stream model of language comprehension. Neural Networks, 1999. IJCNN '99. International Joint Conference on.

Nevill-Manning, C. G. (1996). Inferring Sequential Structure. Department of Computer Science, University of Waikato, New Zealand. Ph.D.

Nevill-Manning, C. G. and I. H. Witten (1997). "Compression and explanation using hierarchical grammars." Computer Journal 40(2-3): 103-116.

Nevill-Manning, C. G. and I. H. Witten (1997). "Identifying Hierarchical Structure in Sequences: A linear-time algorithm." Journal of Artificial Intelligence Research 7: 67-82.

Nomura, M., T. Aoyagi, et al. (2003). "Two-level hierarchy with sparsely and temporally coded patterns and its possible functional role in information processing." Neural Networks 16(7): 947-954.

Nordström, T. (1991). Sparse distributed memory simulation on REMAP3, Luleå University of Technology, Sweden.

Notredame, C. (2002). "Recent progress in multiple sequence alignment: a survey." Pharmacogenomics 3(1): 131-144.

Okada, M. (1996). "Notions of Associative Memory and Sparse Coding." 9(8): 1429.

Olivier, D. C. (1968). Stochastic grammars and language acquisition mechanisms, Harvard University. PhD.

Olshausen, B. A. and D. J. Field (1996). "Natural image statistics and eficient coding." Network: Computation in Neural Systems 7(2).

Pantazi, S., Y. Kagolovsky, et al. (2002). Cluster Analysis of Wisconsin Breast Cancer Dataset Using Self-Organizing Maps. Medical Informatics in Europe, Budapest, Hungary, IOS Press.

Pantazi, S. V., J. F. Arocha, et al. (2004). "Case-based Medical Informatics." BMC Journal of Medical Informatics and Decision Making 4(1).

Pantazi, S. V., A. Kushniruk, et al. (2006). "The usability axiom of medical information systems." International Journal of Medical Informatics (accepted).

Pantazi, S. V. and J. R. Moehr (2004). Automated Knowledge Acquisition by Inductive Generalization. e-Health 2004. Victoria, BC, Canada.

Pantazi, S. V. and J. R. Moehr (2005). An associative memory model for unsupervised sequence processing. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM2005), Victoria, BC, Canada.

Parker, R. and R. Miller (1989). "Creation of a Knowledge Base Adequate for Simulating Patient Cases: Adding Deep Knowledge to the INTERNIST-1/QMR Knowledge Base." Meth Inform Med 28: 346-351.

Patel, V. L., J. F. Arocha, et al. (1994). "The Psychology of Learning and Motivation: Advances in Research and Theory." 31: 187-252.

Patel, V. L., D. R. Kaufman, et al. (2002). "Emerging paradigms of cognition in medical decision-making." Journal of Biomedical Informatics 35(1): 52-75.

Pedersen, B., S. Edelman, et al. (2004). Some Tests of an Unsupervised Model of Language Acquisition. COLING-2004 Workshop on Psycho-computational Models of Human Language Acquisition, Geneva, Switzerland.

Pollack, J. B. (1990). "Recursive Distributed Representations." Artificial Intelligence 46(1): 77□105.

Pomi Brea, A. and E. Mizraji (1999). "Memories in context." Biosystems 50(3): 173-188.

Ponte, J. and W. Croft (1996). USeg: a retargetable word segmentation procedure for information retrieval. Symposium on document analysis and information retrieval (SDAIR '96).

Priss, U. (2005). Linguistic Applications of Formal Concept Analysis.

Pylyshyn, Z. W. (1987). The Robot's dilemma: the frame problem in artificial intelligence. Norwood, N.J., Ablex.

Rapp, P. E., I. D. Zimmerman, et al. (1994). "The algorithmic complexity of neural spike trains increases during focal seizures." J of Neuroscience 14(8): 4731-4739.

Rassinoux, A.-M., P. Ruch, et al. (2000). Semantic Handling of Medical Compound Words through Sound Analysis and Generation Processes. Proc AMIA Symp.

Rector, A., A. Rossi, et al. (1998). "Practical development of re-usable Terminologies: GALEN-IN-USE and the GALEN Organisation." Int J Med Inf 48(1-3): 71-84.

Rector, A. L. (1999). "Clinical terminology: why is it so hard?" Method Inf Med 38(4): 239-252.

Reimann, S. (1998). "On the design of artificial auto-associative neuronal networks." Neural Networks 11(4): 611-621.

Riesbeck, C. K. and J. L. Kolodner (1986). Experience, memory, and reasoning. Hillsdale, N.J., L. Erlbaum Associates.

Rissanen, J. (1978). "Modeling by shortest data description." Automatica 14: 465-471.

Rolls, E. T. and A. Treves (1990). "The relative advantages of sparse versus distributed encoding for associative neuronal networks in the brain." Network: Computation in Neural Systems(1): 407-421.

Rumelhart, D. E., G. E. Hinton, et al. (1986). Parallel Distributed Processing, MIT Press.

Saffran, E. M. (2000). "The Organization of Semantic Memory: In Support of a Distributed Model." Brain and Language 71(1): 204-212.

Salas, E. and G. A. Klein (2001). Linking expertise and naturalistic decision making. Mahwah, NJ, Lawrence Erlbaum Associates Publishers.

Sandberg, A., A. Lansner, et al. (2001). "Selective enhancement of recall through plasticity modulation in an autoassociative memory." Neurocomputing 38-40: 867-873.

Sandberg, A., J. Tegner, et al. (2003). "A working memory model based on fast Hebbian learning." Network: Computation in Neural Systems 14(4): 789-802.

Sanford, A. J. and P. Sturt (2002). "Depth of processing in language comprehension: not noticing the evidence." Trends in Cognitive Sciences 6(9): 382-386.

Schank, R. C. (1972). "Conceptual Dependency: A Theory of Natural Language Understanding." Cognitive Psychology 3: 552-631.

Schank, R. C. (1982). Dynamic memory: a theory of reminding and learning in computers and people. Cambridge [Cambridgeshire] New York, Cambridge University Press.

Schank, R. C. and R. P. Abelson (1977). Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures. Hillsdale, N.J., Erlbaum.

Schmidhuber, J. (1997). A Computer Scientist's View of Life, the Universe, and Everything. Foundations of Computer Science: Potential - Theory - Cognition. C. Freksa, M. Jantzen and R. Valk. Berlin, Springer. 1337: 201-208.

Schone, P. and D. Jurafsky (2000). Knowledge-Free Induction of Morphology Using Latent Semantic Analysis. Proceedings of the Fourth Conference on Computational Natural Language Learning and of the Second Learning Language in Logic Workshop, Lisbon, 2000. C. Cardie, W. Daelemans, C. Nedellec and E. T. K. Sang, Association for Computational Linguistics: 67-72.

Schone, P. J. (2001). Toward Knowledge-Free Induction of Machine-Readable Dictionaries. Department of Computer Science, University of Colorado. PhD: 269.

Schultz, S., M. Honeck, et al. (2002). Biomedical text retrieval in languages with a complex morphology. Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain, Association for Computational Linguistics.

Schulz, S. and U. Hahn (2000). "Morpheme-based, cross-lingual indexing for medical document retrieval." International Journal of Medical Informatics: 87-99.

Shannon, C. E. (1948). "A Mathematical Theory of Communication." The Bell System Technical Journal Vol. 27: pp. 379–423.

Shortliffe, E. H. and M. S. Blois (2001). The Computer Meets Medicine and Biology: Emergence of a Discipline. Medical Informatics: Computer Applications in Health Care and Biomedicine. E. H. Shortliffe, L. E. Perreault, G. Wiederhold and B. G. Buchanan, Springer Verlag.

Sloane, N. J. A. (accessed Oct 2005). "Sequences A000108/M1459 in "The On-Line Encyclopedia of Integer Sequences." (http://www.research.att.com/cgi-bin/access.cgi/as/njas/sequences/eisA.cgi?Anum=A000108)." Retrieved Oct 29, 2005, from http://www.research.att.com/cgi-bin/access.cgi/as/njas/sequences/eisA.cgi?Anum=A000108.

Snelting, G. (2005). Concept Lattices in Software Analysis.

Solan, Z., E. Ruppin, et al. (2003). Automatic acquisition and efficient representation of syntactic structures. Advances in Neural Information Processing, MIT Press. Cambridge, MA.

Solan, Z., D. Horn, et al. (2004). Unsupervised context sensitive language acquisition from a large corpus. Proc. 2003 Conf. on Neural Information Processing Systems (NIPS), MIT Press.

Solomon, W. D., A. Roberts, et al. (2000). Having our cake and eating it too: How the GALEN Intermediate Representation reconciles internal complexity with users requirements for appropriateness and simplicity. Proc. AMIA Symp 2000.

Solomonoff, R. J. (1964). "A Formal Theory of Inductive Inference." Information and Control 7(1): 1-22.

Solomonoff, R. J. (2003). "The Kolmogorov Lecture - The Universal Distribution and Machine Learning." The Computer Journal 46(6): 598-601.

Sommer, F. T. and G. Palm (1999). "Improved bidirectional retrieval of sparse patterns stored by Hebbian learning." 12(2): 281.

Sommer, F. T. and T. Wennekers (2001). "Associative memory in networks of spiking neurons." Neural Networks 14(6-7): 825-834.

Spackman, K., K. Campbell, et al. (1997). SNOMED RT: A Reference Terminology for Health Care. Proceedings of the 1997 AMIA Annual Fall Symposium, Philadelphia, Hanley & Belfus.

Stolcke, A. (1994). Bayesian Learning of Probabilistic Language Models, University of California at Berkeley. PhD.

Stolz, W. (1965). "A probabilistic procedure for grouping words into phrases." Language & Speech 8: 219-235.

Svenson, O. and A. J. Maule (1993). Time pressure and stress in human judgment and decision making. New York, Plenum Press.

Szolovits, P. (1982). Artificial intelligence in medicine. Boulder, CO, Westview Press.

Teahan, W. J. (1998). Modeling English Text. Department of Computer Science. Hamilton, The University of Waikato. PhD: 260.

Tilley, T., R. Cole, et al. (2005). A Survey of Formal Concept Analysis Support for Software Engineering Activities.

Timpf, S. and A. U. Frank (1997). Using hierarchical spatial data structures for hierarchical spatial reasoning. Spatial Information Theory - A Theoretical Basis for GIS (International Conference COSIT'97). S. C. Hirtle and A. U. Frank. Berlin-Heidelberg, Springer-Verlag. 1329: 69-83.

Valverde-Albacete, F. J. (2005). Explaining the Structure of FrameNet with Concept Lattices.

Van Zaanen, M. (2002). Bootstrapping Structure into Language: Alignment-Based Learning. Leeds, UK, University of Leeds. PhD.

Venkataraman, A. (2001). "A Statistical Model for Word Discovery in Transcribed Speech." Computational Linguistics 27(3): 352-372.

Ventos, V. r. and H. Soldano (2005). Alpha Galois Lattices: An Overview. Lecture Notes in Computer Science: 299.

Vervoort, M. (2000). Games, walks and Grammars, University of Amsterdam. PhD.

Vitányi, P. M. B. and M. Li (2000). "Minimum Description Length Induction, Bayesianism, and Kolmogorov Complexity." IEEE TRANSACTIONS ON INFORMATION THEORY 46(2).

Vogel, D. D. (1998). "Auto-associative memory produced by disinhibition in a sparsely connected network." Neural Networks 11(5): 897-908.

Wallace, C. S. and P. R. Freeman (1987). "Estimation and inference by compact coding." Journal of the Royal Statistical Society 49: 240-265.

Watson, I. and F. Marir (1994). "Case-Based Reasoning: A Review." The Knowledge Engineering Review 9(4): 355-381.

Weisstein, E. W. (2005). "From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/." Retrieved Oct 29, 2005.

Wichert, A. (2002). "Learning of associative prediction by experience." Neurocomputing 48(1-4): 741-762.

Wieland, W. (1975). Diagnose: Uberlegungen zur Medizintheorie. Berlin, New York, de Gruyter.

Wille, R. (1982). Restructuring lattice theory: an approach based on hierarchies of concepts. Ordered sets. I. Rival. Dordrecht-Boston, Reidel: 445-470.

Wille, R. (2005). Conceptual Knowledge Processing in the Field of Economics.

Wille, R. (2005). Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies.

Winston, P. H. (1984). Artificial intelligence. Reading, Mass., Addison-Wesley.

Wolff, G. (2004). REFERENCES FOR "HIERARCHICAL CHUNKING". R. J. Solomonoff, (personal communication).

Wolff, J. G. (1975). "An algorithm for the segmentation of an artificial language analogue." British J Psychology 66(1): 79-90.

Wolff, J. G. (1977). "The discovery of segments in natural language." British J Psychology 68: 97-106.

Wolff, J. G. (1980). "Language acquisition and the discovery of phrase structure." Language and Speech 23(3): 255-269.

Wolff, J. G. (1982). "Language acquisition, data compression and generalisation." Language & Communication 2(1): 57-89.

Wolff, J. G. (1988). Learning syntax and meanings through optimization and distributional analysis. Categories and Processes in Language Acquisition. Y. Levy, I. M. Schlesinger and M. D. S. B. Hillsdale. NJ, Lawrence Erlbaum.

Wolff, J. G. (2003). "Information Compression by Multiple Alignment, Unification and Search as a Unifying Principle in Computing and Cognition." Artificial Intelligence Review 19(3): 193.

Woods, W. A. (1997). Conceptual Indexing: A Better Way to Organize Knowledge. SML Technical Report Series. J. Treichel, Sun Microsystems, Inc.: 99.

Zakay, D. (1993). The impact of time perception processes on decision making under time stress. Time pressure and stress in human judgment and decision making. New York, Plenum Press: 59-69.

Zaki, M. J., N. Parimi, et al. (2005). Towards Generic Pattern Mining.

Zipf, G. K. (1949). Human behavior and the principle of least effort. Cambridge, MA, Addison-Wesley.

Zsambok, C. E. and G. A. Klein (1997). Naturalistic decision making. Mahwah, N.J., L. Erlbaum Associates.

Zuse, K. (1969). Rechnender Raum. English translation: "Calculating Space". Cambridge, Mass., Massachusetts Institute of Technology.

Appendices

Appendix 1. Basic notation and fundamental definitions

Some basic notation choices are listed in Table 87.

Notation	Meaning
$\times$	product of two integer numbers or Cartesian product of two sets
$\Rightarrow$	logical implication
$\in$	set membership
$\subseteq$	set inclusion
$\mathbb{N}$	set of natural numbers (positive integers including zero)
$\mathbb{Z}^+$	set of positive integer numbers (not including zero)
$\{\}$	the empty set
$n!$	*n* factorial, i.e., $n(n-1)(n-2)\ldots\times 3\times 2\times 1$
$\neq$	not equal to
$\wedge$	logical and (conjunction)

Table 87. Some basic notation used in this chapter

Set theory

Definition 1. (Grimaldi 2004) For the purpose of this dissertation, a *set* is considered a finite collection of objects in which the order is not important, and the *multiplicity* (i.e., to the existence of duplicate, or multiple instances of an element) is ignored. Members of a set are said to be elements of that set. For example the elements a, b and c are members of the set $S=\{a,b,c\}$, i.e., $a,b,c\in S$. $S=\{a,b,c\}$ is equivalent to the sets $\{b,c,a\}$ or $\{c,a,b\}$ (i.e., order is not important) as well as to $\{a,a,b,c\}$ or $\{b,b,a,a,a,c,c,c,c\}$ (i.e., multiplicity is ignored).

Definition 2. (Grimaldi 2004) The *cardinality* of a set S, denoted as $|S|$ is given by the number of its elements. For example, the sets $\{a,b,c\}$, $\{c,a,b\}$ as well as $\{a,a,b,c\}$ or $\{b,b,a,a,a,c,c,c,c\}$ have a cardinality of 3.

Definition 3. (Weisstein 2005) *Multisets* are generalizations of sets in which the multiplicity of elements matters and therefore duplicates or multiple instances of an element are considered distinct. For example, $\{a,b,c\}$, $\{a,a,b,c\}$ and $\{b,b,a,a,a,c,c,c,c\}$ are multisets which are not equivalent. In order to emphasize the

multiplicity, the multiple instances of the same element could be indexed. If, for example the index set is the set of positive integers, the multiset $\{b,b,a,a,a,c,c,c,c\}$ could be written as $\{b_1,b_2,a_1,a_2,a_3,c_1,c_2,c_3,c_4\}$.

Definition 4. The *cardinality* of a multiset M, denoted as $|M|$ is given by the number of distinct elements (including instances of the same element) in it. For example, the sets $\{a,b,c\}$, $\{c,a,b\}$ as well as $\{a,a,b,c\}$ or $\{b,b,a,a,a,c,c,c,c\}$ have cardinalities of 3, 4 and 9 respectively.

Definition 5. Two sets A and B are *isomorphic* if there exists a bijective function (or isomorphism) defined on A and taking values in B. For example, the sets $A=\{a,b,c\}$ and $B=\{d,e,f\}$ are isomorphic because there exists $g:A\to B$ such that $g(a)=f$, $g(b)=d$, and $g(c)=e$.

Binary relations

Definition 6. (Grimaldi 2004; Weisstein 2005) A *binary relation* on a set L, denoted $\prec$, is a set of pairs (x,y), where $x,y\in L$. Therefore $\prec$ is a subset of $L\times L$, i.e., $\prec\subseteq L\times L$. For example, if $L=\{a,ab,abc\}$ then

$$L\times L=\begin{pmatrix}(a,a) & (a,ab) & (a,abc)\\ (ab,a) & (ab,ab) & (b,abc)\\ (abc,a) & (c,ab) & (abc,abc)\end{pmatrix}$$ and $\prec$ could be the subset $\begin{pmatrix}\cdot & (a,ab) & \cdot\\ (ab,a) & \cdot & (b,abc)\\ \cdot & (c,ab) & \cdot\end{pmatrix}$, where the dots denote the missing elements.

Definition 7. (Grimaldi 2004) A binary relation $\prec$ on a set L is *reflexive* if $(x,x)\in\prec$, i.e., $x\prec x$, for all $x\in L$. For example, if $L=\{a,ab,abc\}$ then the set

$\{(a,a),(ab,ab),(abc,abc)\}$ must be included in $\prec$, or in other words, the subset of $L\times L$ that defines $\prec$ must contain the elements on the first diagonal of $L\times L$.

Definition 8. (Grimaldi 2004) A binary relation $\prec$ on a set L is *anti-symmetric* if $(x,y)\in\prec$ and $(y,x)\in\prec$ imply $x=y$, for all $x,y\in L$. In short, $\prec$ is anti-symmetric if $(x\prec y)\wedge(y\prec x)\Rightarrow x=y$, for all $x,y\in L$. By contrast, the binary relation $\prec$ would be *symmetric* if $(x,y)\in\prec$ implied $(y,x)\in\prec$, for all $x,y\in L$. In short, $\prec$ would be symmetric if $x\prec y\Rightarrow y\prec x$, for all $x,y\in L$.

For example, if $L=\{a,ab,abc\}$, a binary relation $\prec$ on L is anti-symmetric if it does not contain elements from both sides of the first diagonal of $L\times L$, such as (a,ab) and (ab,a), or (abc,ab) and (ab,abc), at the same time. A *symmetric binary relation*, on the other hand, for each element must also include their symmetric located the other side of the first diagonal.

Definition 9. (Grimaldi 2004) A binary relation $\prec$ on a set L is *transitive* if $(x,y)\in\prec$ and $(y,z)\in\prec$ implies $(x,z)\in\prec$, for all $x,y,z\in L$. In short, $\prec$ is transitive if $(x\prec y)\wedge(y\prec z)\Rightarrow x\prec z$, for all $x,y,z\in L$. For example, if $L=\{a,ab,abc\}$ and $\{(a,ab),(ab,abc)\}\subset\prec$ then $(a,abc)\in\prec$ as well.

Definition 10. (Grimaldi 2004) A binary relation $\prec$ on a set L is a *partial order* on L, if $\prec$ is reflexive, anti-symmetric and transitive. By contrast, if $\prec$ were reflexive, transitive but symmetric it would be an *equivalence relation* on L. For example, if $L=\{a,b,ab,abc\}$ then the subset

$\begin{pmatrix} (a,a) & \cdot & (a,ab) & (a,abc) \\ \cdot & (b,b) & (b,ab) & (b,abc) \\ \cdot & \cdot & (ab,ab) & (ab,abc) \\ \cdot & \cdot & \cdot & (abc,abc) \end{pmatrix}$ of $L \times L$ is a partial order on L while the subset

$\begin{pmatrix} (a,a) & \cdot & (a,ab) & \cdot \\ \cdot & (b,b) & (b,ab) & \cdot \\ (ab,a) & (ab,b) & (ab,ab) & (ab,abc) \\ \cdot & \cdot & (abc,ab) & (abc,abc) \end{pmatrix}$ is an equivalence relation on L.

Definition 11. (Grimaldi 2004; Weisstein 2005) A *cover relation* $\hat{\prec}$ on a set L is a binary relation on L and a transitive reflexive reduction of a partial order relation $\prec$ on L. An element $z \in L$ is said to cover an element $x \in L$, i.e., $x \hat{\prec} z$ if there is no $y \in L$, such that $x \prec y$ and $y \prec z$. In other words, an element cannot cover itself (the cover relation is not reflexive) and if $x \hat{\prec} y$ and $y \hat{\prec} z$ then x is not covered by z (the cover relation is not transitive), for all $x, y, z \in L$.

For example, the transitive reflexive reduction of a partial order $\begin{pmatrix} (a,a) & \cdot & (a,ab) & (a,abc) \\ \cdot & (b,b) & (b,ab) & (b,abc) \\ \cdot & \cdot & (ab,ab) & (ab,abc) \\ \cdot & \cdot & \cdot & (abc,abc) \end{pmatrix}$ on the set $L = \{a, b, ab, abc\}$ is

$\begin{pmatrix} \cdot & \cdot & (a,ab) & \cdot \\ \cdot & \cdot & (b,ab) & \cdot \\ \cdot & \cdot & \cdot & (ab,abc) \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$.

The set theory of strings

Definition 12. (Grimaldi 2004) An *alphabet* is a nonempty, finite set of symbols. For example $\Sigma = \{a, b, c\}$.

Definition 13. (Grimaldi 2004) A *list* (*sequence*, *n-tuple*) of length $n, n \in \mathbb{Z}^+$, is a finite, ordered set of elements denoted as $(x_1, x_2, \ldots, x_n)$ with $x_1, x_2, \ldots, x_n$ being elements of another set such as, for example, an alphabet.

Definition 14. (Grimaldi 2004) A *string* over an alphabet Σ is a list (or sequence, n-tuple) of elements (or characters) in Σ. For example, $x = (a, b, c)$ is a string over $\Sigma = \{a, b, c, d\}$. For brevity, strings are often represented without parentheses and commas between their elements (e.g., $x = abc$). The *empty string* denoted as λ, is a string with no elements.

Definition 15. (Grimaldi 2004) The number of elements in a string x gives its *length*, and is denoted as $\|x\|$. The string $x = abc$ has the length $\|x\| = 3$. The empty string has a zero length, i.e. $\|\lambda\| = 0$.

Definition 16. (Grimaldi 2004) The *set of all finite strings* over an alphabet Σ, denoted as Σ^* is the union of all strings over Σ, of any length and includes the empty string λ. When the empty string is not included, the union is denoted as Σ^+, i.e., the set of all strings of positive length. Σ^n is the set of all strings of length $n, n \in \mathbb{Z}^+$ and Σ^0 is the set $\{\lambda\}$.

Definition 17. A *language* L over an alphabet Σ is a subset of Σ^* i.e., $L \subseteq \Sigma^*$ (Grimaldi 2004). For the purpose of this dissertation, languages are always considered finite.

Definition 18. The *concatenation* of two strings $x, y \in \Sigma^+$, written as xy, is an operation closed on Σ^+ (i.e, $xy \in \Sigma^+$ for all $x, y \in \Sigma^+$). The concatenation of a string x with the empty string λ is x, i.e., $\lambda x = x$ (Grimaldi 2004). In addition $\|xy\| = \|x\| + \|y\|$ for all $x, y \in \Sigma^*$. For example, the concatenation of the strings $x_1 = (a, b)$ with $x_2 = (c, d)$ is (a, b, c, d) or *abcd* for short.

Definition 19. (Grimaldi 2004) A string $y \in \Sigma^*$ is a *substring* of $w \in \Sigma^*$ if there exist two strings $x, z \in \Sigma^*$ such that $w = xyz$. If $x \neq \lambda$ **or** $z \neq \lambda$, then y is a *proper substring* of w. The relation *is a proper substring of* is the reflexive reduction of *is a substring of* since a string cannot be a proper substring of itself. For example, a is a substring of both a and abc but is a proper substring of abc only. The empty string λ is a substring of any string including itself, and a proper substring of all nonempty strings.

Partial order sets (posets)

The definitions marked by black diamonds ♦ while not used explicitly in current models, are very often found in literature relevant to partial order sets. They are therefore included in order to provide a more complete picture of such mathematical objects and with the hope that the concepts they define will prove useful for the future developments of the theoretical underpinnings of the associative memory models presented in this chapter.

Definition 20. (Engel 1997; Grimaldi 2004; Weisstein 2005) A *poset* P is an ordered pair $P = (L, \prec)$ comprising a *base set* (or ground set) L and a partial order $\prec$ on L. If the base set L is finite then the poset is also *finite*. Posets can be depicted using Hasse diagrams (Grimaldi 2004; Weisstein 2005). The reflexivity and transitivity properties of $\prec$ would cause partial sets to contain many elements and their Hasse diagrams to be complicated. For example, reflexivity entails drawing the links from a node to itself, while transitivity entails showing links between all nodes that are pair-wise comparable (e.g., (a, abc), $(a, abcd)$, (b, abc), $(b, abcd)$, etc.). In order to avoid this and simplify diagrams such as those in Figure 62, the reflexivity and transitivity properties of the partial order are not depicted explicitly through directed edges. In effect, this simplified representation shows the string cover relations LPP and LPS defined previously as transitive reflexive reductions of the generic "*is substring of*" relation.

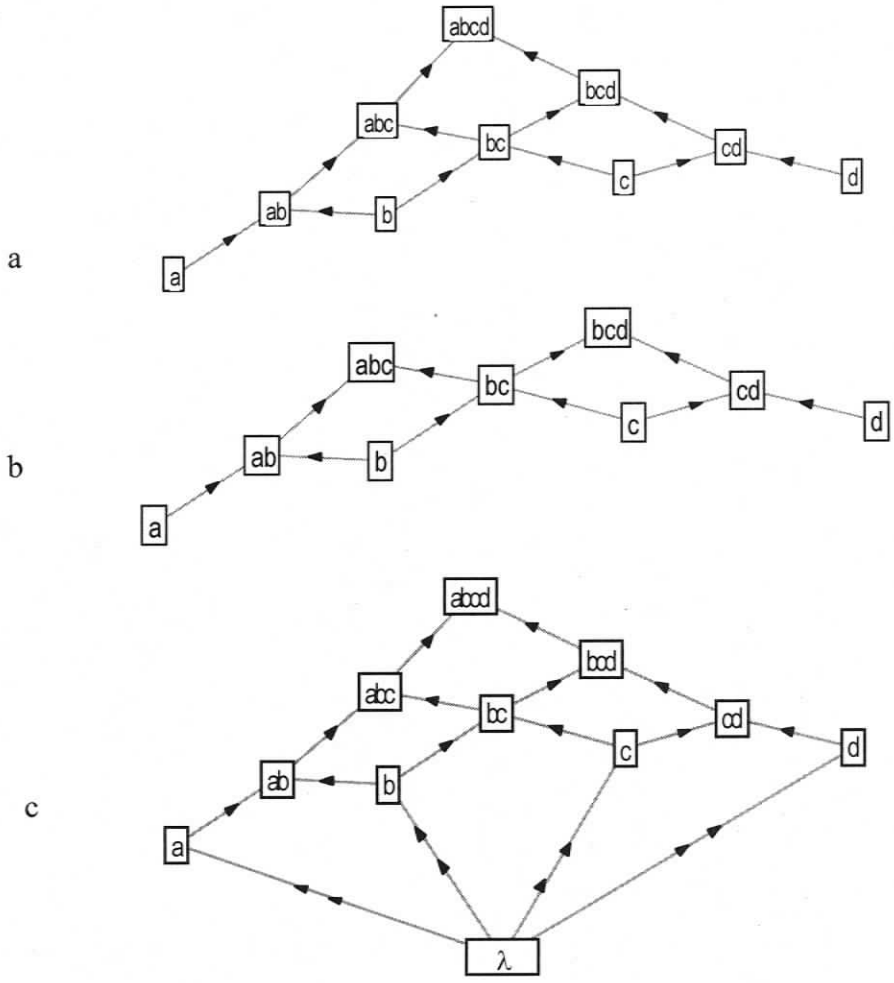


Figure 62. Hasse diagrams of partial ordered sets (posets) $(L,\prec)$; the partial order relations $\prec$ are the generic relation "is substring of" (e.g., $ab \prec abc$) depicted by directed edges as the LPP and LPS reductions; the double arrows in the last diagram suggest the fact that the edges from λ to a, b, c, d stand for both the LPP and LPS relations, at the same time

The general context of the following definitions is a poset $P=(L,\prec)$ with the partial order $\prec$ is given by the *is substring of* relation. The examples in the definitions are based on the diagrams in Figure 62.

Definition 21. (Engel 1997; Grimaldi 2004) An element $x \in L$ is a *maximal element* if, $x \prec y$ implies $x = y$ for all $y \in L$. For example, in Figure 62b, the elements *abc* and *bcd* are both maximal elements.

Definition 22. (Engel 1997; Grimaldi 2004) A maximal element $x \in L$ is a *greatest element* if $y \prec x$ for all $y \in L$. A greatest element, if exists, is always unique. For example, in Figure 1, *abcd* is the greatest element.

Definition 23. (Engel 1997; Grimaldi 2004) An element $x \in L$ is a *minimal element* if $y \prec x$ implies $y = x$, for all $y \in L$. For example, in Figure 62a and Figure 62b, the elements *a*, *b*, *c* and *d* are all minimal elements.

Definition 24. (Engel 1997; Grimaldi 2004) A minimal element $x \in L$ is a *least element* if $x \prec y$, for all $y \in L$. A least element, if exists, is always unique. The empty string λ in Figure 62c is a least element.

Definition 25. ♦(Engel 1997; Grimaldi 2004) An element $x \in L$ is an *upper bound* of a subset *S* of *L* if, for $w \prec x$ for every $w \in S$. For example, in Figure 62a, the elements ab, abc and abcd are all upper bounds of S={a,b,ab}.

Definition 26. ♦(Engel 1997; Grimaldi 2004) An element $x \in L$ is a *least upper bound* (LUB) for a subset *S* of *L* if it is a upper bound of *S* and if, for all other upper bounds *y* of *S* we have $x \prec y$. For example, $x = ab$ is a least upper bound for $S = \{a, b, ab\}$ and it is also included in *S*. In addition, any subset *S* of *L* can only have at most one LUB.

Definition 27. ♦(Engel 1997; Grimaldi 2004) An element $x \in L$ is a *lower bound* of subset *S* of *L* if $w \prec x$ for every $w \in S$. For example, in Figure 62c, the elements λ, *b* and *c* are all lower bounds of $S = \{bc, abc, bcd, abcd\}$.

Definition 28. ♦(Engel 1997; Grimaldi 2004) An element $x \in L$ is a *greatest lower bound* (GLB) for a subset *S* of *L* if it is a lower bound of *S* and if, for all other lower bounds *y* of *S* we have $y \prec x$. For example, $x = bc$ is a greatest lower bound for $S = \{abc, bcd, abcd\}$ and it is not included in *S*. In addition, any subset *S* of *L* can only have at most one GLB.

Definition 29. ♦(Engel 1997) A poset is *bounded* if it has both a least and a greatest element. For example, the poset in Figure 62c is bounded.

Definition 30. ♦(Engel 1997) A poset $P=(L,\prec)$ is a *lattice* if the subset $S=\{x,y\}$ has a LUB and a GLB which are included in L, for all $x,y\in L$.

Definition 31. ♦(Engel 1997) An *interval* for two elements $x,z\in L$, denoted $[x,z]$ is the set of all elements of L lying between x and z, that is $[x,z]=\{y\in L|x\prec y\prec z\}$. For example, the interval $[bc,abcd]=\{abc,bcd\}$.

Definition 32. (Engel 1997) A poset $(S,\prec)$ is a *totally ordered set* if either $x\prec y$ or $y\prec x$ for all $x,y\in S$, i.e., all its elements are pair wise comparable. For example, if $S=\{a,ab,abc\}$ and $\prec$ is the *is substring of* relation, then $(S,\prec)$ is a totally ordered set. Partial order sets are a generalization of total order sets.

Definition 33. (Engel 1997) Let S be a subset of L, $S\neq\{\ \}$. If the poset $(S,\prec)$ is a totally ordered set, then $(S,\prec)$ is a *chain* in the poset $(L,\prec)$. If the total order is given by the cover relation $\hat{\prec}$ (i.e. the transitive reflexive reduction) of $\prec$, then the chain $(S,\hat{\prec})$ is called *saturated*. If a saturated chain also contains a minimal and a maximal element, then that chain is called *maximal*. For example, in Figure 62a, $(\{b,bc,abcd\},\hat{\prec})$ is a chain and $(\{b,ab,abc,abcd\},\hat{\prec})$ is a maximal chain.

Definition 34. (Engel 1997) The *length of a partial ordered set P* is given by the cardinality of a maximal chain in *P*. For example, the length of the poset in Figure 62a is 4 (e.g., $S=\{b,bc,abc,abcd\}$, that of the poset in Figure 62b (e.g., $S=\{a,ab,abc\}$) is 3 and that of the poset in Figure 62c is 5 (e.g., $S=\{\lambda,c,bc,bcd,abcd\}$).

Definition 35. (Engel 1997) A poset $(S,\prec)$ is a *totally unordered set* if neither $x \prec y$ nor $y \prec x$ for all $x, y \in S$, i.e., all its elements are pair-wise non-comparable. For example, for any of the diagrams in Figure 62, if $S = \{a, b, cd\}$ then $(S,\prec)$ is a totally unordered set.

Definition 36. (Engel 1997) Let S be a subset of L, $S \neq \{\ \}$. If the poset $(S,\prec)$ is a totally unordered set, then $(S,\prec)$ is an *anti-chain* in the poset $(L,\prec)$. For example, $(\{a, bc, d\},\prec)$ and $(\{ab, bc, cd\},\prec)$ are both anti-chains.

Definition 37. (Engel 1997) The *width of a partial ordered set* P is given by the cardinality of the longest anti-chain in P. For example, the width of any of the posets in Figure 62 is 4 and is given by their longest anti-chain $\{a, b, c, d\}$.

Definition 38. (Engel 1997) A poset $P = (L,\prec)$ is *ranked* if there exists a *rank function* ϕ, $\phi : L \to \mathbb{N}$, such that $\phi(x) = 0$ for a minimal element $x \in L$ and $\phi(y) = \phi(z) - 1$ for all $y, z \in L, y \prec z$. The set of elements in L with the same rank n is called the nth level. A ranked, finite and bounded poset is called *graded*. The poset in Figure 62c is a graded poset having the rank function given by the length of each of its elements (e.g., $\phi(abc) = 3$).

Definition 39. ♦(Engel 1997) The *lower shadow* of an element $x \in L$ is the subset $\wedge(x)$ of L such that $y \prec x$, for all $y \in \wedge(x)$.

Definition 40. ♦(Engel 1997) The *upper shadow* of an element $x \in L$ is the subset $\vee(x)$ of L such that $x \prec y$, for all $y \in \vee(x)$.

Combinatorics

Definition 41. (Weisstein 2005) The nth central binomial coefficient is defined as:

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2}$$

Definition 42. (Weisstein 2005; Sloane accessed Oct 2005) *Catalan numbers* are integer sequences that appear in many situations, a few of them being:

- tree enumeration problems (Euler's polygon division problem);
- the number of binary bracketings of n letters (Catalan's problem);
- the number of extended binary trees with n internal nodes;
- the number of mountains which can be drawn with n upstrokes and n downstrokes;

The Catalan number C_n is defined as:

$$C_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{(n+1)(n!)^2} = \frac{(2n)!}{(n+1)!n!}$$

The first fifteen Catalan numbers are 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, 742900, 2674440, 9694845 (Sloane accessed Oct 2005).

Definition 43. (Weisstein 2005) An *integer partition* is one way of writing an integer as a sum of positive integers where the order of the addends is not significant. For example, the integer 4 can be written 1+2+1, 2+2, 1+3, etc. All integer partitions on an integer n correspond to the set of solutions $\{x_1, x_2, \ldots, x_n\}$ to the Diophantine equation (an equation for which solutions can only be integer values) $1x_1 + 2x_2 + \ldots + nx_n = n$, where $x_1, x_2, \ldots, x_n \in \mathbb{N}$.

Definition 44. (Weisstein 2005) A *combinatorial composition* of an integer n is an integer partition in which the order of the elements counts. For an integer n, there are 2^{n-1} combinatorial compositions. For example, for n=3 there are $2^2 = 4$ combinatorial

compositions (i.e., 1+1+1, 1+2, 2+1 and 3) and for n=4 there are $2^3 = 8$ combinatorial compositions (1+1+1+1, 1+1+2, 1+2+1, 2+1+1, 1+3, 3+1, 2+2 and 4).

Definition 45. (Deutsch 1998) A *Dyck path* (or excursion) of semilength n, $n \in \mathbb{Z}^+$, is a staircase-like walk in the XY plane from $x = (0,0)$ to $y = (2n,0)$ and consisting of steps $(1,1)$ (or North-East, called rises) and $(1,-1)$ (or South-East called falls) (Figure 63).

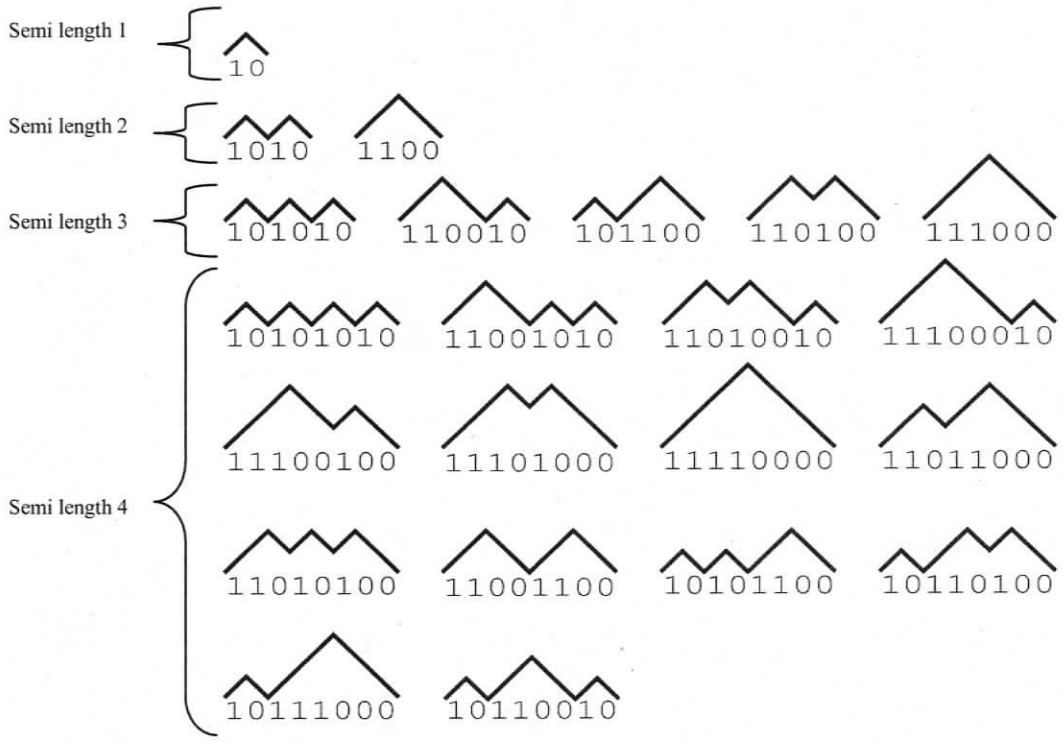


Figure 63. Dyck paths of semilength 1, 2, 3 and 4

Definition 46. (Deutsch 1998) A *Dyck word* is the encoding of a Dyck path that makes use of a pair of two symbols (e.g., u,d or $1,0$), one for the rise step and the other symbol for the fall step. For example, *udududud* or 10101010 are two equivalent Dyck words, which encode the first Dyck path of semilength 4 listed in Figure 63. Dyck words start always with a 1/*u* and end in a 0/*d* and contain always the same number of 1/*u*-s and 0/*d*-s.

In a Dyck path, a *peak* is the occurrence of the sequence *ud* in the corresponding Dyck word, while a *valley* is the occurrence of sequence *du*. An *ascent* of a Dyck path is a

maximal string of *u*'s together with its length while a *descent* is a maximal string of *d*'s together with its length.

The total number of Dyck paths of semilength *n* and is given by the Catalan number C_n. For example, there are $C_3 = 5$ possible Dyck paths of semilength 3:

$$C_3 = \frac{1}{4}\binom{6}{3} = \frac{6!}{4\times(3!)^2} = \frac{6\times5\times4\times3\times2\times1}{4\times(3\times2\times1)^2} = \frac{30}{6} = 5$$

Appendix 2. Delphi Pascal Code Listing

Naïve implementation of the DDAMv1 model

```
program cposet;
////////////////////
//    Stefan V. Pantazi, spantazi@uvic.ca
//    Aug 13, 2004
//    in order to use this code, also take a look at the following paper:
//    Pantazi, S.V. and J.R. Moehr. An associative memory model for unsupervised
//    sequence processing. In IEEE Pacific Rim Conference on Communications,
//    Computers and Signal Processing (PACRIM2005). 2005.
//    Victoria, BC, Canada. p. 233-236.
//
//    This code follows closely the formal description of the constrained poset
//    model found in the paper cited above.
//    However, it is a NAIVE implementation because:
//      - it uses dynamic arrays as main data structures,
//      - it builds the unconstrained poset first, and then it derives
//        the constrained poset from it
//          (a non-naive implementation will build the constrained poset gradually
//          in order to avoid the combinatorial (quadratic space requirements)
//          explosion of the unconstrained poset)
//    This implementation is limited in many ways:
//      - can only work with ONE sequence at a time
//      - does not implement recall functions
//      - etc.
//////////////////////
{$APPTYPE CONSOLE}

uses
  SysUtils,Math;
const
//the constraint parameters
  DELTA:Integer=3;//suggested DELTA>=1, WARNING delta<1 is dangerous
                  //due to combinatorial explosion!
  BETA:Integer=0;
type

  PTx=  ^Tx; //typed pointer to the pattern record
  Tx=record //the pattern record type
    w:            string;//pattern, substring
    c:            Integer;//count of occurrences of pattern in input
    OutputCount:   Integer;//count of binary relations with longer (upper) patterns
    Outputs:       array of PTx;//array with pointers to longer (upper) patterns
    InputCount:   Integer;//count of binary relations with shorter (lower) patterns
    Inputs:       array of PTx; //array with pointers to longer (lower) patterns
  end;

var
  s:      string;//the input sequence
  m:      Integer;//the length of the input sequence
  Mul:    array of array of string; //the "universe of discourse for our input",
                                    //i.e., the set of all nonempty, substrings
                                    //(words) in s, including s itself
                                    //it is a multiset because it has duplicates
                                    //(substrings are NOT unique)
  Xbase:   array of array of Tx;    //array of pattern records derived from Mul;
                                    //stores count information associated
                                    //with each pattern in Mul
                                    //it is the base set of the
                                    //unconstrained substring poset
//Mul and Xbase are bidimensional arrays
  i:      Integer;


function CheckConstraints(x1,x2:Tx):Boolean;
```

```
//evaluates a binary relation between two patterns
//returns true if constrains are satisfied
var
  m1,m2:  Integer;
  n1,n2:  Real;
begin
  Result:=False;
  if (x1.c>DELTA) and (x2.c>DELTA) then
  begin
    m1:=Length(x1.w);
    m2:=Length(x2.w);
    Assert(m2=m1+1,'Pattern 2 must be 1 char longer than pattern 1');
    n1:=m1*Log2(x1.c);
    n2:=m2*Log2(x2.c+BETA);
    if n2>=n1 then Result:=True;
  end;
end;

procedure Build_Mul;//this is a multiset as it has duplicate elements
//builds Mul from the input sequence
//Mul may contain duplicate patterns, hence it is a multiset
var
  i,j:  Integer;
begin
  for i:=0 to m-1 do
  begin
    Write('Mul['+IntToStr(i)+']=(');
    for j:=0 to m-i-1 do
    begin
      Mul[i][j]:=Copy(s,j+1,i+1);
      Write(Mul[i][j]);
      if j<m-i-1 then Write(',');
    end;
    WriteLn(')');
  end;
end;

procedure Build_Xbase;
//builds base set X of the substring poset from Mul
//Xbase contains the complete partial ordered set
//because of this, this implementation is "naive"
// for long inputs, X gets HUGE because the poset is not constrained
//and the number of elements in it are n(n-1)/2 where n is the sequence length!!!
var
  i,j,k:  Integer;
  currentMaxIdx:Integer;
  Found:  Boolean;
begin
  for i:=0 to m-1 do
  begin
    currentMaxIdx:=-1;
    for j:=0 to m-i-1 do
    begin
      k:=0;
      Found:=False;
      while not Found and (k<=currentMaxIdx) do
      begin
        if Xbase[i][k].w=Mul[i][j] then Found:=True
        else Inc(k);
      end;
      if not Found then
      begin//new element in X
        Inc(currentMaxIdx);
        Xbase[i][k].w:=Mul[i][j];
        SetLength(Xbase[i][k].Outputs,2*(m-i-1));
        if i>0 then SetLength(Xbase[i][k].Inputs,2*(m-i+1));
      end;
      Inc(Xbase[i][k].c);
    end;
  end;
```

```
end;

procedure BuildEdges;
//builds the binary relations between pattern records in Xext
//a relation between two patterns is created only if constrains are satisfied
var
  i,j,k,p,q:  Integer;
begin
  for i:=0 to m-2 do
  begin
    Write('X['+IntToStr(i)+']=(');
    j:=0;
    k:=0;
    while k<m-i do
    begin
      Write('('+Xbase[i][j].w+','+IntToStr(Xbase[i][j].c)+')');
      if (i=0) or (Xbase[i][j].InputCount>0) then
      begin
        p:=0;
        q:=0;
        while q<m-i-1 do
        begin
//here's where we check for constraint satistfaction
          if (Pos(Xbase[i][j].w,Xbase[i+1][p].w)>0)
            and CheckConstraints(Xbase[i][j],Xbase[i+1][p]) then
          begin
            Xbase[i][j].Outputs[Xbase[i][j].OutputCount]:=@Xbase[i+1][p];
            Xbase[i+1][p].Inputs[Xbase[i+1][p].InputCount]:=@Xbase[i][j];
            Inc(Xbase[i][j].OutputCount);
            Inc(Xbase[i+1][p].InputCount)
          end;
          Inc(q,Xbase[i+1][p].c);
          Inc(p);
        end;
      end;
//------------------
      Inc(k,Xbase[i][j].c);
      if k<m-i-1 then Write(',');
      Inc(j);
    end;
    WriteLn(')');
  end;
  WriteLn('X['+IntToStr(m-1)+']=('+Xbase[m-1][0].w+','
    +IntToStr(Xbase[m-1][0].c)+'))');
end;

procedure PrintCPoset;
//can be done in many different ways, depending what one wants to show
var
  i,j,k,p:  Integer;
begin
  for i:=0 to m-2 do
  begin
    j:=0;
    k:=0;
    while k<m-i do
    begin
      if (Xbase[i][j].OutputCount>0) or (i=0) then
        WriteLn('('+Xbase[i][j].w+','+IntToStr(Xbase[i][j].c)+')->');
      for p:=0 to Xbase[i][j].OutputCount-1 do
        WriteLn(#9#9+'--->-('+Xbase[i][j].Outputs[p]^.w+','
          +IntToStr(Xbase[i][j].Outputs[p]^.c)+')');
      Inc(k,Xbase[i][j].c);
      Inc(j);
    end;
  end;
end;

const
//this is the test input string
```

```
TEST_INPUT='abcdefcdefababefcdefabcdcdefabcdefefefabcdef';

begin
  s:=TEST_INPUT;
  m:=Length(s);
  Assert(m>0,'The input sequence must be at least one character long.');
//initializing the dynamic arrays
  SetLength(Mul,m);
  SetLength(Xbase,m);
  for i:=0 to m-1 do
  begin
    SetLength(Mul[i],m-i);
    SetLength(Xbase[i],m-i);
  end;
//building arrays
  Build_Mul;
  Build_Xbase;
  BuildEdges;//creates the hierarchical, binary relations between pattern records
  WriteLn('The constrained poset');
  WriteLn;
  PrintCPoset;
//finalizing, i.e., destroying dynamic arrays
  for i:=0 to m-1 do
  begin
    Mul[i]:=nil;
    Xbase[i]:=nil;
  end;
  Mul:=nil;
  Xbase:=nil;
  ReadLn;
end.
```

Appendix 3. MedTest collection documents #803 and #1972

Document #803

Title: *Diagnosis of spirochetal meningitis by enzyme-linked immunosorbent assay and indirect immunofluorescence assay in serum and cerebrospinal fluid.*

Authors: *Stiernstedt GT ; Granstrom M ; Hederstedt B ; Skoldenberg B*

Journal: *J Clin Microbiol 1985 May;21(5):819-25*

Abstract: *The antibody response against a spirochetal strain isolated from Swedish Ixodes ricinus ticks was determined by enzyme-linked immunosorbent assay (ELISA) and indirect immunofluorescence assay of cerebrospinal fluid (CSF) and serum specimens from 45 patients with chronic meningitis. Samples of CSF, serum, or both from patients with various infections of the central nervous system, multiple sclerosis, syphilis, or infectious mononucleosis and from healthy individuals served as control samples. Probable spirochetal etiology could be demonstrated for 41 of 45 (91%) patients with clinical symptoms of chronic meningitis. Approximately 25% of the patients had significantly elevated titers of antibody to the spirochete in CSF but not in serum. The highest diagnostic sensitivity, 91%, was demonstrated by measurement of CSF antibodies and calculation of a spirochetal CSF titer index, which is the ratio of (ELISA titer in CSF/ELISA titer in serum) to (albumin in CSF/albumin in serum) and which also considers the degree of blood-CSF barrier damage. The highest specificity, 98%, was obtained by calculation of a CSF titer index.* ***Patients with short duration of disease were especially prone to be antibody negative in serum but positive in CSF. Significant rise in serum antibody titers was seldom demonstrated in patients treated with antibiotics****. It is concluded that measurement of CSF antibodies, especially by ELISA, is a highly sensitive and specific method for the immunological diagnosis of spirochetal meningitis.*

Document #1972

Title: *Serological diagnosis of Borrelia meningitis.*

Authors: *Stiernstedt G ; Granstrom M ; Hederstedt B ; Skoldenberg B*

Journal: *Zentralbl Bakteriol Mikrobiol Hyg [A] 1987 Feb;263(3):420-4*

Abstract: *The antibody response against a Borrelia strain isolated from Swedish Ixodes ricinus ticks was determined by enzyme linked immunosorbent assay (ELISA) and indirect immunofluorescence assay (IFA) of cerebrospinal fluid (CSF) and serum specimens from 45 patients with chronic meningitis. Probable Borrelia etiology could be demonstrated in 41 of 45 (91%) patients with clinical symptoms of chronic meningitis. Approximately 25% of the patients had significantly elevated titer of antibody to the spirochete in CSF but not in serum.* ***Patients with short duration of disease were especially prone to be antibody negative in serum but positive in CSF. Significant rise in serum antibody titers was seldom demonstrated in patients treated with antibiotics****.*

Appendix 4. Selected morphological equivalence sets induced by DDAM model from the MedTest collection

angi {itis | ocardiographic | oedema | ogram | ograms | ographical | omas | omatous | oplasty}
arterio {lopathy | sclerosis | sclerotic | sus | venous}
bacter {ascites | iologic | iological | iologically | iology | iuri | oides}
bio {activation | availability | degradable | feedback | log | mechani | mechanically | mechanics | medical | synthesis | synthetic | transformation}
candid {a | ate | ates | iasis | osis | uria}
cardio {acceleration | acceleratory | circulatory | cyte | cytes | depressive | dynamic | genic | graph | lipin | logical | logist | logy | megaly | myopathies | myopathy | protective | pulmonary | respiratory | selectivity | thoracic | toxic | toxicity | vascular | version}
cyt {ochemical | ochemistry | okines | okinetic | ologic | ological | ology | olysis | olytic | ometric | ometry | oplasm | oplasmic | oprotective}
dermat {ologic | ologist | ologists | ology | opathic | osis}
dihydro {alprenolol | chloride | folic | testosterone}
epidem {iologic | iological | iology}
equi {analgesic | diuretic | libration | librium | molar | p | valent | vocal}
esophag {ectomies | ectomy | ogram | ographic}
extra {adrenal | articular | cted | cts | cutaneous | diol | dural | esophageal | hepatic | medullary | muscular | neous | pulmonary | renal | skeletal | stimulus | systoles | systolic | thoracic | thymic | vas | vasation | vascular | version}
fluoro {cytosine | immunoassay | photometry | scopy | sis | tic | uracil}
gastro {duodenal | enter | esophageal | intestinal | paresis}
granul {ocyte | ocytes | ocytic | ocyto | ocytopenia | ocytotoxic | omas | omatous | opoiesis}
haemo {chromatosis | lysis | netics | phil | philia | philia | poietic | rrhag | rrhages | stasis | static | thorax}
hemat {ologic | ological | ology | omas | opoiesis}
hemoglobin {opathic | opathies | opathy | uria}
hepato {cellular | cyte | fugal | genic | ma | megaly | petal | renal | splen | toxic | toxicity | toxin | trophic | tropic | venous}
hydroxy {apatite | cortico | dopamine | lamin | lation | lations | myristoyl | phen | phenylglycol | progesterone | quinidine | urea | valproic}
hyper bilirubinaemia | dense | emesis | emia | fractionated | fusion | gammaglobulinaemia | gammaglobulinemic | glyc | graphic | immune | immuno | immunoglobulinaemia | natremia | natriuria | oxic | re | reactivity | secretion | sensitive | sensitivity | stimulation | tensive | thyroid | thyroidism | trans | transfused | transfusion | uricaemia | uricemia | ventilated | viscosity}
hypo {baric | chloremic | chromic | emic | gammaglobulinaemia | gammaglobulinemia | gammaglobulinemic | glycaemia | glycaemic | glycemia | glycemic | methylation | osmolality | perfusion | proliferative | respond | secretors | xemi}
hypo {activity | echoic | gastric | kal | kalaemia | natremia | thyroid | thyroidism | volemia | xaemia | xia}
immuno {adsor | adsorption | assay | blastic | chemical | cytochemical | cytological | cytology | depressive | diffusion | electrophoresis | fluorescence | fluorescent | gen | genic | genicity | gens | globulin | histochemically | incompetence | logic | logical | logically | logist | logy | morphological | patho | pathogenesis | pathogenetic | pathogenic | pathologic | pathologically | precipitated | precipitates | precipitation | proliferative | protein | reactive | reactivity | regulation | staining | therapy}
immunosuppress {ant | ants | ed | ion | ive | ives}
intra {cutaneous | cutaneously | dural | glomerular | hepatic | individual | lesional | lipid | medullary | myocardial | nasal | nuclear | operative | operatively | renal | sinusoidal | spinal | thecally | trabecular | uterine | vascular | ve | ventricular | vertebral}
lymph {ocytes | ocytic | ography}
macro {globulin | globulinemia | lide | melanosomes | nutrient | phag | phages | scopic | vesicular}
mega {colon | karyoblastic | karyocyte | karyocytes | karyocytic | mitochondrial | voltage}
multi {agent | determined | hospital | injection | mer | meric | modal | modality | nodular | nuclear | nucleated | organ | plication | system | therapy | transfused | variate}
my {algia | algias | astheni | celial | enteric | oblasts | ocytes | oelectric | ogenesis | ogenic | oglobin | ometrial | opathic | opathy |

oplasm | riad}
myel {oblastic | oblasts | ocytes | ocytic | ogenous | ogram | ographic | ographically | ography | opathy | oproliferative | otoxicity}
myo {cardium | clonic | clonus | cytolysis | filament | inositol | lemma | pericarditis | tubes}
naso {duodenal | enteric | gastric | pharyngeal | pharynx | tracheal}
nephro {calcinosis | n | ns | sclerosis | tic | toxic | toxicity | toxins}
neur {ographic | ography | ologic | ological | ologically | ologists | ology | opathic | opathies | opathologic | opathological | opathology | opathy | oprotective | osis | otropic}
neuro {active | anesthesiologists | behavior | chemical | chemistry | circulatory | cytoprotective | endocrine | epidemiology | functional | genic | hypophyseal | hypophysial | leptic | leptics | muscular | ns | peptides | psych | radiological | ses | surgical | tics | toxic | toxicity}
osteo {arthropathy | articular | clast | clasts | dystrophy | genesis | lytic | mas | myelitis | necrosis | pathic | sarcoma | sclerosis}
ov {arial | aries | erload | erloading | erly | erlying | erutilization | ulate | ulations | ulators}
path {ogen | ogenesis | ogenic | ogenicity | ogens | ologic | ological | ologically | ologies | ologist | ologists | ology | omorphology | way | ways}
peri {arteritis | conceptional | fistular | fusion | hemorrhagic | infarct | menopausal | myocardial | myocarditis | natal | natally | neal | operative | operatively | ovulatory | pancreatic | stomal | tonsillar | transplant | valvular | vasculitis}
poly {arteritis | chlorinated | cythemic | ethylene | graph | graphic | hydramnios | mer | myositis | neuritis | neuropathies | neuropathy | radiculitis | saccharide | saccharides | sorbate | therapy | tomography | transfused | urethane | uria | valent}
post {administration | angiographic | carditic | coitus | conceptional | conversion | erior | eriorly | exposure | glomerular | grafting | hepatic | infarction | infectious | injection | ischemia | ischemic | menopausal | myocardial | natally | operative | operatively | ovulation | renal | resuscitation | shunt | streptococcal | surgical | test | transfusion | transplant | transplantation | treated | treatment | ulate | ulated | ural | ures}
pre {albumin | capillary | cede | clinical | clude | cluded | culture | cur | cursors | determined | diagnostic | ference | ferences | glomerular | gnancies | implantation | ischemia | ischemic | leukemic | lude | mature | maturity | menopausal | mised | mycotic | natal | operative | operatively | renal | requisite | selected | serv | specified | ss | syncope | test | tested | transfused | transfusion | transplant | treated | treatment | tumour | valent}
psych {ical | odynamic | ologically | ologists | ology | opathological | osis | otherapy}
spondyl {arthropathies | arthropathy | itis | osis | otic | otomy}
sub {divided | division | divisions | endothelial | epithelial | fertility | glottic | group | ictal | ictally | ject | jected | jective | jectively | jects | mitted | ordinate | unit | units | verted}
tachy {arrhythmias | cardia | phylaxis | pnea | pnoea}
thrombo {cyte | cytopenia | cytopenic | cytosis | emb | globulin | lysis | lytic | phlebitis | sed | ses | sis | tic | xane}
trans {aminase | ected | endothelial | facial | fer | formation | fusing | fusional | fusions | glut | glutaminases | hepatic | ients | lation | mission | mitted | mitting | peptidase | position | sternal | temporal | thoracic | zygomatic}

{mono | photo}	chemotherapy
{ampi | beni | cycla | cyclo | peni}	cillin
{nonthrombo | pan | thrombo}	cytopenic
{cholecyst | esophag | ile | lob | prostat | splen | sympath | thym | thyroid}	ectomy
{francis | legion | pasteur | rub}	ella
{chemo | neuro | non}	endocrine
{clon | deoxyur | guan | guan | pyr | quin}	idine
{anti | non | post | pre}	ischemic
{extra | intra | juxta | supra}	medullary
{cardio | hepato | organo | spleno}	megaly
{psycho | radio | spectro}	metric
{anti | intra | multi}	nuclear
{angi | phon}	ocardiographic
{crypt | enter | strept}	ococcal
{ellipt | granul | hepat | histi | leuc | leuk | lymph | mon | my | myel | ne | phag | plasm | splen | thym}	ocytes
{granul | histi | lymph | mon | myel | phag}	ocytic
{ellipt | leuc | leuk | mast | mon | phag | plasm | ple | reticul}	ocytosis
{haem | hem | pharmac | psych}	odynamic
{cancer | histi | leukem | my | onc | opath | path | path | therm}	ogenesis
{all | aut | end | ex | hom | myel | onc}	ogenous
{cyt | dermat | haemat | hemat | hist | method | morph | neur | nos | onc | opath | ophthalm | path | path | phenomen | physi | rheumat | ur}	ologic
{cyt | endocrin | enzym | haemat | hemat | hist | mat | method | morph | morph | neur | neuropsych | nos | omorph | onc | opath | path | rhe | rheumat | vir}	ological
{dermat | enter | nephr | neur | ophthalm | path | psych | techn}	ologists
{amid | anxi | cyt | em | fibrin | glyc | haem | hem | nonhem | prote | spasm}	olytic
{aden | angi | granul | myx}	omatous
{cyt | gas | ge | is | man | man | morph | pupill | spir}	ometric
{clinic | hist | neur | physi}	opathologic
{co | intra | non | peri | post | pre | re | unco}	operative
{bas | eosin}	ophil
{cyt | end | tox}	oplasmic
{colon | end | laryng}	oscopy
{aden | anastom | aspergill | candid | collagen | cryptococc | dermat | diagn | mononucle | myc | necr | neur | oplasm | papul | psych | scler | spondyl | toxoplasm | tubercul}	osis
{ana | pro | tachy}	phylaxis
{hemi | tetra}	plegia
{eu | non | normo}	prolactinemic
{hypo | immuno | lympho}	proliferative
{cross | hyper | immuno | sero}	reactivity
{chole | haemo | meta | ortho}	stasis
{cyto | haemo | homeo | hydro | ortho}	static
{cu | instan | simul | spon}	taneously
{dia | pros | syn | syn}	thesis
{cardio | hepato | myelo | nephro | neuro}	toxic
{hyper | multi | non | poly | pre | re | un}	transfused

Appendix 5. Selected lexical equivalence sets induced by DDAM model from the MedTest collection

antibody	{activity_to_ | detectable | levels | response | screen | titer | titers | titers_ | titres}
_in_patients_with_	{a_ | acute | acute_ | acute_myocardial_infarction_ | advanced_ | advanced_breast_cancer | advanced_breast_cancer_ | aids | alcoholic | aplastic_anemia | ascites_due_to_ | barrett | barrett's_esophagus | barrett's_esophagus_ | barrett's_syndrome | cardiac | chronic | chronic_ | cirrhosis | cirrhosis_and_ | cirrhosis_and_ascites | cirrhosis_of_the_liver | complex | complex_ | congestive_heart_failure | crohn | crohn's | crohn's_disease | extensive | heart_ | hemophilia | hepatic | hepatic_ | impaired | inducible | inducible_ | liver_ | liver_cirrhosis_and_ | malignant_ | mastocytosis | mitral_valve_prolapse. | myocardial_infarction | or_without_ | other_ | previously | prior | severe_ | symptoms_of_ | symptoms_of_gastroesophageal_reflux | systemic_ | thalassemia | thalassemia_major | transient | ulcerative_colitis}
studies	{comparing_ | demonstrated_ | examining | included_ | indicate | indicate_that_ | investigating_ | performed | reported | reported_ | showed | suggest_that_}
_study_was_	{carried_out_on_ | conducted | designed | designed_to_ | performed | undertaken | undertaken_in_ | undertaken_to_}
_was_used_to_	{assess | evaluate | examine | measure}
_were_significantly_	{better | better_than_the_ | different | different_ | elevated | greater | higher | improved | increased_in_ | less_ | lower | more_ | reduced | smaller | smaller_in_}
blood_	{bank | centers | clots | components | flow | gas | gases | loss | pressure | pressures | products | supply | transfusion | transfusions | vessels | volume}
cerebral	{_and_ | _arterial_ | _arterial_spasm | _blood_flow | _blood_flow,_ | _blood_flow_ | _blood_flow_(| _blood_flow_and_ | _blood_flow_was_ | _blood_vessels | _cortical_ | _energy_ | _ischemia._ | _ischemia_and_ | _perfusion_pressure | _protection._ | _vascular_ | _vasospasm_ | _vasospasm_in_}
liver	{,_and_ | ._ | _and_ | _biopsies_from_ | _biopsy_ | _biopsy_material_ | _blood_flow | _blood_flow._ | _blood_flow_ | _blood_flow_(| _blood_flow_was_ | _cell_ | _cirrhosis._the_ | _cirrhosis_ | _cirrhosis_and_ | _cirrhosis_and_ascites._ | _damage,_ | _disease,_ | _disease._ | _disease_ | _disease_and_ | _disease_due_to_ | _disease_in_ | _diseases._ | _function_ | _function_tests | _function_tests_ | _function_tests_(| _metastases,_ | _metastases_and_ | _tests,_}
pulmonary	{_arterial_ | _arterial_hypertension | _arterial_pressure | _arterial_pressure,_ | _artery_ | _capillary_ | _complications_in_ | _consequences_of_ | _disease_ | _fibrosis._ | _function,_ | _function_ | _hypertension,_ | _hypertension_(| _hypertension_and_ | _infection_ | _vascular_ | _vascular_resistance_ | _wedge_pressure_}
tumor	{,_and_ | _cell_ | _doses_ | _is_ | _mass_ | _of_ | _size_and_ | -associated_}
-year-old_	{boy | female | girl | male | man | woman}

{breast | cervical | lung | metastatic | national institute | ovarian}	_cancer_
{congestive | idiopathic}	_cardiomyopathy,_
{coronary | critical | extended | health | intensive | medical | patient | primary | supportive}	_care_
{_non-cross-resistant | adjuvant | cancer | combination | initial | maintenance}	_chemotherapy_
{_ascitic | _extracellular | ascitic | body | cerebrospinal | extracellular | transformer pyrolysate}	_fluid_
{_a_high | _a_higher | _the_high | _with_a_low | annual | greater | high | higher | increased | low | lower | overall | similar}	_incidence_of_
{affected | autonomic | cranial | median | parasympathetic | peripheral | phrenic | redundant}	_nerve_
{._five | ._two | _fewer | fewer | four | two}	_patients_in_the_
{2 | 3 | 5 | 6 | 7 | 10 | 11 | 12 | 14 | 15 | 17 | 20 | 22 | 24 | 25 | 28 | 30 | 31 | 32 | 36 | 40 | 42 | 45 | 47 | 48 | 50 | 57 | 58 | 68 | 72 | 73 | 80 | 84 | 89 | 105 | 120 | 131 | 164 | 200 | 298 |). | ,_all | ,_nine | ._many | ._most | ._three | ._two | ._we_conclude_that | ._when | _12 | _20 | _among | _and_two | _breast_cancer | _cirrhotic | _consecutive | _control | _diabetic | _evaluable | _female | _from_two | _hospitalized | _in_12 | _in_28 | _in_57 | _in_diabetic | _in_eight | _in_many | _in_selected | _in_some | _in_the_5 | _in_two | _male | _nine | _of_11 | _of_15 | _of_18 | _of_200 | _of_21 | _of_25 | _of_50 | _of_all | _of_eight | _of_five | _of_some | _of_those | _patients. | _postmenopausal | _premenopausal | _selected | _stroke | _symptomatic | _ten | _therapy_for | _to_treat | _transfused | _treated | _treatment_for | _unselected | adult | all | almost_all | among | between | cirrhotic | consecutive | eight | er-positive | euthyroid | fifteen | five | for | four | from | identifying | in | include | many | of | -one | postmenopausal | seven | -seven | seventy-four | six | -six | stroke | ten | those | three | -three | to | transfused | transplant | twenty | twenty-four | two | -two | when}	_patients_with_
{data | findings | reports | strongly | studies | we}	_suggest_that_
{_assess | _reduce | _reduced | _reduces | increases | reduced}	_the_risk_of_
{_adjuvant | _endocrine | adjuvant | antibiotic | chemohormonal | endocrine | induction | pacemaker}	_therapy_in_
{._one_patient | _animals | _being | _colds | _in_children | _in_rats | _in_the_group | _infections | _mycosis_fungoides | _previously | _women | animals | being | children | group | hosts | initially | myasthenia_gravis | not | patient | patients | rats | recipients | successfully | were | women}	_treated_with_
{_breast_cancer | _infections | _mice | _multiple_myeloma | _patients_with_multiple_myeloma | _rats | _stage | infections | patients}	_were_treated_with_
{_basilar_ | _carotid_ | _coronary_ | _for_coronary_ | _middle_cerebral_ | _on_hepatic_ | basilar_ | coronary_ | femoral_ | pulmonary_ | vertebral_}	artery
{_left_ventricular_ | _renal_tubular_ | autonomic_ | organ_ | renal_	dysfunction
{,_and_ | ._the_ | _after_ | _and_ | _cardiac_ | _failure_and_patterns_of_ | _heart_ | _hepatic_ | _renal_ | _to_treatment_ | _treatment_ | _with_renal_ | graft_ | heart_ | hepatic_ | renal_}	failure
{_proliferative_ | membranous_ | necrotizing_ | proliferative_}	glomerulonephritis
{cyclosporine | drug | endotoxin | exercise | fluoride | furosemide | insulin | methylnaphthalene | pregnancy | radiation | serum | spironolactone | steroid | transfusion | valproate | valproic_acid | vasopressin | virus}	-induced_
{_days_ | _five_years_ | _h_ | _months_ | _weeks_ | _years_ | days_ | h_ | months_ | weeks_ | years_}	later
{,_weight_ | ._ | _blood_ | _bone_ | _by_a_ | _the_ | blood_ | graft_ | pregnancy_ | weight_}	loss
{_and/or_ | _and_viral_ | _bacterial_ | _chronic_ | _of_bacterial_ | _tuberculous_ | _with_listeria_monocytogenes_ | bacterial_ | purulent_ | tuberculous_ | tuberculous_}	meningitis
{_during_the_ | _during_the_exposure_ | _during_the_follow-up_ | _during_the_ischemic_ | _during_this_ | _follow-up_ | _incubation_ | _latency_ | _month_follow-up_ | _over_a_four-year_ | 4-week_ | -day_treatment_ | during_this_ | follow-up_ | ischemic_ | month_ | -	period

month_ | post-operative_ | six-month_ | study_ | treatment_ | week_ | -week_ | year_ | -year_ }	
{ _thrombocytopenic_ | anaphylactoid_ | henoch-sch:onlein_ | sch:onlein-henoch_ | thrombocytopenic_ }	purpura
{_bile_acid_excretion_ | _death_ | _heart_ | _in_heart_ | _in_response_ | _mortality_ | _of_heart_ | _of_the_heart_ | _prevalence_ | _response_ | _survival_ | _the_ | glomerular_filtration_ | heart_ | mortality_ | remission_ | response_ | survival_}	rate
{_(guillain-barr:e_ | _bowel_ | _guillain-barr:e_ | _malignant_ | _nephrotic_ | _patients_with_barrett's_ | _polycystic_ovary_ | _reiter's_ | _this_ | _with_barrett's_ | acquired_immune_deficiency_ | acquired_immunodeficiency_ | bannwarth's_ | barrett's_ | bartter_ | bartter's_ | fisher's_ | hellp_ | hepatorenal_ | immunodeficiency_ | -like_ | polycystic_ovary_ | respiratory_distress_ | reye-like_ | 's_ | sezary_ | this_ | uremic_}	syndrome
{_after_ | _cadaveric_renal_ | _cardiac_ | _in_bone_marrow_ | _in_the_ | _renal_ | bone_marrow_ | marrow_ | renal_}	transplant