A Study of Semantics Across Different Representations of Language

by

Dhanush Dharmaretnam
BTech., Dr. M.G.R University, Chennai, India, 2012

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

MASTER OF SCIENCE

in the Department of Computer Science

A Study of Semantics Across Different Representations of Language

by

Dhanush Dharmaretnam
BTech., Dr. M.G.R University, Chennai, India, 2012

Supervisory Committee

Dr. Alona Fyshe, Supervisor
(Department of Computer Science)

Dr. Kwang Moo Yi, Departmental Member
(Department of Computer Science)

ABSTRACT

Semantics is the study of meaning and here we explore it through three major representations: brain, image and text. Researchers in the past have performed various studies to understand the similarities between semantic features across all the three representations. Distributional Semantic (DS) models or word vectors that are trained on text corpora have been widely used to study the convergence of semantic information in human brain. Moreover, they have been incorporated into various NLP applications such as document categorization, speech to text and machine translation. Due to their widespread adoption by researchers and industry alike, it becomes imperative to test and evaluate the performance of different word vectors models. In this thesis, we publish the second iteration of BrainBench: a system designed to evaluate and benchmark word vectors using brain data by incorporating two new Italian brain datasets collected using fMRI and EEG technology.

In the second half of the thesis, we explore semantics in Convolutional Neural Network (CNN). CNN is a computational model that is the state of the art technology for object recognition from images. However, these networks are currently considered a black-box and there is an apparent lack of understanding on why various CNN architectures perform better than the other. In this thesis, we also propose a novel method to understand CNNs by studying the semantic representation through its hierarchical layers. The convergence of semantic information in these networks is studied with the help of DS models following similar methodologies used to study semantics in the human brain. Our results provide substantial evidence that Convolutional Neural Networks do learn semantics from the images, and the features learned by the CNNs correlate to the semantics of the object in the image. Our methodology and results could potentially pave the way for improved design and debugging of CNNs.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

My supervisor, Dr. Alona Fyshe, for her valuable guidance, and support in my research and academics over the entire course of my graduate study. I am also really grateful to her for the mentoring and feedback that she offered during the writing phase of this thesis.

Isabelle, Chris, Ed, Maryam, Cole - my lab mates from the language and learning lab of the University of Victoria for their valuable advice and encouragement over the last two years of my graduate study.

Compute Canada, and Westgrid Computing, for providing me with the necessary computing resources to do my research.

DEDICATION

Dedicated to my loving grandfather
who always supported my quest for knowledge and education.

Chapter 1

Introduction

Semantics is the branch of linguistics which is concerned with the study of language and how we understand the meaning of words and concepts. Cognitive Psychologists study semantics to understand various mechanisms that are involved in the thought process and mental representations which are fundamental to any language. Over the recent years, several studies have been performed to understand the similarities between semantic features across different representations such as the brain, image and text. The results from these studies have significant implications in the field of Computation Linguistics and Artificial Intelligence (AI). For example study of the semantics of image features could help us build better computer systems that are capable of performing object recognition such as those used in self-driving cars.

Distributional Semantic (DS) models or word vectors are based on the occurrence of words in large corpora of text. The main intuition behind these models is that words which are used in the same context in texts could have some similarities with each other. In these semantic models, the meaning of a word is represented by a vector which approximates the relative position of a word in a large high dimensional space. The words which are used in the same context and thus have similar meaning are positioned closer to one another in the vector space. Its vital to perform evaluation of these word vectors since their usage is widely spread across the field of AI. Some of the popular DS models are discussed under chapter 2 of this thesis.

1.1 The Study of Human Brain

Computational linguists have also adopted semantic models based on text to study semantic representations in the human brain. The study of the human brain helps us to gain knowledge about how language is processed and interpreted by us. Over the last decade, we have made significant progress in unlocking the complex process through which our brain decodes and understands the meaning of various concepts. This could be credited to the development of various brain imaging techniques such as Functional Magnetic Resonance Imaging (fMRI), Electro-Encephalogram (EEG), Magnetoencephalography (MEG) etc. Linguists studying the brain often use Distributed Semantic models as ground truth in their research.

The inception of such studies began with Mitchell et al. (2008) who were the first to use text-based DS models to study semantic features of concepts in human brain collected using fMRI [52]. They showed that a computational model trained on text corpora could predict neural patters of participants recorded using fMRI. Murphy et al. (2009) showed that text-based language models could predict EEG activity related to semantics [54]. This was followed by Sudre et al. (2012) who performed similar work using MEG data collected from subjects viewing images of concrete nouns [71]. They also evaluated the performance of various corpus-based models on their task. These works indicate that the semantic representation extracted from corpus could contribute to the study of the brain.

1.2 Introduction to BrainBench

Anderson et al. argued that a strong correlation between the neural signal and corpus-based models [52, 54, 71] is a good indicator that brain data could be used to test corpus-based models [2]. If a corpus-based model could approximate and predict the semantic feature representation in the brain, then that model could have features that represents how humans learn and understand language. Similar arguments were made by Murphy et al. (2012) [55] and Anderson et al. (2015) [5]. Based on these recommendations, Xu et al. (2016) introduced *BrainBench*: a system designed to test corpus-based distributional models of semantics using brain data [34].

BrainBench used two brain image datasets: a fMRI dataset [52] and a MEG dataset [71] collected from nine participants imagining 60 concrete nouns. *"Concrete nouns (e.g. Car, Apple etc.) are things which can be experienced by the five senses*

whereas abstract nouns are intangible concepts such as ideas and emotions (e.g. freedom, happy)" [77]. They tested six popular word vector models against fMRI and MEG datasets and reported comparable performance to other systems which evaluate and benchmark DS models based on behavioral data. Another notable feature of BrainBench was that it is fast and computationally cheap.

However, BrainBench tests include just 60 concrete nouns, and does not include any abstract nouns. This could imply that the BrainBench tests could be more biased towards word vectors which has a higher distribution of concrete nouns over abstract. Moreover, the tests are derived from only two dataset sources and based on one language (English). This thesis aims to address some of the above limitations of BrainBench.

The contributions made to BrainBench (*Contribution A*) by this thesis are addressed below:

- Addition of an Italian fMRI brain data into BrainBench and study of the performance of non-English word vectors.
- Introduction of abstract nouns into the BrainBench tests and evaluation of the performance of various word vectors on abstract nouns compared to concrete nouns.
- Addition of an EEG dataset to BrainBench
- The integration into Brainbench the ability to study the performance of word vectors across anatomical brain regions.
- An increase in the coverage of concepts supported by BrainBench from 60 to 190 concepts.

1.3 Learning semantics from Images

BrainBench and other works are based on the study of semantics in the human brain which could improve our understanding of how we learn and interpret languages. Humans also learn semantics from visual input (sight). In fact, there has been research done to show that our first exposure to the semantics is through the visual stimulus captured by our eyes [57]. Looking at images, we can recognize content, derive

semantics, and sometimes images may even elicit an emotional response in us. It is the human visual cortex that helps us to interpret these visual stimuli and derive semantics from our sight.

Today, we have Artificial Intelligence (AI) which can recognize objects from images. Most popular of these AI is the Convolutional Neural Network (CNN) which is a type of Artificial Neural Network (ANN) loosely inspired by the human visual cortex. CNNs has importance in many areas of science. Self-driving cars use CNNs as their eyes on the road. Astronomers use CNNs to identify planets and stars from the millions of images taken by telescopes across the world. Doctors and diagnosticians use CNNs to study medical images and predict life-threatening diseases like cancer in patients. Its applications are limitless and ubiquitous.

Researchers strongly believe that CNNs and deep learning are the future of AI. Yet, we cannot currently explain why these networks work so well, sometimes even better than us humans at various tasks. Could we trust an AI to do our daily tasks if we don't understand its decisions or how it is working?

1.3.1 Black-Box nature of Convolutional Neural Networks

Researchers in the field of computer vision have created deeper and more complex CNN architectures in search of improved accuracies at the task of object recognition. This trend of going deeper with CNNs has lead to the creation of a myriad of black box architectures which works well at the task of object classifications. Yet, there is a lack of understanding about why they perform so well or how could they be improved [81]. There has been some interest in the computer vision community to provide insights into the performance of these networks. Such insights could help us to train faster and more robust CNNs. *How do Convolutional Networks see the world? What are the features learned by these networks? What makes one network architecture better than the other?* These are some of the questions asked by the deep learning community.

Researchers have focused on methods such as visualization of features of different layers of CNN [81, 78, 63] or even mathematical models [49] to understand and explain the predictions of CNNs. However, in this thesis, we propose a novel method to understand CNNs by studying the semantic representation through the layers of CNNs. Convolutional Neural Networks might learn semantics from the images, and we believe that the features learned by the CNN could correlate with the semantics

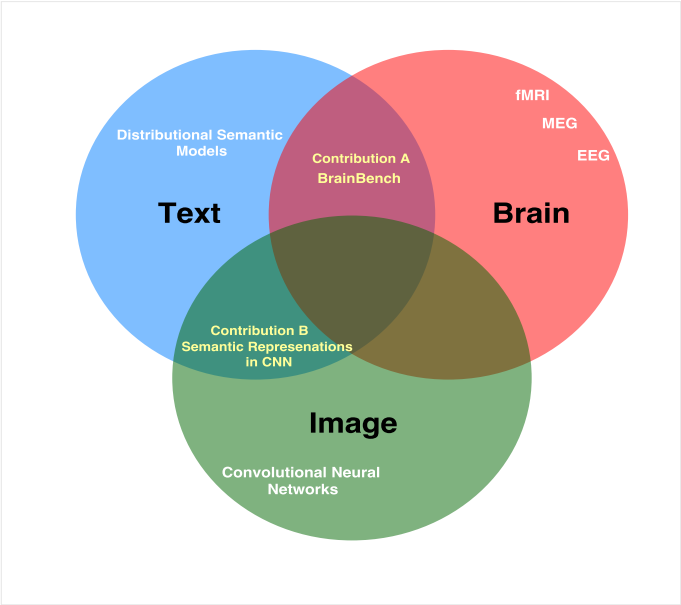


Figure 1.1: Representations of Semantics highlighting contributions of this thesis

of the image. We, therefore, propose that studying the semantic representation instead of feature representation through the layers of CNN could provide us with other valuable insights.

The convergence of semantic information in these networks could be studied with the help of word vectors following the same methodologies used to study semantics in human brain [52, 54, 71]. The study of semantic representation through the layers of CNN could help us understand how various CNN architectures differ from one another and could potentially provide us with a methodology to improve them. Research has shown that a CNN does learn abstract features from images [81] but does it understand semantics from the images that it's trained on? If a CNN does understand semantics, could we then use that knowledge to explain the decisions made by these networks and unlock the black box?

CNNs are trained on images, and DS models are trained on the text. Yet they could share a similar semantic representation since they are modeling the same real-world concepts. Studying semantic representations in CNNs could contribute to better understanding of CNNs and potentially pave the way for improved design and debugging of CNNs. To demonstrate an application, we also conduct experiments to determine the position in the hierarchy of the CNN where misclassifications of images could occur.

The contributions made by this thesis to understand CNNs (*Contribution B*) are summarized below:

- A novel methodology to study semantic representation through hierarchical layers of CNN.
- The study of hidden representations of images that are misclassified.

1.4 Thesis Organization

Chapter 1 includes a brief introduction to the world of semantics; it's various representations and description of the problem statements.

Chapter 2 talks about related work in this area of research and provides a detailed walkthrough of all the major contributions made in this field.

Chapter 3 talks about contributions made to BrainBench including the addition of two new datasets and study of semantic features within various regions of human brain. The results of these experiments along with the discussions are also included in this chapter.

Chapter 4 describes the study of semantic representations through the layers of popular convolutional neural networks with results and discussions.

Chapter 5 gives a summary of the problem statements and the contribution of this thesis in solving those problems. We also discuss various future improvements in this area of research.

Chapter 2

Background and Related Work

Computational Linguistics is a field which focuses on understanding the various process through which humans learn, represent, and interpret languages. The lessons learned from such studies could help us to engineer various AI models which could then be used to mimic humans at various tasks. For example, the study of semantic and syntactic properties of language using large text corpora has helped us to develop better speech to text and machine translation systems. The study of semantic properties of images has enabled us to build computational models which have surpassed the average humans capabilities at the task of object recognition from images. Researchers have also studied language representation in human brain through brain imaging technologies such as fMRI, MEG, EEG etc.

Researchers in the field of psycholinguistics have tried to isolate regions in the brain corresponding to the processing of language. These studies could help us develop better medical techniques to help people suffering from language deficiencies such as Alzheimers disease, severe brain damage, Semantic Dementia etc. The language representations in the brain are often studied by comparing neural activations extracted using brain imaging technologies with language models derived from images and text. Brain, images and text could be considered as different representations of language and are modeling the same world phenomenon. For example, reading or seeing an image of a concept such as *cat* brings about the same response in us human beings.

This thesis mainly focuses on the study of similarities between semantic models build from brain, text and images sources. To summarize, this chapter will focus on various related work in the below areas,

1. The study of semantics in the human brain with the help of semantic models build from text corpora.
2. Methods to evaluate Distributed Semantic models build from text corpora.
3. Study of semantics in images.
4. Convolutional Neural Networks (CNN) and need to visualize and understand these networks.
5. Existing techniques developed to visualize and understand CNN models and its limitations.

2.1 Learning Semantics in Brain

Language acquisition is considered as an important human trait, and the study of the human brain could help us to gain knowledge about how language is decoded and interpreted by us. The development of brain imaging technologies such as Functional Magnetic Resonance Imaging (fMRI), Electro-Encephalogram (EEG), Magnetoencephalography (MEG) has helped Linguists studying semantic decoding in the human brain. Semantic models build from text-based corpora are often used as ground truth to study neural activations related semantic decoding in the brain. The primary reason for such adoption is that corpus-based models are easier to construct and use in the experiments as compared to models build on brain imaging data or images.

Brain imaging technologies such as fMRI, EEG, and MEG are expensive, and construction of semantic models based on these techniques is not feasible. Moreover, the time required to collect enough brain data to build semantic models is extensive. It is not feasible to construct models based on images either since collection and storage of large database of images is not an easy task. The extraction of semantic features from images are computationally expensive. These reasons could explain why distributional semantic models based on the text are widely adopted by researchers to study other representations of language.

The use of semantic models derived from textual data to study neural activations in the brain began with the work of Mitchell et al. (2008) [52]. In their experiments, nine right-handed participants were presented with line drawings, and noun labels of 60 concrete nouns on a screen and the neural patterns of the participants were recorded using fMRI. Semantic features corresponding to 25 verbs based on

their co-occurrence with each other were then extracted from a large text corpus. Subsequently, a computational model was then trained to predict the neural activity corresponding to unseen concepts.

Murphy et al. (2009) showed that text-based language models could predict EEG activity related to semantics in the brain [54]. This was followed by Sudre et al. (2012) who performed similar work using MEG data collected from subjects viewing images of 60 concrete nouns [71]. Anderson et al. (2015) found that text-based distributional semantic models are better at predicting conceptual similarity in fMRI brain scans of areas linked with linguistic processing, whereas semantic features extracted from images are better at accounting for similarity in visual processing areas in the human brain [2]. In short, semantic similarity based on co-occurrence of words in text corpora could be used to study semantic representations in the human brain.

2.2 Learning Semantics in Text

In the previous section, we discussed various techniques that are adopted by the Computational Linguistics community to study semantic representation in the human brain. We also discussed in brief about Distributional Semantic (DS) models or word vectors extracted from text-based corpora which are used as ground truth to predict activity related to semantics in the brain. In this section, we discuss some of the most popular word vector models that are available today. These models are extensively used in various NLP related tasks such as sentiment analysis, document classification, machine translation etc. The main idea behind word vectors is that meaning of a word can be inferred from contextual information.

People learn the meaning of words that they have never seen before by guessing the meaning of the new word from its context (by understanding the meaning of the known nearby words). In linguistics, we explain this in terms of word co-occurrence which is often associated with semantic proximity and is the principle idea behind the creation of most of present-day DS models or word vectors.

The idea of representing semantic information of a word using vectors is not new and could be traced back to Charles Osgoods work *Semantic Differentials* in 1965 [59]. In his work, Osgood used handcrafted methods to build feature representations to study semantic similarities between words. In a hand-crafted lexical source, the relationship between words is assigned by human annotated based on a set rule. In the early 2000s, the use of neural networks to generate word vector representations

became popular among the linguistics community. In 2003, Bengio et al. trained a probabilistic neural network model to learn the joint probabilistic distribution of sequences of words in a language [10]. Collobert and Weston, 2008 were perhaps the first to successfully demonstrate that neural network architecture could be used to learn word vectors representing semantic information from a text corpus [16]. They experimentally argued that word vectors learned from a large text corpus could be used as an effective tool to perform various downstream tasks in NLP.

Mikolov et al. introduced the **Word2Vec** model in 2013 and triggered a revolution in the field of NLP [50]. The Word2Vec model was released as a toolkit which made it easier to train new word vectors from a text corpus which culminated in their rapid adoption by the industry. Word2Vec uses a shallow two-layer neural network architecture to learn the semantic features from the text. The input to the network is a one-hot sparse vector representing a word and output layer is a probabilistic layer which predicts probabilities of various words in the vocabulary given the input word. This simple neural network is trained to perform a task such as to predict words used in context of a given word. After the network is fully trained, the weights corresponding to the hidden layers for a given input word is extracted as its word vector. The dimensions of the word vector is equivalent to the number of neurons in the hidden layer of the neural network. Word2Vec model consists of two distinct algorithms, a continuous bag of words (CBOW) and Skip-gram model.

- A CBOW model was trained to predict a target word given a set of words surrounding it. In vector space, this can be visualized as predicting the distance between two different word vectors which are in context to one another.

- For Skip-gram, the direction of prediction is reversed, i.e., from a source word it tries to predict all the words which are in context to the source word. The Skip-gram model trained on Google news dataset is a 300-dimensional vector.

Glove is a regression-based semantic model published by Pennington et al. in 2014 [60]. It introduces the concept of representing the relationship between two word as their co-occurrence probabilities. This 300-dimensional vector model was trained on a combined corpus of Wikipedia and Gigaword 5 [33].

Global Context is another DS model that takes into account both local, and global context of a document to learn the semantics of the word (Huang et al., 2012) [38]. This model could encode into the word vectors properties such as homonymy

and polysemy of a word. Homonymy refers to the relationship between words with identical forms, but different meanings. Oxford dictionary defines polysemy as *"the coexistence of many possible meanings for a word or phrase"* [67].

Cross-lingual is a model proposed by Faruqui and Dyer, 2014, takes into account semantic properties across various languages [21]. It was trained using both German and English words using WMT-2011 corpus and uses a shared semantic space to learn the word embeddings.

RNN model is based on a Recurrent Neural Network (RNN) that is trained to predict the next word in the sequence (Mikolov et al., 2011) [51]. Unlike Skip-gram and other models which derives word vectors from an n-gram of words (Three or four words which occur together), a RNN model can theoretically represent words at infinite distance from one another. This model trained on broadcast news transcriptions has a dimension of 640 for its word embeddings.

A Recurrent Neural Network is a type of Artificial Neural Network designed to model sequential information. A RNN considers inputs not only in the current time step but also the inputs from previous time steps to perform various network operations. Compared to other neural models, RNNs pass their hidden state across invocations which functions as a memory through the previous time steps. This network is mostly used as a generative model.

Non-Distributional is another semantic word vector model created by combining various hand-crafted lexical sources such as WordNet (Fellbaum,1998) [24], FrameNet (Baker,1998) [7], Emotion & Sentiment (Mohammad and Turney,2013) [53] etc. by Faruqui et al. [22]. In a hand-crafted lexical source, the relationship between words is assigned by human annotated based on a set rule. For example in WordNet, the words are arranged in a hierarchical manner based on their meaning. Word vectors produced by this model are highly sparse, and each dimension of a vector in this model could be traced back to a linguistic feature. An example of a linguistic feature would be emotions which are triggered by a concept (positive, negative, fear, anger etc.). Another example would be part of speech features such as noun, proverb, adjective etc.

The various word vector models described in this section are also referred to as *Distributed models of Semantics* (DS) because the semantic information of a word is distributed throughout the dimensions of the vector. Due to their widespread adoption into various NLP tasks, it becomes important to evaluate and benchmark these word vector models.

2.3 Evaluation of Word Vectors

Traditionally computational linguistic community has relied on word similarity tasks to evaluate and benchmark various word vectors. These methods are based on computing the distance between word vectors and some similarity measure assigned by human annotators for the same word pairs. Some popular word similarity tasks used to evaluate word vectors are discussed below:

MEN

A dataset that consists of 3000-word similarity pairs with human-assigned similarity judgments collected using Amazon Mechanical Turks (AMT) [12]. The words that occur at least 700 times from the *ukWaC* [26] and *Wackypedia* [9] corpora were included in this dataset. The human annotators were presented with a pair of words and asked to assign a similarity score in a standard 1-7 Likert scale.

WS-353

This dataset contains 353-word pairs assigned similarity scores by human annotators [27]. Agirre et al., 2009 claimed that similarity and relatedness denote different relationships among words. Based on this argument, the WS-353 dataset was further split into WS-SIM and WS-REL with each set containing 353 pairs of words [1].

SimLex-999

SimLex-999 dataset focuses on measuring similarity in semantic models rather than relatedness or association between words [36]. SimLex-999 has word pairs with varying degree of concreteness. The whole dataset is further divided into 666 noun-noun pairs, 222 verb-verb pairs and 111 adjective-adjective pairs.

Evaluating Word vectors Using Brain Data

The evaluation of word vectors using word similarity datasets are quite popular and widely accepted by the research community. However, Faruqui et al., 2016 highlighted some serious limitations with the use similarity datasets to benchmark and compare word vectors [23]. Most of the Distributed models for semantics are task specific and not entirely trained to capture word co-occurrence. Moreover, similarity datasets do not capture the notion of polysemy and therefore penalizes word vectors which

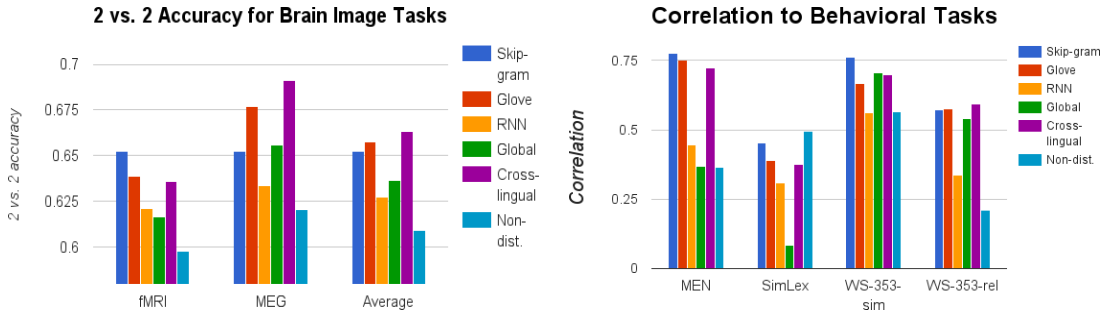


Figure 2.1: Comparison of BrainBench Performance against other word Similarity dataset
Left shows the BrainBench test results for 6 popular DS models and Right shows the comparative performance of the same DS models on other word similarity tasks. Image source: Xu et al.,2016 [34].

take into account of polysemy in words. The current evaluation techniques also fail to take into consideration of statistical significance when measuring the difference in the performance of two word vectors models on a similarity dataset. Based on their observations, Faruqui et al. concluded that more research into methods to evaluate word vectors were necessary.

In 2016, Xu et al. introduced *BrainBench*: a system designed to test text-based Distributed Semantic models using brain data [34]. As discussed in chapter 1, the first iteration of BrainBench was based on two brain image datasets: a fMRI dataset [52] and a MEG dataset [71] collected from nine participants. They tested six popular word vector models against the fMRI and MEG datasets and the results are summarized by the Figure: 2.1). In this thesis, we release the second iteration of BrainBench which is described in detail in the chapter 3 of this thesis.

2.4 Learning Semantics from Images

The beginning of 21st century witnessed tremendous development in the field of digital camera technology which resulted in the explosion of high resolutions images on the Internet. Image search and retrieval tasks on the Internet required the images to be annotated with semantic keywords labels. However, the volume of pictures uploaded to the Internet grew exponentially since the year 2000 and Labeling these images manually using human annotators were costly and time-consuming. Moreover, such methods had a high degree of annotation inaccuracy due to the subjectivity of

human perception. Therefore it became imperative to develop techniques to label these images into semantic categories automatically.

A popular technique introduced by researchers in computer vision is content-based image retrieval (CBIR). In this method, the images were indexed by low-level visual features such as color, texture, shapes. The low-level features such as color, texture, shape etc. are extracted from images using techniques such as HOG, SIFT etc. from whole or segmented regions of an image. Once these low-level features are extracted, supervised machine learning models such as Support Vector Machines (SVM) [25], Decision Trees [65] and Bayesian classifiers [40] were trained on these low-level features to learn high-level concepts from images. However, these supervised models still required the images to be manually annotated before they could be trained on extracted features. The use of unsupervised machine learning models such as k-means clustering was also popular in CBIR systems.

Most of state of the art systems in computer vision used today are based on neural networks. Unlike classical text-based and context based image retrieval techniques, neural networks learn features automatically from images. The origin of these computational models could be traced back to early 1970's and mimics the perceptual system in mammals.

The human visual cortex helps us to interpret the visual stimulus and derive semantics from our sight. In 1962, Hubel and Wiesel studied the perceptual system in cats and concluded that the visual cortex in the brain is made up of special arrangements of cells which are sensitive to only specific regions in the visual field [39]. These cells or neurons only fire in the presence of edges of certain orientations or patches of colour. They act as local filters and are tiled across the entire field of view. They also identified other complex cells which have a much larger receptive field and takes input from the cells which act as local filters. They, in turn, are connected to other cells with an even larger receptive field. This results in a columnar arrangement of cells and results in visual perception. The semantic representation in images could be studied with the help of Convolutional Neural Networks (CNN).

Convolutional Neural Networks is a type of Artificial Neural Networks (ANN) loosely inspired by the human visual cortex. They are quite popular among the computer vision community and are used for image recognition. Just like the brain, they also contain neurons (perceptron) which work together to form filters. These filters are slid across the entire image, and this results in feature maps. This sliding operation is commonly referred to as convolution. The first layer of any CNN is

a convolutional block which takes an image as input. The neurons in the beginning layers of a CNN learn low-level features such as edges, orientation, colour patches etc. These low-level features learned by the initial layers are given as input to latter layers which learns high-level feature representations. Based on these high-level features, a CNN can identify and label objects correctly [6, 47].

Prior to Convolutional Neural Networks, classical computer vision techniques such as the Naive Bayes classifier using a bag of visual features [17], hierarchical Bayesian models for object categorization [68] and many others were used for object recognition. These methods required a lot of preprocessing and extraction of handcrafted features such as Histogram of Oriented Gradients (HOG) or Scale-invariant Feature Transform (SIFT) from the images before they could be trained to predict images. However, in CNNs the features are learned automatically by the filters, and the input image requires very little pre-processing. CNNs also outperformed other classical methods in various image recognition tasks which led to their rapid adoption by researchers and industry.

2.4.1 A brief history of Convolutional Neural Networks

Neocognitron, the first CNN developed in 1982 by Kunihiko Fukushima used a hierarchical multiple layer architecture [29]. The Neocognitron could recognize various patterns in images based on the difference in their shapes. This was followed by LeNet architecture developed by Yann LeCun of Bell labs who demonstrated that Convolutional Neural Networks could be used for recognition of handwritten digits [44]. The development of faster computers and Graphical Processing Units (GPU) created a revolution in the field of CNNs. A major limitation of CNN is that it requires a large number of labelled images to learn from various patterns in images and make successful predictions.

In 2009, Fei-Fei et al. released *ImageNet* - a free database of 14 million images of 1000 categories, collected and labeled using the Internet [19]. This lead to the genesis of ImageNet Large Scale Visual Recognition Competition (ILSVRC)- a benchmark competition for object category classification from images [64]. Equipped with the processing power of GPU and labeled training images from ImageNet, Krizhevsky et al. released AlexNetwhich achieved a top 5 test error rate of 15.4% in the ILSVRC, 2012 competition becoming state of the art in the field of object recognition [43]. A top 5 error is defined as the rate at which the model does not predict the correct label

in its top 5 predictions. It may be worth noting that the second place entry in the competition was a non-CNN variant that had a top-5 error rate of 26.2%.

The success of AlexNet caught the attention of the computer-vision community, and Convolutional Neural Networks started gaining popularity. AlexNet had an eight-layer architecture and one of the deepest network created at that time. Zeiler and Fergus made small modifications to the AlexNet by reducing the stride and filter size of the first layer of AlexNet and performed an extensive hyper-parameter search during training. This modified network dubbed as ZFNet won the ILSVRC, 2013 competition with a top-5 error of 14.8% [80]. The success of AlexNet and ZFNet set the trend among the research community that having a large number of convolutional blocks and hidden layers in the architecture could somehow improve performance. In short, to get better performance, you need to have a deeper architecture with more convolutional layers. The term *deep learning* was coined to represent the creation and study of such large neural networks.

The VGGNet Architecture

ILSVRC, 2014 competition showcased further improvements in the development of the CNN architecture and image recognition. Two notable entries into this competition were the VGGNet and GoogLeNet. The VGGNet architecture designed by Zisserman and Simonyan of the Oxford University was the runner-up for the ILSVRC, 2014 competition in the object recognition category [66]. The VGGNet have six different architecture variants as shown in Figure: 2.2 and the configuration D produced the best results in the competition with a top-5 error rate of 7.3%. This network referred to as VGG16 has 16 layers (13 Convolutional layers and 3 fully connected layers) with over 138 million parameters. It has a homogeneous architecture and performs only 3*3 convolutions and 2*2 max-pooling throughout the architecture. Another important observation from Figure: 2.2 is that the number of filters doubles after each max-pooling layer which instill the notion of growing depth while shrinking dimensions of the features.

The VGGNet was among the first few CNN architectures which showed that depth of the network was a necessary component for achieving good performance in object recognition tasks. This network is also the preferred network among the computer vision community for extracting features from the images and its weights pre-trained on ImageNet are readily available for download.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 2.2: Variants of the VGGNet architecture
This table shows the different variants of the VGGNet architecture. The configuration D (VGG16) produced the best results in the ILSVRC, 2014 competition in the task of object classification. We use the VGG16 variant for our experiments. Image source: Zisserman and Simonyan.,2014 [66].

The Inception architecture

The GoogLeNet introduced by Szegedy et al. was the winner of the ILSVRC 2014 competition at the task of object recognition with a top-5 error of 6.7% [72] on the ImageNet cross-validation dataset. The main contribution of this network was the introduction of the **inception module** which drastically reduced the number of parameters in the network without loss in performance.

Before the introduction of GoogLeNet, other states of the art networks such as AlexNet [43], VGGNet [66], and ZF Net [80] etc. had a sequential structure formed by stacking convolutional layers, pooling and fully connected layers on top of each other. However, the inception modules have network operations ($1 * 1$, $3 * 3$ and

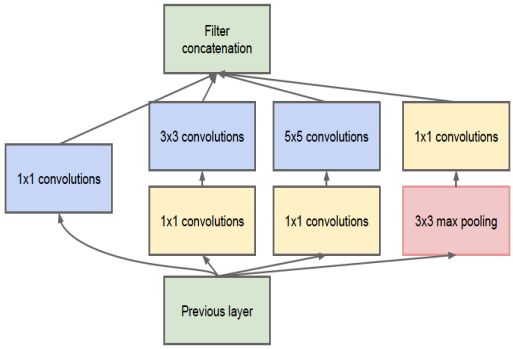


Figure 2.3: An Inception module.
An Inception module with convolution, max-pooling operations happening in parallel. Here the 1∗1 convolutions are used to perform dimensionality reduction. Image source: Szegedy et al.,2004 [72].

$5 * 5$ convolutions, max-pooling etc.) happening in parallel as shown in Figure: 2.3. The use of different filter patches with different strides helps in multi-level feature extraction and gives the option of varying receptive fields. The authors also argued that since pooling layers are an important component in state of the art CNNs, adding a max-pooling layer in parallel to the convolution operations could provide a boost in performance. The pooling layers help to reduces spatial sizes and helps to prevent over-fitting.

However, such an implementation could increase the number of parameters in the network and could lead to a *"computational blow up within a few stages"* [72]. In the optimized inception module, $1 * 1$ convolutions are used to perform dimensionality reductions before the expensive $3 * 3$ and $5 * 5$ convolutions. The $1 * 1$ convolutions reduce the depth of the volume of the output from the previous inception module by performing cross-channel convolutions. The $1 * 1$ convolutions also help to prevent over-fitting [45].

To understand how $1 * 1$ convolutions work, let us say for example the input to the inception module is $150 * 150 * 60$, and applying 20 filters in the $1 * 1$ convolution block could reduce the volume to $150 * 150 * 20$. Then this volume is used as the input to the $3 * 3$ and $5 * 5$ convolution blocks. Max-pooling reduces the width and height of the volume whereas 1*1 convolutions minimize the depth of volume.

In 2015, Szegedy et al. introduced Inception-v2 and Inception-v3 architectures which are slight variants of the original inception modules used in GoogLeNet [73]. In the Inception-v2 and Inception-v3, they introduced factorizations for convolutions

greater than $3*3$. For example, a $5*5$ convolution can be replaced by two consecutive layers of 3x3 convolutions and a $7*7$ convolution can be replaced by three consecutive layers of $3*3$ convolutions. Moreover, a $3*3$ convolution could be further broken down into a $3*1$ convolution followed by a $1*3$ convolution which results in a 33% reduction in computational cost. These factorizations result in different types of inception blocks as summarized in the Figure: A.2.

Residual Networks

The success of Inception and VGGNet architecture brought us closer to achieving human error at the task of object categorization in the ImageNet dataset. The human top-5 classification error on the ImageNet dataset is 5.1% [64]. GoogLeNet and VGG16 achieved error rates of 6.7% and 7.1% respectively which are very close to the human error. However, the ResNet architectures introduced as a part of the ILSVRC 2015 achieved top 5 error rate of 3.57% surpassing human error on the task [35].

The ResNet or Residual Networks by Kaiming He et al. introduced the concept of residual learning to tackle the problems that arise due to increasing depth of the convolutional neural networks [35]. The ResNet architecture incorporated techniques such as batch normalization, Skip-connections and use of ReLu as activation function to mitigates the effects of over-fitting and vanishing gradients. The final model which won the ILSVRC 2015 was 152 layers deep with only 60 million parameters compared to VGG19 which had 168 million parameters for 19 layers.

The principle idea behind the ResNet is the introduction of *"identity short-cut connections"* [35] that skips through one or more Identity layers in the architecture. An identity layer is a series of convolution layers followed by its activations group together to form one unit. In a simple CNN, an identity layer takes some input x and computes the transformation $H_{(x)}$ which is an entirely new representation of x. However, stacking multiple identity layers to form deeper networks is usually associated with a degradation problem. The authors Kaiming et al. argued that instead of computing the direct transformation from x to $H_{(x)}$, we could calculate the term $F_{(x)}$ and add it to the original input x to get $H_{(x)}$ [35]. This means that the identity module is only computing a small change $F_{(x)}$ and adding it to the original input x. This is termed as residual learning and is indicated by the Figure: 2.4

The Residual function is defined as $F_{(x)} = H_{(x)} - x$, where $H_{(x)}$ is the output after a residual operation. To summarize $H_{(x)}$ is learned as $F_{(x)} + x$ which is accom-

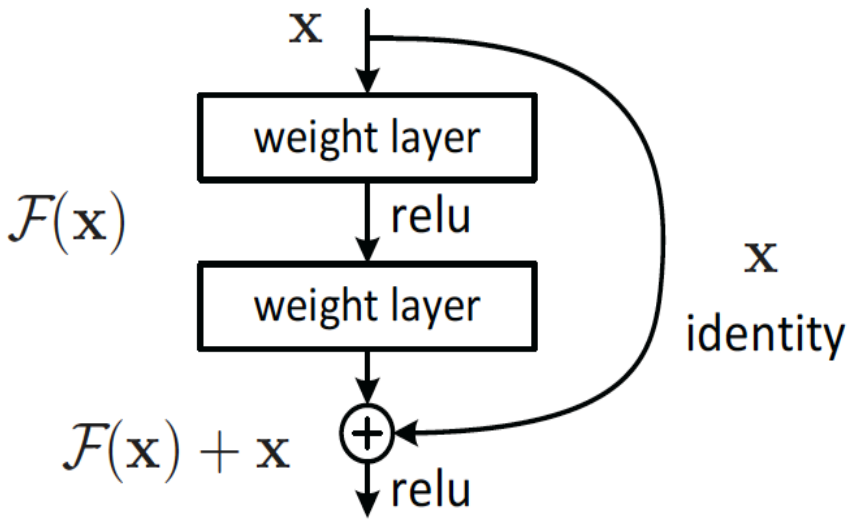


Figure 2.4: A Residual Learning Block
In Residual learning, each identity block tries to learn a small change $F_{(x)}$ and adds it to the input x to get a slightly changed output representation of $H_{(x)}$.
Image source: Kaiming et al.,2015 [35]

plished by skip-connections between one or more identity blocks in a neural network. Skip-connections were initially introduced as a part of *Highway Networks* [69]. Theoretically, the skip-connections this could allow a large portion of information in the input layers to reach the output layers without degradation. The short-cut connections in residual blocks resemble the ones used in highway networks but does not contain the gated units which act as adaptive mechanism determining if the information should be passed through the short connection or not based on the data. In short, short-cut connections in Residual blocks are data-independent, whereas the ones in highway networks are data-dependent.

ResNets are still considered state of the art and used in the practical applications in computer vision for object recognition tasks. Following ResNet, there were a large number of others networks released combining key architectural elements from ResNet and Inception, but they are beyond the scope of this thesis. Moreover, this work focuses on three main architectures- VGGNet, Inception and ResNet which are quite diverse in their architectural design and also popular among the researchers.

2.4.2 Training CNNs

Training Convolutional Neural Networks is not easy. The labeled dataset is first split into train, validation, and test set. The pixels of the images are then scaled to have mean zero and variance one. The most important step in the training process is the selection of the learning rate which is initially set at some higher value. The common practice in training deep networks is to reduce the learning rate by half when the learning plateaus (validation loss does not change over a couple of epochs) [32]. During each epoch, we train the CNNs on the training set of images, and validate on the validation set.

Validation set should not be confused with cross-validation. In Convolutional Neural Networks we mostly perform hyperparameter tuning on the validation dataset. The entire data available is divided into training, validation, and test sets. The validation set is used only for hyperparameter tuning, and not used as a part of training. However, in cross-validation, the training data itself is divided into k partitions. A machine learning model would be trained on k-1 partitions, and the left out partition is used as validation set. Then, a different partition of the training data could be left out and used for validation. This process is repeated till all the folds are included in the training in the leave-one-out fashion. This process of leaving out a fold, and training on k-1 folds of data is referred to as k-fold cross-validation [70]. Cross-validation method is not recommended for training CNNs due to high computational cost.

Training CNNs often takes many hours or even days. The concept of momentum was introduced to speed up the training process. The main objective of the learning algorithms is to minimize the error (loss) function, and reach the global minimum through repeated iterations of learning. However, in real life scenarios, the error function is not smooth and consists of many local minimum. Sometimes the algorithm could get stuck at local minimum regions of higher loss. The momentum term increases the step size of the learning iterations of the algorithm and helps the algorithm to converge faster to an ideal local minima of minimal loss. Momentum also ensures that the gradients move towards the bottom of the bowl shaped error function without much zig-zagging which is generally the case with stochastic gradient descent.

Adaptive learning algorithms such as RMSProp, and Adam [41] automatically adapts the learning rate based on the gradients calculated during every batch of training. These learning algorithms keep track of gradients calculated over multiple

batches, and scales the learning rate for current update. The learning rate is reduced when the gradient is very large and vice versa.

2.4.3 Understanding and Visualization of CNN

CNNs have the potential to change the way machines see our world. However, they are currently black-box, and we don't understand how they come up with their decisions. Their widespread adoption into fields such as diagnostic medicine, space exploration, autonomous driving machines etc. have made them part of our every day human life. Therefore, it becomes essential that we come up with methodologies to unravel the secrets behind their remarkable learning abilities. CNNs designed today have hundreds of hidden layers and millions of parameters, making them even more complex models. The black-box nature of CNNs has been expressed as a major concern by both researchers and early adopters involved in the field of computer vision.

Researchers have adopted several approaches to improve our understanding of convolutional nets. The most common techniques include visualization of activations of the network or visualization of weights, especially in the convolutional layers. Zeiler et al., 2011 proposed Deconvolutional Network (Deconv) which approximates the input signal at pixel level by approximating the feature activations in the convolutional blocks [79]. Zeiler and Fergus used the Deconv net to deconstruct and improve *AlexNet* through visualization of activations for an input image [80]. They found properties such as increasing invariance and class discrimination as we moved deeper through the CNN layers. They also occluded parts of images using a mask and studied the class probability predicted by the network. Another simplified approach is to make a small perturbation to the input image such as blurring and to understand the response of the network [28]. These techniques help to identify parts of the image that contributes to the classification decisions made by the network.

Another approach is to feed a convolutional net with a large number of images and then identify images which maximally activates a neuron. This helps us to understand and visualize the features that a single neuron might be looking for in its receptive field [30]. We could also extract high dimensional feature embeddings from the last hidden layer of a CNN and then use dimensionality reduction techniques such as t-SNE to get a 2-dimensional vector for each image [46]. These images can then be plotted on a grid or clustered using unsupervised approaches to visualize them.

The challenges in the visualization of CNN is mainly due to the fact that most

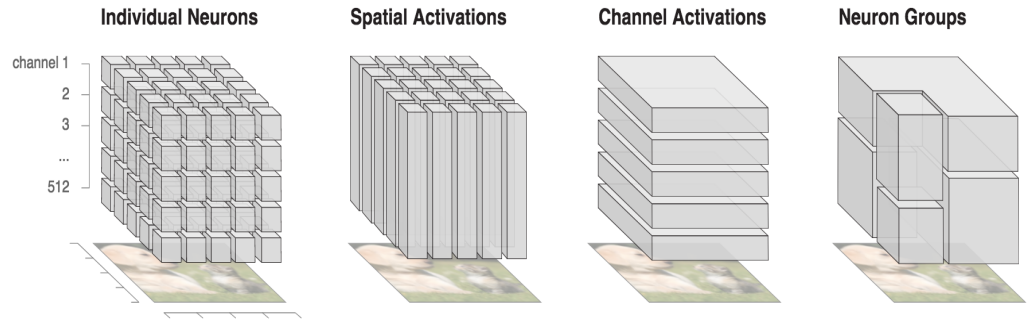


Figure 2.5: Complexity in CNN Visualizations
For each hidden layer in CNN, it is possible to study activations of individual neurons- receptive field of a neuron, spatial activations- spatial positions of certain attributes in an image eg: *wheels of a car*, channel activations- The contribution of a single feature map to the output classifications or even group of neurons- Neurons that activate together for some higher abstract concept
Image source: Olah et al.,2018 [58].

modern architectures have a very large number of neurons, features maps and hidden layers. For each hidden layer, it is possible to study individual neurons, a group of neurons, spatial activations, channel activations or even entire layer (see figure: 2.5). Therefore, it becomes challenging to bring down the visualization of CNNs to a human-scale. In short, we need to find more efficient and straightforward methods to understand these networks.

2.4.4 Study of Semantics in CNNs

Recently there has been some interest among the computer vision community to study semantic representations instead of the internal feature representations in CNN. Gonzalez-Garcia et al., 2017 [31] studied emergence of semantics in *AlexNet* using PASCAL-Part dataset [13]. The PASCAL-Part dataset is a subset of PASCAL VOC, 2010 dataset [20] with bounding boxes separating semantic areas in images. They created visualizations for filter activations in the L-5 layer of AlexNet (256 filters) for 16 object classes. The human annotators were asked to label all the 256 filter activations as *semantic part or not* for every object category. They found that filter activations corresponding to semantic areas only contributed to 7% of the total filter activations.

However, studying semantics of a particular layer of a network may not offer

us more insights as compared to studying the growth of semantic representation as a function of depth of the network. Moreover, use of human annotators to label and identify filter activations corresponding to semantics parts of an image is not feasible for larger networks. The process could become tedious and time-consuming. PASCAL VOC, 2010 dataset has only 20 object classes compared to 1000 classes in ImageNet. This further strengthens our argument about the need for a simple and efficient method to study semantic representation in CNN.

Computational models such as Convolutional Neural Networks could also be used to study visual feature representation in the human brain. Visual features extracted from CNNs have been shown to correlate with fMRI patterns of participants viewing visuals of objects [37]. The features extracted from these models have also been able to predict fMRI activities of unseen object classes (Concept class the CNN was not trained initially to predict). Another work by Wen et al., 2017 could predict fMRI activities corresponding to humans watching movies using features extracted from ConvNet layers [76]. This is despite the fact that fMRI does not capture any temporal properties of brain activations.

Cichy et al., 2016 found similarities between voxel-level activations in dorsal and ventral regions in the human brain and the hierarchical features extracted from layers of CNN using fMRI and MEG data [15]. The authors concluded that initial layers of CNN showed similarities to the occipital lobe (low and mid-level visual region) and for deeper layers of CNN, the similarities corresponded to the dorsal and ventral regions in the anterior brain. In short, layers of these networks could be mapped back to regions in the human brain.

Despite all these remarkable results, brain data is prohibitively expensive and time consuming to collect, process and analyze. Most of the studies comparing CNN to brain explored simple networks such as AlexNet (just eight layers) as compared to deeper networks such as ResNet or VGGNet. Moreover, those studies incorporated very few coarse concepts (usually up to 20 concepts classes) compared to 1000 classes in ImageNet that includes fine-grained concepts such as different species of dogs for a single coarse concept class *dog*. These limitations reiterate the need for a simple methodology to study semantic representations in CNN.

This thesis proposes to use Distributional Semantic models such as Skip-Gram, Glove etc. that are trained on text corpora to study the convergence of semantic representations through the depth of a CNN architecture. These DS models usually have ample coverage for concepts in ImageNet and are available as pre-trained text

files. Moreover, our methodology described in *Chapter 4* generalize very well to study CNN architectures of varying complexity and depth. We also present the details of our experiments with three diverse networks (VGGNet, Inception-v3 and ResNet) in our *Chapter 4*.

2.5 Summary

This chapter discussed in detail about related works in the field of Computational Linguistics focused towards the study of semantics in brain, text and images. We introduced Distributional Semantic models trained on text corpora and discussed some of the most popular DS models. A discussion on various methods to evaluate and benchmark word vectors was done. We then discussed in brief about convolutional neural networks and problems associated with their black-box nature. Some of the existing methods to study CNN architectures were also discussed in this chapter. In the next chapter, we discuss in detail on BrainBench, highlighting our contributions toward improving the tool along with results and discussions on our experiments with various brain datasets.

Chapter 3

Evaluation of Word Vectors using BrainBench V2.0

The previous chapter discussed in detail about major contributions made in the field of study of representations of language. It also explained in depth about various Distributional Semantic models (DSM) or word vectors trained on text corpora and how Computational linguists use these models to study semantic representation in the human brain. The previous chapter also discussed *BrainBench*- a system designed to test, evaluate and benchmark word vector models using brain data [34]. *BrainBench* reported comparable performance to other systems which evaluate and benchmark DSM's.

However, BrainBench tests include just 60 concrete nouns. Another limitation is that the tests do not include any abstract nouns. Moreover, the tests are derived from only two dataset sources (a fMRI and a MEG) and based on one language (English). Considering that DSM's are available in multiple languages and not including brain data from other language sources as a part of BrainBench tests is a limitation.

To address these limitations, we release the second iteration of BrainBench (V2.0) which introduces two new datasets (a fMRI and an EEG dataset collected from Italian participants) to BrainBench tests. This addition improves the coverage of the tests from 60 words to 190 words. The Italian fMRI introduces abstract nouns to the test suite. We also evaluate the performance of word vectors trained on non-English corpora using our Italian brain dataset. Then we compare the performance of word vectors on abstract nouns and concrete nouns separately.

Traditionally, EEG data was not recommended for studying semantics in Brain

due to its weak Signal to Noise ratio (SNR). However, the work of Murphy et al. [54] provided strong argument that EEG dataset along with semantic models is suitable for studying semantics in the brain. The results of our experiments with EEG dataset further reinforces this argument. EEG has higher temporal resolution and makes it ideal for exploring word comprehension and semantic representation in the brain. EEG is more portable and much cheaper as compared to fMRI, making it suitable for experiments.

Anderson et al. studied the performance of the performance of image-based semantic models against anatomical regions in human brain [2]. However, the semantic models and the methodology used in our study are different to their study. We incorporate into BrainBench the ability to study word vector performance across the various anatomical region in the human brain. The methodology is discussed in detail in the section *Evaluation of DS models against anatomical brain region* of this chapter.

The contributions to BrainBench addressed by this chapter are summarized below (*Contribution A*).

- Introduction of abstract nouns into the BrainBench tests and evaluation of the performance of various word vectors on abstract nouns.
- Addition of Italian Brain data into BrainBench and study of the performance of non-English word vectors.
- Addition of an EEG dataset to BrainBench.
- The study of the performance of word vectors across various brain regions using brain Atlas mapping data collected using fMRI.

3.1 Brain Datasets

In this section, we discuss in detail about the four brain datasets that constitute the BrainBench test suite. We primarily focus on the data collection techniques, concept selection and brain signal preprocessing for each of the below datasets.

3.1.1 English fMRI

The first major work in the study of semantics in the human brain using corpus-based semantic models was conducted by Mitchell et al. in 2008 [52]. Nine right-handed participants were presented with 60 concrete nouns from 12 different semantic categories as listed under Table: 3.1 and their brain signals were recorded using a Siemens Allegra 3.0T MRI scanner. The stimulus was presented in the form of line drawings and label text on the screen, and the participants were asked to imagine properties of the concepts presented. The set of 60 concepts were presented six times in random order resulting in 360 stimuli per participant. All the participants were native English speakers.

fMRI uses strong magnetic fields and radio waves to measure the changes in the blood flow in the brain to detect areas of activity. fMRI records blood flow changes in small 3D volume patches throughout the brain. These 3D volume patches are called as voxels. The number of voxels depends on the shape and size of a person's brain and average there were 20000 voxels per participant in this dataset.

The fMRI signals collected from the study were preprocessing using the Statistical Parametric Mapping software (SPM2), corrected for head motion, linear trends. It was then spatially normalized into Montreal Neurological Institute space (MNI) and resampled to 3mm x 3mmx 6mm voxels. The dataset was then reshaped to words * voxels format (denoted as $w * v$) and used for BrainBench tests.

3.1.2 English MEG

In 2012, Sudre et al. [71] investigated the flow of perceptual and semantic information in the brain by studying neural activities captured using Magnetoencephalography (MEG). MEG is a neuroimaging technique which captures the magnetic field changes produced by electric currents in the human brain. MEG has high temporal resolution and could be used to study changes in brain activity over time.

Nine right-handed participants were asked to answer 20 questions about 60 concrete concepts presented as line drawings on the screen. These concepts were the same as the ones studied by Mitchell et al. [52] and listed under Table: 3.1. In their experiment, a question was presented first to the participants followed by the 60 concrete concepts presented in random order. The participant responded with a *yes or no* for each concrete noun. The experiments were repeated for a total of 20 questions resulting in 20 presentations per concept. The questions were on some properties of

Category	Word 1	Word 2	Word 3	Word 4	Word 5
animals	bear	cat	cow	dog	horse
body parts	arm	eye	foot	hand	leg
buildings	apartment	barn	church	house	igloo
building parts	arch	chimney	closet	door	window
clothing	coat	dress	pants	shirt	skirt
furniture	bed	chair	desk	dresser	table
insects	ant	bee	beetle	butterfly	fly
kitchen utensils	bottle	cup	glass	knife	spoon
man made objects	bell	key	refrigerator	telephone	watch
tools	chisel	hammer	pliers	saw	screwdriver
vegetables	carrot	celery	corn	lettuce	tomato
vehicles	airplane	bicycle	car	train	truck

Table 3.1: The concepts studied using fMRI and MEG technology. The concepts studied by Mitchell et al. [52] and Sudre et al. [71] using their experiments with fMRI and MEG respectively.

the concept such as *was it alive?, Can you pick it up?* etc. All the participants were native English speakers.

The MEG data were recorded using an Elekta Neuromag device which has a total of 306 channels. The sampling was done at 200Hz, and MEG recording for each concept was 800ms long. The recorded data were preprocessed using Signal Space Separation method (SSS) [74] and low-pass filtered to 50Hz to remove line noise. The artifacts due to head movement, eye movements and blinking, and MEG sensor failures were subsequently removed using Signal Space Projection method [75]. The data were then reshaped to $w * s * t$ format (words * sensors * time). Each data point in the matrix $s * t$ depicts the electrical activity collected from a sensor s_i at time t_j. For simplicity, we would continue to call these data points as voxels throughout this thesis. The data is then mean normalized and used for our experiments.

3.1.3 Italian fMRI

This dataset was released as part of the experiments conducted by Anderson et al. to study the varying degree of concreteness in taxonomic categories in human brain using fMRI [3]. Brain signal was collected from nine native Italian speakers viewing 70 concepts on a screen and imagining it. The 70 concepts selected were from two domains *music* and *law*. Moreover, these 70 concepts were organized into 7 Taxonomic

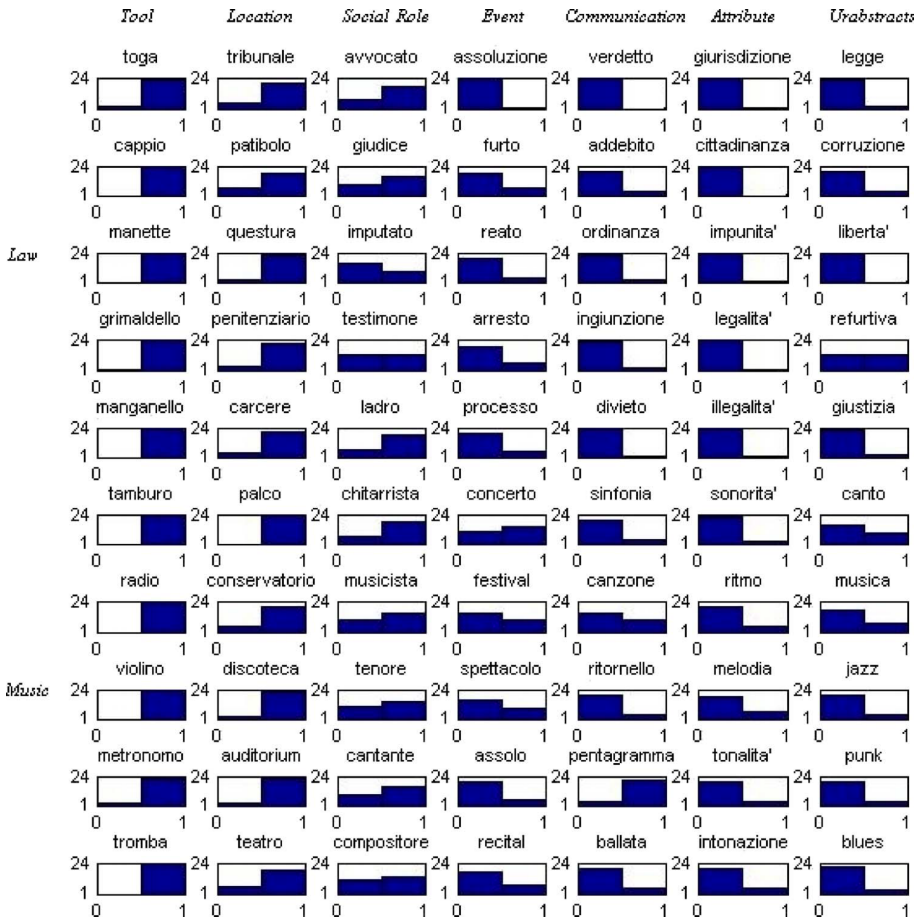


Figure 3.1: Word Norming Results for Concepts in Italian fMRI
70 concepts above organized into 7 Taxonomic categories and 2 Domains (music and law). The concreteness rating was assigned using word norming technique [8] where 24 native Italian speakers were asked to rate the 70 concepts on a concreteness scale of 1 (highly abstract) to 7 (highly concrete). The collected data was binarized as concrete =1 and abstract =0. The y-axis indicates the rating of each of 24 participants.
Image source: Anderson et al.,2014 [3].

categories (Attribute, Communication, Event, Social Role, Tool, Location and Urabstracts) using MultiWordNet [61] (the Italian version of WordNet [24]) resulting in 70 stimuli with ten concepts from each taxonomic categories. Music and law domains had five words each within each taxonomic category. The concept selection was made in such a way that there was a varying degree of concreteness with the tool being the most concrete and Urabstracts being the most abstract taxonomic category.

The concreteness rating was assigned using word norming technique [8] where 24 native Italian speakers were asked to rate the 70 concepts on a concreteness scale

of 1 (highly abstract) to 7 (highly concrete). The collected data was binarized as concrete =1 and abstract =0. The concepts and its concreteness are summarized by the Figure: 3.1.

The brain signal was collected by Bruker MedSpec MRI scanner at the neuroimaging labs, University of Trento. The name of the 70 concepts was presented to the participants in written form on the screen, and the process repeated five times resulting in 350 stimuli per participant. The whole process was divided into five sessions of 70 concepts, and the order of concepts shown to the participants was randomized in each session. For each concept, The participants were asked to think about the role these concepts played in different situations.

The data collected was then corrected for head movement and spatially normalized to the Montreal Neurological Institute (MNI) template and resampled to voxel dimensions of 3mm * 3mm *5mm which represented a 3D volume in the brain. The voxels were then mean normalized to make the mean zero and standard deviation one. The preprocessed dataset is then reshaped to $w * v$ format, where w is the number of words (350) and v is the number of voxels. The Italian words were then translated into English concepts using Google Translate and used for our experiments with various DS models. It should be noted that this dataset is referenced as *"Italian fMRI"* throughout this thesis since the concepts presented to the participants were initially in Italian.

3.1.4 Italian EEG

The EEG dataset used in BrainBench was released as a part of behavioural experiments conducted by Murphy et al. at University of Trento [54]. The brain data was collected from 7 college educated native Italian speakers who were presented with photographic images of concepts belonging to two semantic categories (Mammals and Tools), and their brain activity was recorded using an EEG device. The concepts were presented to the participants as photographs (contrast normalized greyscale photographs) on a screen, and they were asked to assign a label to the concept. Here the label is of the actual concept rather than the semantic category. The 30 concepts from each category were presented to the participants in six times in random order resulting in 360 image presentations. The post-experiment survey conducted after the study determined that the participants agreed to almost 90% of the labels that were assigned to the images. The list of concepts presented to the participants are

MAMMALS		
Anteater	Elephant	Ilama
Armadillo	Fox	Mole
Badger	Giraffe	Monkey
Beaver	Gorilla	Mouse
Bison	Hare	Otter
Boar	Hedgehog	Panda
Camel	Hippopotamus	Rhinoceros
Chamois	Ibex	Skunk
Chimpanzee	Kangaroo	Squirrel
Deer	Koala	Zebra

Tools		
Allen key	Mallet	Power drill
Axe	Nail	Rake
Chainsaw	Paint brush	Saw
Craft knife	Paint roller	Scissors
Crowbar	Pen knife	Scraper
File	Pick axe	Screw
Garden fork	Plaster trowel	Screwdriver
Garden trowel	Pliers	Sickle
Hacksaw	Plunger	Spanner
Hammer	Pneumatic drill	Tape measure

Table 3.2: The Concept list for EEG dataset

shown under Table: 3.2. EEG records electrical activity using electrodes placed along the scalp. EEG measures fluctuations in voltage over time in the brain resulting from the neurons firing due to the presence of stimuli. EEG signals have poor signal to noise ratio but offer high Temporal Resolution (TR) as compared to fMRI. TR refers to the precision of a measurement over time. The EEG Signals were recorded using 64 electrodes placed at the various location on the scalp and were recorded at 500Hz. The collected signals were pre-processed using the EEGLAB package [18] and band-pass filtered at 1-50Hz to remove high-frequency noise. The signals were then downsampled to 120Hz. The signal components related to the eye movement were manually removed after ICA decomposition [48]. The pre-processed brain signal in time domain was reshaped to words * sensors * time format and then mean normalized.

3.2 Distributional Semantic Models (DS)

We evaluate six popular Distributional Semantic models as a part of our analysis. A detailed explanation of these models could be found in chapter 2 of this thesis.

Word2Vec and Skip-gram:

The Word2vec model, is probably the most widely used model in NLP tasks. It was proposed by Mikolov et al. in 2013 and uses a shallow neural network to learn the embedding space [50]. The Skip-gram model trained on Google news dataset is a 300-dimensional vector. We use the pre-trained Skip-gram word vector for our analysis.

Glove:

Glove is a regression-based semantic model published by Pennington et al. in 2014 [60]. It introduces the concept of representing the relationship between two word as their co-occurrence probabilities. This 300-dimensional vector model was trained on a combined corpus of Wikipedia and Gigaword 5.

Cross-lingual:

This model, proposed by Faruqui and Dyer, 2014, takes into account semantic properties across various languages [21]. It was trained using both German and English words using WMT-2011 corpus and uses a shared semantic space to learn the word embeddings. The resulting word vector have 512 dimensions.

RNN:

A Recurrent Neural Network (RNN) trained to predict the next word in the sequence (Mikolov et al., 2011) [51]. This model trained on broadcast news transcriptions have a dimension of 640 for its word embeddings.

Global Context:

This model takes into account both local, and global context of a document to learn the semantics of the word (Huang et al.;2012) [38]. This model could encode into the word vectors properties such as homonymy and polysemy of a word. This model trained on *Wikipedia* have vectors of 50 dimensions per word.

Non-Distributional:

This semantic model was created by combining various lexical sources such as WordNet (Fellbaum,1998) [24] and FrameNet (Baker,1998) [7] by Faruqui et al. [22]. The words modelled by this semantic model are extremely sparse and high dimensional (171,839 dimensions).

3.3 Methodology

In this section, we discuss the methodology used in the BrainBench test suite which follows a similar method to the original paper by Xu et al. [34]. The four datasets described in the section *Brain Datasets* are added to the BrainBench test suite following the methodology described in this section. The five Distributional Semantic models (DS) evaluated by this methodology is detailed in the section *Distributional Semantic Models (DS)*. The entire process is summarized in the Figure: 3.2.

Brain data collected using multiple imaging technologies such as fMRI, MEG and EEG are preprocessed before they are added to the BrainBench test suite. The preprocessing depends upon the type of dataset and is different for fMRI, MEG, and EEG. This is explained in detail in the section *Brain Datasets*. After preprocessing, a small portion of the brain signals collected using imaging technologies could be correlated with the visual features captured during the experimental trials. These low-level features might include the number of white pixels on the screen, the length of the words presented to the participant as a part of the experiment, features derived from drawing lines on the screen [71], etc. These visual features act as confounding variables and should be removed from the brain imaging data before they could be used in our experiments.

Removing Visual Features using Linear Regression

The visual features are removed by training a linear regression model which predicts a value per voxel corresponding to the visual features in the brain signal. These learned visual features are subtracted from the original brain signal. A linear regression model tries to fit a straight line for a set of data points in such a way that the sum of squared errors is minimal [Figure: 3.3]. We learn the weight matrix ($\mathbf{w}$) which parameterizes the best fit line for every data point $(\mathbf{x}_i, y_i)$. Let us define the error e_i as the distance between the true and predicted values such that $e_i = y_i - \mathbf{w}^T\mathbf{x}_i$. The objective

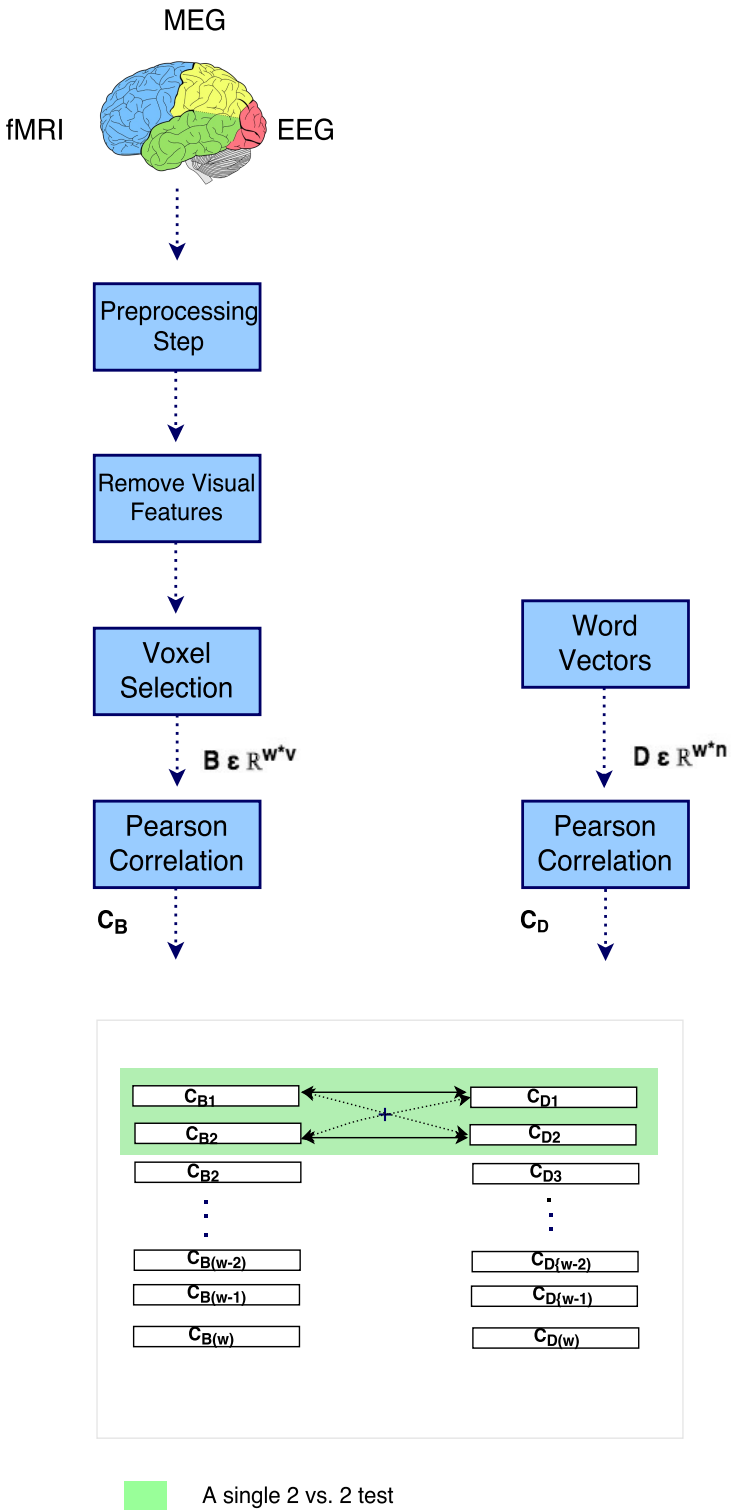


Figure 3.2: The methodology for BrainBench test suite

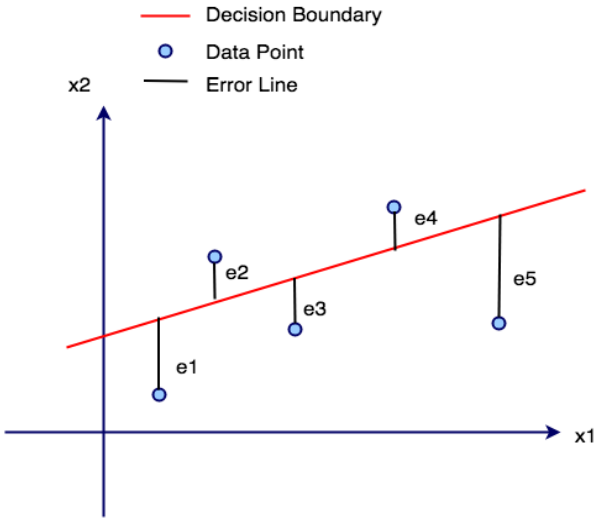


Figure 3.3: A concept diagram for Linear Regression
A Linear Regression model tries to fit a straight line for a set of data points in such a way that the sum of squared errors is minimal.

function of the linear regression model is to minimize the the sum of squared errors such that $\sum e_i^2 = \|X\mathbf{w} - \mathbf{y}\|^2$, where ($\mathbf{X}$) in our context are the visual features for each trial in our experiment and $\mathbf{y}$ is the brain signal.

Sometimes the visual features might be correlated with one another. An example could be the length of words could be correlated with the number of black pixels on the screens used to represent the word. This can result in the weight matrix ($\mathbf{w}$) being poorly determined and could result in over-fitting. This collinearity between visual features could cause the $\mathbf{w}$ to have substantial updates for a particular training sample and therefore may not generalize across all other samples in the data. Therefore, we introduce weight decay to restrict the updates to the weight matrix ($\mathbf{w}$) in the presence of very large values in the feature matrix ($\mathbf{X}$). This is also known regularization. The objective function of a linear regression model with regularization is given as

$$\min_{\mathbf{w}} \|X\mathbf{w} - y\|^2 + \lambda \|\mathbf{w}\|^2$$

Where λ is a hyper-parameter. The regularization followed here is also knows as L2 regularization. The L2 regression performed here has a closed form solution. For each voxel in the brain signal, the regression model predicts a value as a function

of the visual features, and this predicted value is then subtracted from the original voxel value. This method is also termed as **"partialling out"** an effect. It should be noted that the only visual features removed from the Italian fMRI and EEG data were the length of words.

Voxel Selection

The brain data after the removal of visual features still contains noise. We followed the same methodology used by Mitchell et al. [52] to select the most stable voxels. In this process, we choose only those voxels which show strong self-correlation across various trials of the same word. The voxels which have such strong self-correlations would have a high stability score. Here, we assume that any voxel with a low stability score could be noise and is not considered for our experiments. Roughly, around 500 voxels from each dataset were selected and used in our study.

Pearson Correlation

The brain data corresponding to repetitions of the same word are then averaged for each participant resulting in a matrix $B \in \mathbb{R}^{w*v}$, where w is the number of words, and v is the number of voxels selected. We then calculate the Pearson correlation of every row in brain data B with every other row resulting in the brain correlation matrix (C_B) where $C_B \in \mathbb{R}^{w*w}$. Each row C_{B_i} of the matrix C_B represents the correlation of the word i with every other word from 1 to w in the brain imaging dataset.

The same approach is followed for the DS models. From each DS model, we extract the words that are present in the model and the brain datasets resulting in the matrix $D \in \mathbb{R}^{w*n}$, where w is the number of words and n is the number of dimensions of the word vector. We then compute the Pearson correlation of every word in word vector matrix D with every other word in the matrix resulting in the correlation matrix C_D where $C_D \in \mathbb{R}^{w*w}$.

The similarity between C_B and C_D could be studied using the 2 vs. 2 test which is described below.

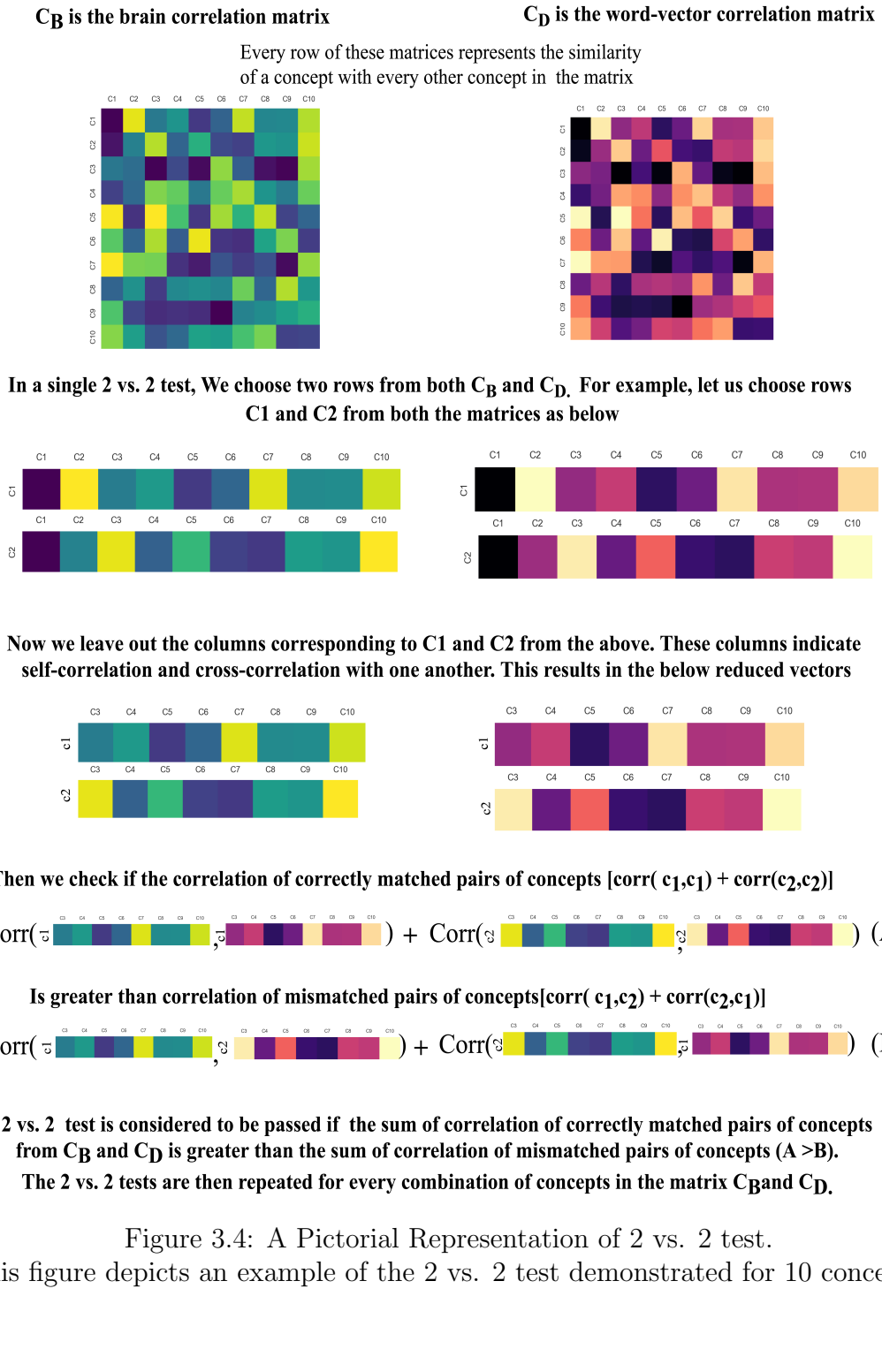


Figure 3.4: A Pictorial Representation of 2 vs. 2 test.
This figure depicts an example of the 2 vs. 2 test demonstrated for 10 concepts.

Now we have created two correlation matrices C_B and C_D representing similarity of words in different vector spaces. We could use the Representational Similarity Analysis (RSA) method to compute the correlation between matrices C_B and C_D [42]. RSA is a simple method adopted from the field of neuroscience to study the relationship between two vector spaces. However, the RSA has a disadvantage that it produces one aggregate score for the relationship between two matrices. As we will see, it is sometimes essential to study and understand the parts of the correlation matrix which may result in a high or low score.

2 vs. 2 Tests

Instead of measuring the correlation between the matrices C_B and C_D using RSA, we perform the 2 vs. 2 test derived from the early works of Mitchell et al. searching for word embeddings in the brain [52]. This methodology introduced by Anderson et al. [4] and Xu et al. [34] could be considered as an extension of RSA with the 2 vs. 2 test. In 2 vs. 2, we select the rows corresponding to two concepts (c_1 and c_2) from our correlation matrices C_B and C_D. We then omit the columns corresponding to two concepts resulting in a reduced vector with $w - 2$ columns from both the matrices where w is the total number of concepts. These vectors represent the similarity of the two concepts with every other concept, except for their self correlation and their correlation with one another. Let us call the reduced vectors as C_{B_1}, C_{B_2} from correlation matrix C_B and C_{D_1}, C_{D_2} from correlation matrix C_D. The correlation of the concepts c_1 and c_2 from C_B and C_D are then computed to check if the correlation of the correctly matched pairs:

$$corr(C_{B_1}, C_{D_1}) + corr(C_{B_2}, C_{D_2})$$

is greater than the correlation of the mismatched pairs:

$$corr(C_{B_1}, C_{D_2}) + corr(C_{B_2}, C_{D_1})$$

A 2 vs. 2 test is considered to be passed if the correlation of the matched pairs is greater than the correlation of the mismatched pairs. The test is repeated for all possible pairs of concepts in our dataset. This results in ${}^{w}Choose_2$ tests for a dataset, where w is the number of rows in the correlation matrix. The 2 vs. 2 accuracies is the percentage of the number of 2 vs. 2 tests passed to the total number of 2 vs. 2 tests. since this is a binary classification task, the chance accuracy is 50%. A pictorial

representation of the test is described by Figure: 3.4.

The methodology described in this section (Figure: 3.2) varies from the method described by Xu et al. [34]. We remove visual features before voxel selection as compared to the Xu et al. where voxel selection was performed before removing visual features from the brain signal. We believe that, if voxel selection is performed without removing visual features from the signal, the most stable voxels selected could be the ones which are correlated with visual features rather than semantics.

Statistical Significance Tests

The 2 vs. 2 test was designed to study and compare the relationship between concepts in different vector spaces. The chance accuracy is 50%. In statistics, a null hypothesis is a hypothesis that there exists no relationship between the variables that we are trying to compare. In our tests, the null hypothesis is that there is no similarity between correlation concepts in brain data and word-vector space. Let us define the level of significance (α) as the probability of rejecting the null hypothesis and p-value (p) as the probability of our tests returning extreme values, both assuming that the null hypothesis is true. Therefore, for us to reject the null hypothesis and to prove concretely that our results are statistically significant, we look for $p < \alpha$

We conduct 1000 permutation tests by randomly shuffling the assignment of word identity to word-vector, recomputing the C_D, and re-running the 2 vs. 2 tests for each permutation. After the permutation tests, the p-value is calculated as

$$p = R/T$$

where R is the number of random permutation accuracies which is greater than or equal to the observed accuracy of our tests and T is the number of random permutations conducted. For example, if the 2 vs. 2 accuracy is 0.63 and 1000 random permutations tests returned 14 2 vs. 2 accuracy values greater than our observed accuracy, then the p-value for this experiment is 0.014 (R=14 and T=1000). The statistical significance depends heavily on the level of significance (α) which is usually fixed at 5%. In our example here $p < \alpha$ ($0.014 < 0.050$) and therefore, we can reject the null hypothesis that there is no correlation between the two vector spaces.

3.4 Evaluation of DS models against anatomical brain region

Anderson et al.,2015 performed exploratory analysis studying the correlation of both image and corpora based features with various Regions of Interest (ROI) in human brain [5]. They also provided evidence that for a given concept, text-based DS models showed similarity in fMRI scans in ROI's related to linguistic processing. Similarly, image-based models showed similarity with ROI related to visual processing in the human brain. We incorporate into BrainBench the ability to evaluate DS models against various ROI in human brain using the same fMRI data collected by Mitchell et al., 2008 [52]

The size and shape of brain vary across different individuals. It, therefore, becomes essential to spatially normalize fMRI data from various participants in the study such that a location in one person's brains scan corresponds to the same location in other persons brain scan. The fMRI data after pre-processing was spatially normalized into Montreal Neurological Institute space (MNI) and resampled to 3mm x 3mmx 6mm voxels. MNI is a template which divides the brain into various regions of interest. Each voxel in the fMRI data is associated with a spatial position involving 3D coordinates (x, y, z) in the human brain. The voxel coordinates could be converted into MNI coordinates using the SPM software in Matlab to map a single voxel to an ROI in the brain.

After regressing out the visual features from the brain data, we extracted the voxels corresponding to various ROIs in the brain for each of 60 concepts. This results in $60 * v$ matrix for every anatomical region that we were included in our experiments. Here v is again the number of voxels. We ignore regions with $v < 200$ in our fMRI data across all 9 participants. It should also be noted that voxel selections were not performed as a part of this study since the number of available voxels per region was limited. The experiments then followed the process explained under the section 3.3. The 2 vs. 2 tests were conducted for every anatomical ROIs in the brain which satisfied the condition of a minimum of 200 voxels in all 9 participants. Statistical significance tests were run for all combinations of ROIs and DS models.

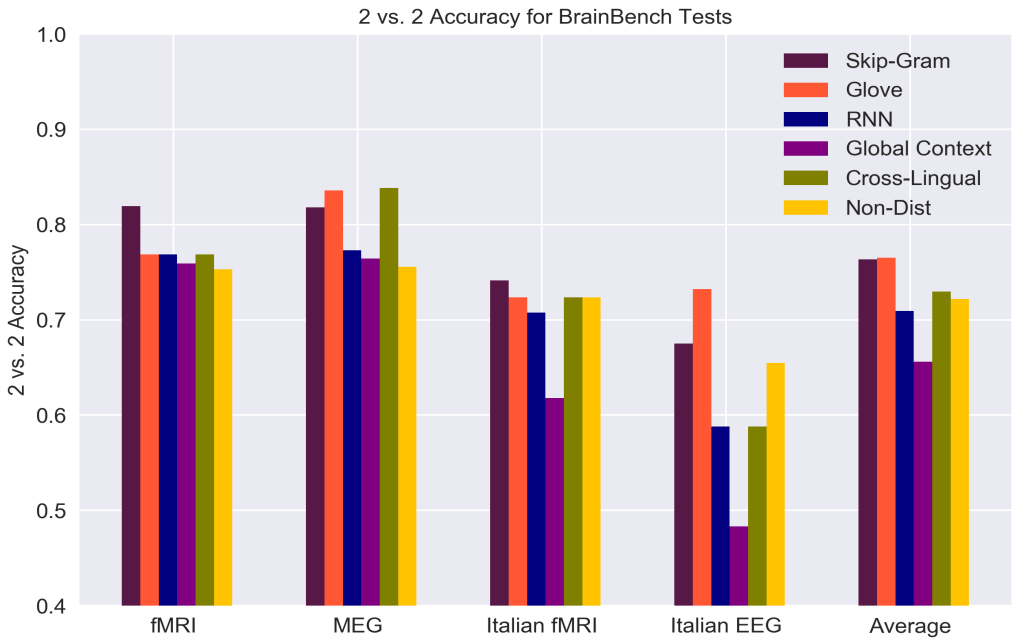


Figure 3.5: Summary of BrainBench Test Results
The fMRI and MEG contains 60 concrete concepts, Italian fMRI contains 70 concepts from two domains, and 7 taxonomic categories with varying degree of concreteness, and finally, the Italian EEG contains 30 concrete concepts from each Mammals, and Tools category.

3.5 Results and Discussions

In this section, we report the results of our experiments involving the evaluation of 6 popular word vector models against 4 different brain datasets using the 2 vs. 2 tests. These results are summarized in Figure: 3.5. The permutations tests were run for all possible pairs of brain datasets and DS models. Except for the 2 vs. 2 accuracy for the Global-Context with the Italian EEG dataset, all other 2 vs. 2 accuracies were found to be statistically significant with $p < 0.01$.

There is a significant variance in the performance of different word vectors on different brain datasets. Skip-Gram and Glove are the best performing DS models across all the 4 datasets whereas Global-Context performs the worst. The Global-Context vectors have the smallest number of dimensions (50) as compared to other word vectors like Skip-gram (300), Glove (300), RNN (640), Cross-Lingual (512) etc. Prior Research by Mikolov et al., 2013 have shown that there is a positive correlation between the number of vector dimensions in a word vector and their corresponding

performance on various downstream NLP tasks [50]. They found that for the Skip-gram model, vectors with dimensions around 300 were most suited for downstream NLP tasks compared to Skip-gram models with 50 dimensions or less.

Global-Context vectors may not also be encoding enough semantic information due to the smaller size of its vectors. Moreover, Global-Context vectors were also shown to perform poorly on other word similarity datasets (Figure: 3.7) which is consistent with the results of BrainBench tests.

The fMRI and MEG dataset in BrainBench has higher accuracies compared to both Italian fMRI and EEG (Figure: 3.5). It should also be noted that performance of Glove is comparatively better in both MEG and EEG dataset. Prior research has shown that word vectors based on word co-occurrences do capture temporal dynamics of semantics [62]. EEG and MEG also incorporate temporal information from the brain signal, and therefore one could speculate that Glove vectors could also be incorporating higher temporal dynamic related to semantics compared to all other word vectors.

3.5.1 Concrete Vs Abstract Nouns

The 2 vs. 2 accuracy for Italian fMRI scores are slightly lower (8-10% for various DS models) compared to the English language based fMRI. It should be noted that the Italian fMRI has a larger distribution of abstract words as compared to English fMRI (only concrete nouns). Anderson et al., 2015 studying the *varying degree of concreteness* through their experiments on the same Italian dataset have shown that decoding semantics from neural signals corresponding abstract nouns are more difficult as compared to concrete nouns [3]. We also attribute our lower 2 vs. 2 accuracy for the Italian fMRI due to the higher distribution of abstract nouns in the dataset. We performed a comparative study of the 2 vs. 2 accuracy for concrete and abstract nouns separately in the Italian fMRI. We selected words from three taxonomic categories (Tool, Location and Social Role) as concrete and other four taxonomic categories (Event, Communication, Attribute, Urabstracts) as abstract based on the Figure: 3.1. This figure represents the taxonomic categories on a binarized concreteness scale with concrete being 1 and abstract being 0. The three taxonomic categories (Tool, Location and Social Role) in the figure are more shifted towards being concrete as compared to other four taxonomic categories. Thus the dataset was divided into 30 concrete and 40 abstract words.

Brain Datasets	Total concepts	Skip-Gram	Glove	RNN	Global Context	Cross-Lingual	Non-Dist
fMRI	60	60	60	60	60	60	60
MEG	60	60	60	60	60	60	60
Italian fMRI	70	66	66	62	65	66	66
Italian EEG	60	43	43	42	43	44	43

Table 3.3: Concept coverage in various DS models

The 2 vs. 2 accuracies for abstract words were at least 10% to 12% lower than the 2 vs. 2 accuracies for concrete words in the datasets. These results are again consistent with the ones reported by Anderson et al., 2015 [3]. Moreover, unlike concrete nouns, *"an individuals experience of abstract nouns may be subjective"* and could vary from person to person [3]. It is also highly unlikely that the corpus-based language models could capture the large variability in the abstract nouns representations arising from subjectivity based on human experience. The performance of various DS models on Italian fMRI dataset is summarized by the Figure: 3.6.

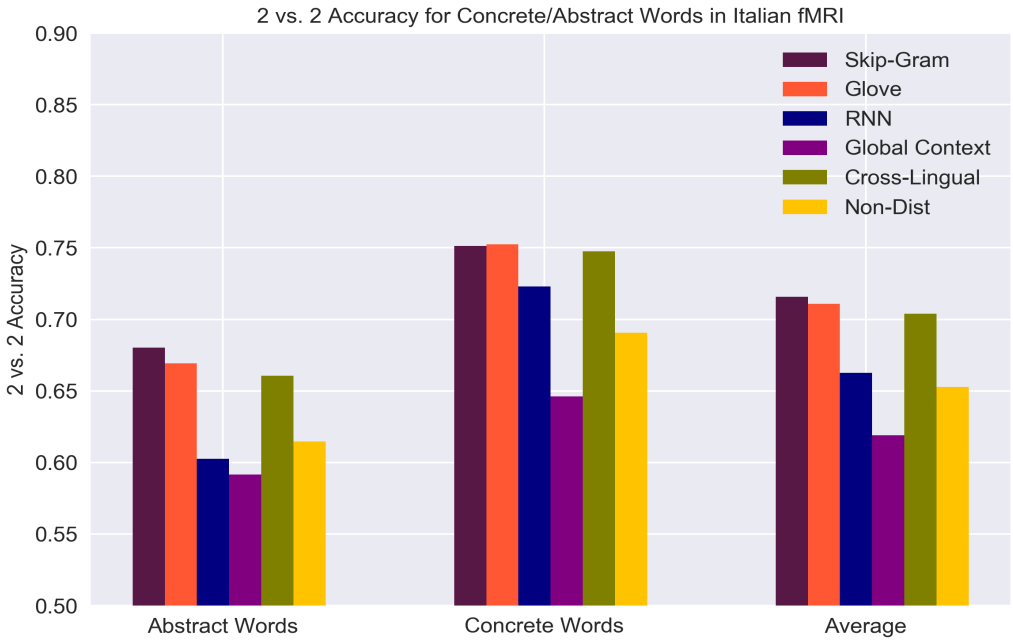


Figure 3.6: Concrete Vs Abstract scores for Italian fMRI

3.5.2 Evaluation of DS models against EEG datasets

EEG datasets are generally considered to be noisy and have poor signal to noise ratio (SNR). The 2 vs. 2 accuracies for various DS models against the Italian EEG dataset is summarized in the Figure: 3.5. Except for Glove word vector all other DS models are having their lowest accuracies against the EEG data as compared to both the fMRI's and MEG data. In fact, 2 vs. 2 accuracy of Global-Context vectors are below the chance accuracy of 50% and did not pass the permutation tests. EEG dataset also had the lowest coverage of concepts in DS models as compared to other brain datasets in BrainBench (Check Table: 3.3). If a concept is not found in a DS model, the 2 vs.2 tests corresponding to that particular concept is omitted.

Despite the lower 2 vs. 2 accuracies, the performance of BrainBench using the EEG dataset is comparable to another other word similarity datasets. EEG data is much cheaper and convenient to collect as compared to fMRI and MEG. Our results with the EEG dataset should encourage the scientific community to built larger EEG based brain test suites to evaluate and benchmark word vectors.

3.5.3 2 vs. 2 test results for Italian Skip-gram

We also download the Italian Skip-gram vectors and conducted the 2 vs. 2 tests against the Italian fMRI. The 2 vs. 2 test accuracy for this particular experiment was found to be 74.1% which is quite similar to the accuracy with English language version of the Skip-gram vector. This result could imply that the BrainBench tests could generalize for testing word vectors modeled from different languages irrespective of the native language of individuals from whom the original brain data were collected.

3.5.4 Comparing BrainBench with other word similarity datasets

We also compared the performance of the BrainBench with four other word similarity datasets. The similarity scores for MEN, SimLex, WS-353-SIM, and WS-353-REL, were obtained from Xu et al., 2016 [34].

Based on the correlation scores, BrainBench performs as good as MEN and WS-353-SIM. BrainBench also performs significantly better than SimLex and WS-353-REL. Both Skip-gram and Glove are the top performing DS models in both MEN and BrainBench. However, the performance of RNN, Global Context and Non-Distributional models are significantly lower in all the similarity datasets as com-

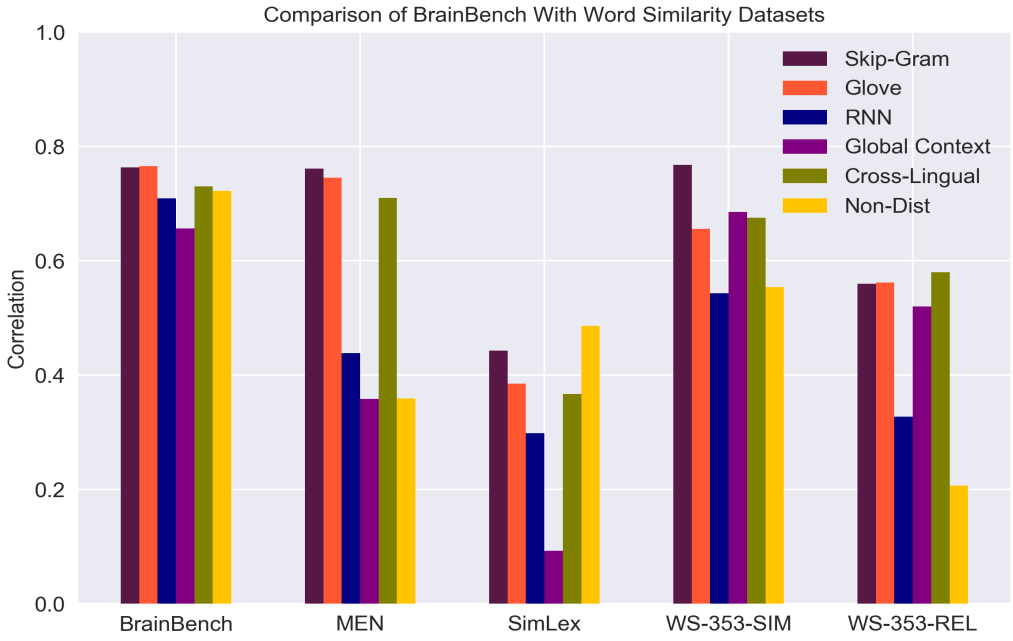


Figure 3.7: Comparison of BrainBench With Other Word Similarity Datasets

pared to BrainBench. Compared to BrainBench, WS-SIM-353 dataset has better performance for Global Context vectors.

The performance of BrainBench differs from previous benchmarks, implying that the BrainBench may test for a kind of semantic information which is not readily available in behavioral data.

3.5.5 2 vs. 2 accuracies for DS models against Anatomical ROIs in human brain

The voxels from the fMRI data collected by Mitchell et al., 2008 were mapped to the anatomical regions of the brain using the MNI template [52]. There were more than 120 regions of interest (ROI) identified in the original fMRI data across all nine participants. However, we included only those regions which had at least 200 voxels in each of the nine participants. 43 such ROIs were filtered out using this criterion. After removing the visual features from the brain data, we extracted the voxels corresponding to these 43 ROIs in the brain for the 60 concepts. This results in $60 * v$ matrix for every anatomical region that we were included in our experiments. Here v is again the number of voxels. We then computed the Pearson correlation to

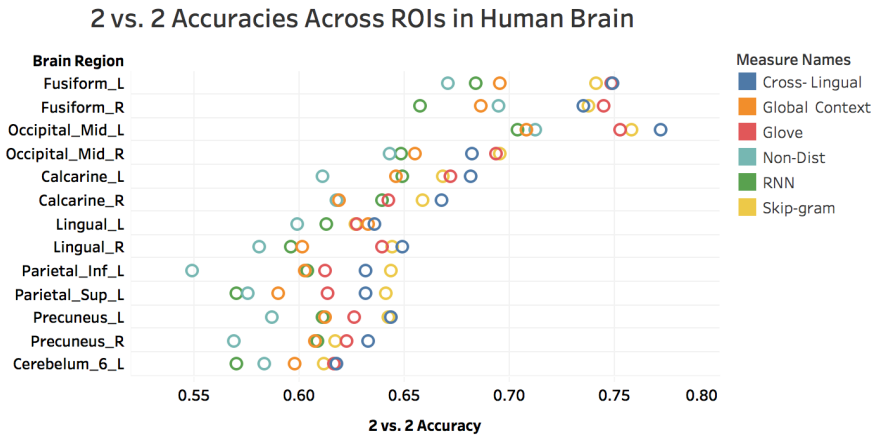


Figure 3.8: Performance of DS models against various ROIs in brain.

generate our brain correlation matrix C_B. Similarly, C_D represents the correlation matrix corresponding to a DS model. Then the 2 vs. 2 tests are conducted for every ROI and DS pairs. This is followed by permutations tests to determine the statistical significance of the tests. We only report 13 top performing ROIs in our results. The results are summarized in Figure: 3.8.

The high performing ROIs included both the fusiform gyrus, and occipital lobe. The fusiform gyrus which forms the part of both occipital and temporal lobe of the brain is the top performing ROI, along with the medial surface of the left occipital lobe. Fusiform gyrus is part of the linguistics processing area of the brain and is known to take part in word recognition (contains word-form area) [56]. Fusiform gyrus is also generally associated with face and body part stimuli. The occipital lobe is the visual processing in the brain and involved in decoding semantics from visual imagery. Skip-gram, Glove and Cross-Lingual performs considerably better as compared to RNN, Non-Distributional and Global-Context in both the fusiform gyrus and the Occipital lobes.

The calcarine sulcus and Lingual regions achieve intermediate performance with an average 2 vs. 2 accuracy of 0.65 for all the 6 word vectors. The calcarine region is associated with the visual processing whereas Lingual regions have been shown to take part in both visual and word processing [2]. The 2 vs. 2 accuracies for all DS models were found to be statistically significant for both these regions ($p < 0.01$).

The inferior and superior left parietal lobe, the left and right precuneus and the left cerebellum regions show the worst performance. In fact, Non-Distributional, Global-context and RNN fail to pass the significance threshold of p=0.05. It should

be noted that these regions are not primarily associated with either linguistic or visual processing in the brain which might explain the poor performance of DS models in these regions. On average the Non-Distributed model which is the worst performing model across all the anatomical brain region. Cross-Lingual, on the other hand, is the best performing DS model. The summary of the performance of all the 6 DS models against all the 43 ROIs could be found in Figure:A.1 in appendix A.

The results described in this section could help the linguistics community studying language and visual representation in the human brain. There seems to be a large variation in the performance of various DS across anatomical brain regions, and our results could be used by researchers as a guideline in selecting the appropriate word vectors to decode semantics in their brain related experiments. For example, a researcher studying visual processing areas in the brain could use Cross-lingual vectors to decode emergence of visual semantics in the brain. Cross-lingual have been shown by our results to perform the best in areas related to visual processing. Similarity Glove could be ideal for studies focused on language centers of the brain.

3.5.6 Comparing BrainBench $v_{1.0}$ vs $v_{2.0}$

The addition of Italian fMRI and EEG dataset is one of the most important contributions of this work. The initial version of BrainBench had only 60 nouns words from English fMRI and MEG brain data sources. We evaluated the Italian Skip-gram word vector against the Italian fMRI dataset and found performance comparable to its English version. The results could imply that brain datasets could be used to evaluate word vectors trained from text corpora of different languages and is independent of the native language of the participants from whom the brain data was collected. Moreover, the Italian fMRI dataset added to BrainBench $V_{2.0}$ has a small coverage of abstract words (40 words) as compared to none in its first iteration.

Prior research has shown that the performance of abstract concepts on semantic tasks is lower than concrete concepts [3]. Despite of the newly added abstract words in BrainBench $V_{2.0}$, the performance of our tool is comparable to other word similarity task datasets.

The fMRI and MEG dataset were part of the first iteration of BrainBench. The reason that we decided to include these datasets is that we have changed the methodology associated with the BrainBench. In the first iteration by Xu et al., 2016 voxel selection was performed before removing visual features from the Brain datasets. In

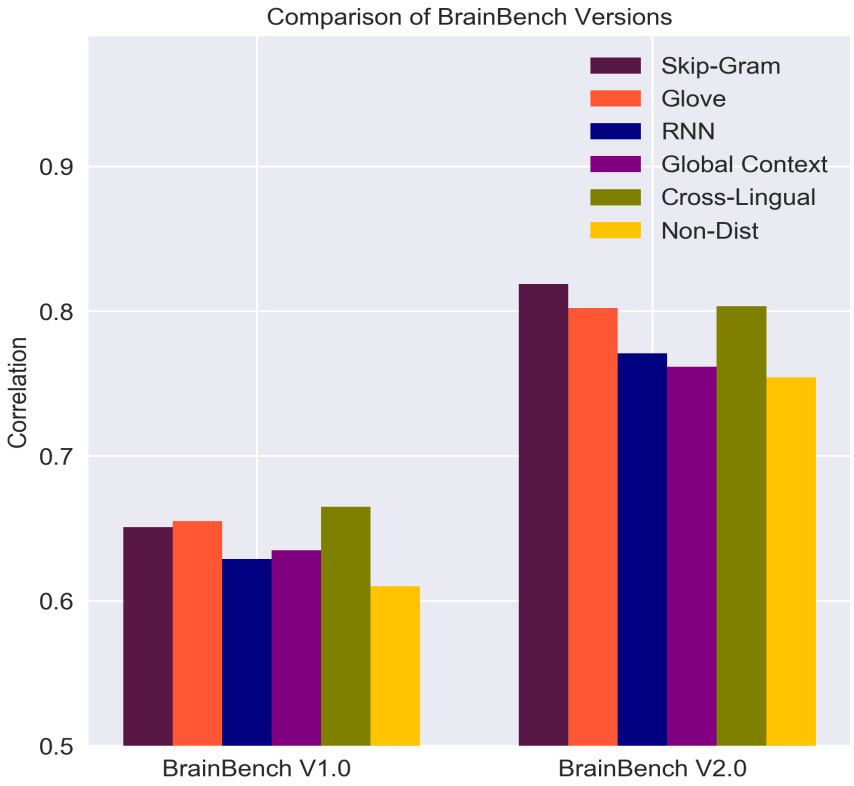


Figure 3.9: Comparison of BrainBench $v_{1.0}$ vs $v_{2.0}$ Test Results

this work, we have reversed the order with visual features being removed before voxel selection. This small change in methodology does have a significant impact on the results. In Figure: 3.9 we have compared the performance of both the versions of BrainBench. It should be noted that the results are based on the average of fMRI and MEG scores only (EEG and Italian fMRI are not included to keep the comparison fair).

Voxel selection selects the most stable voxels and helps to remove noise from the signal. However, if voxel selection is performed without removing visual features from the signal, the most stable voxels selected could be the ones which are correlated with visual features rather than semantics. Therefore, in our methodology visual features were removed from the signal before voxel selection.

Another contribution of this work is the study of the performance of various DS models against anatomical regions of interest (ROIs) in brain. The voxels corresponding to various anatomical brain regions were isolated, and separate 2 vs. 2 tests were executed against various DS models. We found that Cross-Lingual, Skip-gram and

Glove perform the best across various linguistic and visual processing areas in the human brain.

3.6 Summary

This chapter focused on addressing the contributions made in improving the BrainBench tool designed to evaluate and benchmark word vectors (Contribution A). The preliminaries, which include both the brain datasets and the DS models used in our experiments along with the methodology used, were discussed in this chapter. We also discussed and compared our results with the previous iteration of BrainBench released in 2016. This thesis has extended the coverage of BrainBench from 60 concrete nouns from 2 brain datasets to 190 nouns (30 Abstract) from 4 brain datasets. Some key contributions here include the addition of Italian language datasets which includes an EEG sourced dataset. We also found that DS models show a high correlation to EEG brain signals. We also compared the performance of BrainBench with other word similarity datasets and found comparable performance measures. Finally, we studied the performance of DS models against anatomical brain regions and found that some DS models show correlation with brain regions associated with language and visual processing.

Chapter 4

Semantic Representations in CNNs

Convolutional Neural Networks (CNNs), a type of Artificial Neural Network (ANN) loosely inspired by the human visual cortex, has become the state of the art technology in object recognition from images. Over the last five years, the deep learning community has demonstrated that having deeper networks have a direct impact on the performance of these networks [66, 35, 73]. The trend of going deeper with Convolutional Neural network has lead to the creation of a myriad of black box architectures which works well at the task of object classification. However, there is an apparent lack of understanding on why various CNN architectures perform better than the other.

We propose to study the semantic representations through the hidden layers of various CNN architectures to offer insights on the function and performance of these complicated networks. We propose to use the DS models trained on text corpora that have been widely used to study semantics in the brain to help us explore semantic feature representations through the layers of CNN. This proposed methodology could contribute to better understanding of CNN and potentially pave the way for improved designing and debugging of CNN. To demonstrate an application of our methodology, we conducted experiments to determine where in the hierarchy of the CNN misclassifications emerge. The contributions addressed by this chapter are below (*Contribution B*):

- A novel methodology to study semantic representation through hierarchical layers of CNNs.
- The study of hidden representations of images that are misclassified.

4.1 Preliminaries

In this section, we introduce various Convolutional Neural Networks and Distributional Semantic Models that are used in our study. We study the semantic representations in three popular CNNs: VGG16, ResNet50 and Inception-v3. These models were selected based on their architectural diversity, performance on the ImageNet test dataset [19] and the ease of availability as pre-trained models. The Convolutional networks selected are available as a part of *Keras* framework [14] (popular deep learning framework written in Python and is used for designing and training of deep neural networks). The three networks were trained on *ImageNet* and their weights were made available as a part of *Keras*. We used four popular DS models to study the hidden representations in CNN and are described under the subsection 4.1.2.

4.1.1 Convolutional Neural Networks

We selected three CNN architectures - VGG16, ResNet50 and Inception-v3 for our study of the convergence of semantic representations in CNNs using DS models. These models are described in detail in chapter 2.

VGG16:

VGG16 is a variant of VGGNet architecture [66]. It has 16 layers (13 Convolutional layers and 3 fully connected layers) with over 138 million parameters. It has a homogeneous architecture and performs only 3*3 convolutions and 2*2 max-pooling throughout the architecture. The model along with its pre-trained weights on ImageNet is available as a part of *Keras* library and could be used as plug and play for extracting features from various layers of the network. We studied semantic representation in all the 13 convolutional layers as well as the two fully connected layers (FC-4096) before the final classification layer. The convolutional layers separated by the max-pooling layers are grouped into blocks resulting in a total of five convolutional blocks.

Inception-v3:

We studied the Inception-v3 variant of the original GoogLeNet architecture [73]. The inception-v3 architecture has 94 convolutional blocks followed by ReLu activations.

We studied all the 94 activation layers as part of our experiments. One of the important contributions of the inception architecture is the replacement of the fully connected layers in the network with an average pooling layer. Therefore, we considered it to be essential to include this layer in our analysis. The network has roughly ten inception modules. The output of each inception block is concatenated by a filter layer before it is given as the input to the next inception layer. These filter concatenation layers , dubbed as *mixed* layers by *Keras*, were studied using our experiments.

ResNet50:

The ResNet architecture introduced as a part of the ILSVRC 2015 achieved top 5 error rate of 3.57% surpassing human error on the task [35]. In our experiments, we study the ResNet50 architecture which is a 50 layer variant made available as plug and play in *Keras*. The pre-trained ImageNet weights for this model are available for download and were used to study layer-wise representations. This particular network achieved a top-error rate of 5.25% on the ILSVRC 2015 test dataset. We studied all the 49 activation layers spread across 16 residual blocks in this network.

4.1.2 Distributional Semantic Models

We selected four popular Distributional Semantic models for studying the semantic representations in three CNN architectures. These semantic models trained with different methodologies and corpus are described in brief below. More detailed description for these semantic models could be found in chapter 2.

Word2Vec and Skip-gram:

The Word2vec model, is probably the most widely used model in NLP tasks. It was proposed by Mikolov et al. in 2013 and uses a shallow neural network to learn the embedding space [50]. The Skip-gram model trained on Google news dataset is a 300-dimensional vector.

Glove:

Glove is a regression-based semantic model published by Pennington et al. in 2014 [60]. It introduces the concept of representing the relationship between two word as their co-occurrence probabilities. This is a 300-dimensional vector model .

Cross-lingual:

This model, proposed by Faruqui and Dyer, 2014, takes into account semantic properties across various languages [21]. It was trained using both German and English words using WMT-2011 corpus and uses a shared semantic space to learn the word embeddings. The resulting word vectors have 512 dimensions.

RNN:

A Recurrent Neural Network (RNN) trained to predict the next word in the sequence (Mikolov et al., 2011) [51]. This model trained on broadcast news transcriptions has a dimension of 640 for its word embeddings.

4.2 Methodology

In this section, we discuss the methodology that we followed in the study of semantic representations in Convolutional Neural Networks. The process is summarized in Figure: 4.1.

Concept Selection

The CNNs used in our study are pre-trained on the ImageNet Dataset [19] which has 1000 labelled image classes. These image classes are organized as per the WordNet [24] hierarchy. In WordNet, the words are arranged into sets of synonyms called synsets, and these *"'synsets are interlinked with each other by the semantic and lexical relationship"'* [24]. Even though the number of concepts in semantic models is much higher (SkipGram alone has 300,000-word vectors), we only select concepts which are common to the ImageNet classes and all the semantic models. We found 682 concepts in Global context, 573 in RNN, 715 in Cross-Lingual, 828 in Glove and 838 in Skip Gram This resulted in 553 concepts which are common to all semantic models and ImageNet and are used in our experiments described below.

Extracting hidden representations from CNN layers

We randomly selected five distinct images for each of 553 concepts from the cross-validation dataset released as a part of ImageNet Large Scale Visual Recognition Challenge, 2012 (ILSVRC2012) [64]. This resulted in a total of 2765 unique images

(5 disjoint sets of w images, where $w = 553$). All the images were rescaled to the size of 224* 224 for ResNet50 and VGG16. For Inception-v3, they were resized to 299*299 dimensions. After resizing, the pixel values were mean normalized. We then generated layer-wise output for the convolutional networks using *Keras* framework.

The framework also provides convenient functions to extract the hidden layer representations generated by the CNN for each image input. For every layer in a network, this resulted in 5 disjoint CNN matrices $I \in \mathbb{R}^{w*k}$ where k is the dimension of the flattened output of the CNN layer. Each row in the matrix I represents the hidden representation of a concept w_i extracted from a layer of CNN.

Pearson Correlations

We then compute the Pearson correlation of every concept w_i in a CNN matrix I with every other concept in I resulting in a correlation matrix $C_I \in \mathbb{R}^{w*w}$. This implies that every row C_{I_i} in the correlation matrix C_I represents the similarity of the hidden representation of a concept i with every other concept $i = 1 \rightarrow w$ in the matrix I. This process is repeated for all the 5 CNN matrices I resulting in 5 CNN correlation matrices C_I.

For each DS model, we extract the word vectors for the same 553 concepts from the text file (Pre-trained word vectors are available as text file for each DS model) resulting in the matrix $D \in \mathbb{R}^{w*n}$, where w is the number of words and n is the number of dimension of the word vector. We then compute the Pearson correlation of every word in word vector matrix D with every other word in the matrix resulting in the correlation matrix C_D where $C_D \in \mathbb{R}^{w*w}$. The matrix C_D represents the similarity of a word i with every other word $i = 1 \rightarrow w$ in the matrix D. Now we have two matrices C_I and C_D representing similarities of concepts in different vector spaces.

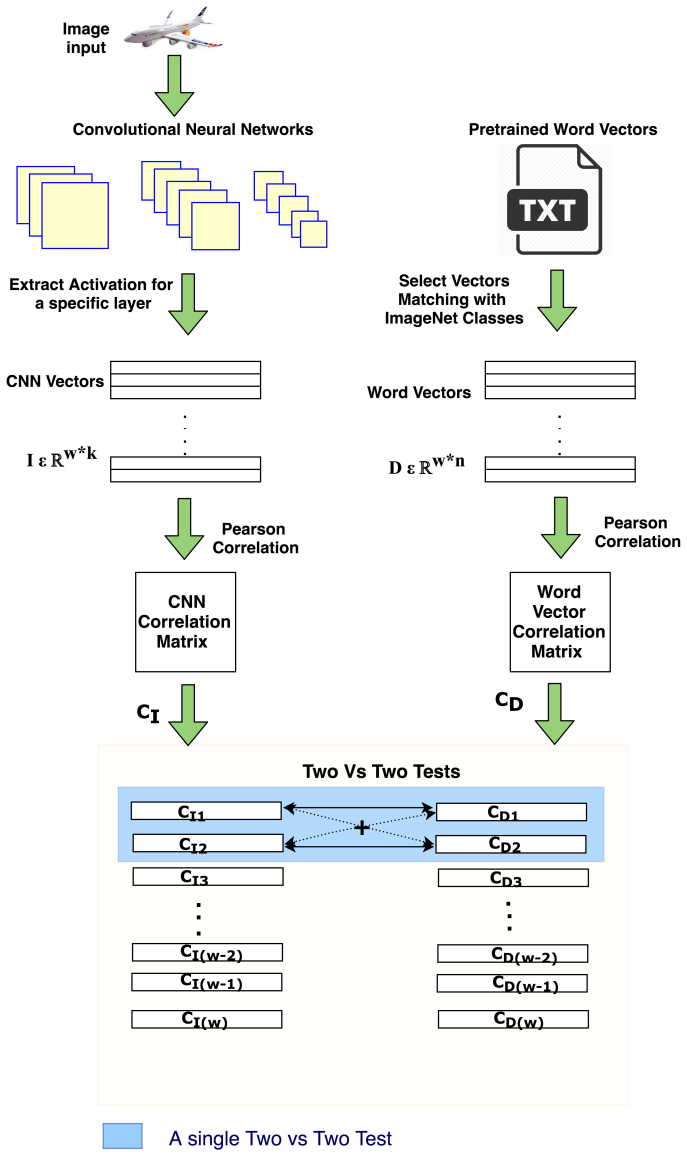


Figure 4.1: The methodology for the Study of Semantic Representations in CNN We extract output representations from various layers of a CNN for 553 concepts resulting in CNN matrix (I). Then we compute Pearson correlation for every concept in I with every other concept resulting in CNN correlation matrix (C_{I_i}). word vectors corresponding to same 553 concepts are extracted from a DS model (D) and Pearson correlations computed to get the word vector correlation matrix C_D. Then we evaluate the correlation between (C_{I_i}) and C_D using the 2 vs. 2 tests. Here w is the number of concepts (553), n is the number of dimensions of word vector, and k is the dimension of the flattened output of the CNN layer.

The similarity between C_I and C_D could be studied using the 2 vs. 2 test which is described below.

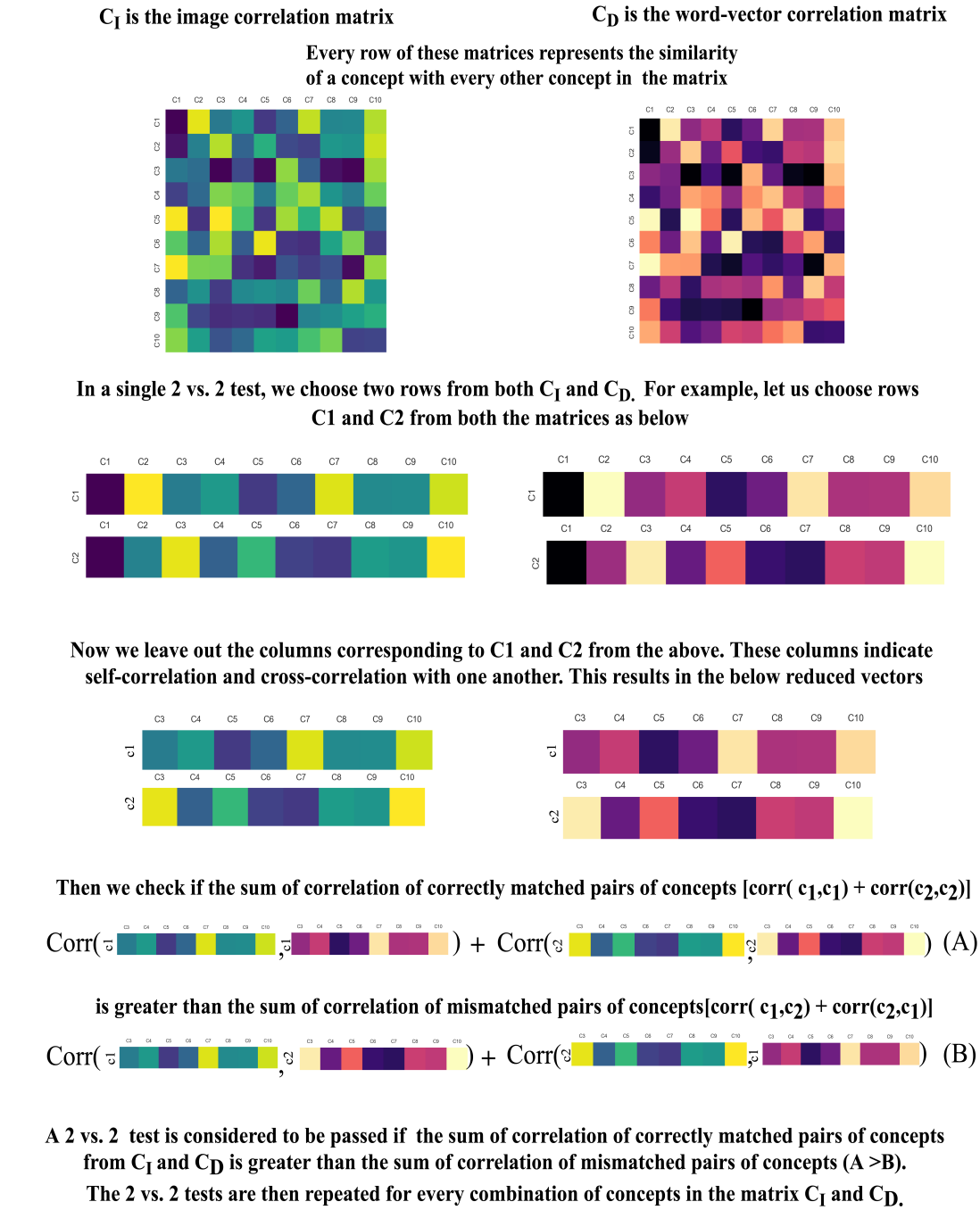


Figure 4.2: A Pictorial Representation of 2 vs. 2 test.
This figure depicts an example of the 2 vs. 2 test demonstrated for 10 concepts.

In 2 vs. 2, we select the rows corresponding to two concepts (c_1 and c_2) from our correlation matrices C_I and C_D. We then omit the columns corresponding to two concepts resulting in a reduced vector with $w-2$ columns from both the matrices where w is the total number of concepts. Lets call the reduced vectors as C_{I_1}, C_{I_2} from correlation matrix C_I and C_{D_1}, C_{D_2} from correlation matrix C_D. The correlation of the concepts c_1 and c_2 from C_I and C_D are then computed to check if the correlation of the correctly matched pairs:

$$corr(C_{I_1}, C_{D_1}) + corr(C_{I_2}, C_{D_2})$$

is greater than the correlation of the mismatched pairs:

$$corr(C_{I_1}, C_{D_2}) + corr(C_{I_2}, C_{D_1})$$

A 2 vs. 2 test is considered to be passed if the correlation of the matched pairs is greater than the correlation of the mismatched pairs. The test is repeated for all possible pairs of concepts in our dataset. This results in ${}^{w}Choose_2$ tests for a dataset with w concepts. The 2 vs. 2 accuracy is the percentage of the number of 2 vs. 2 tests passed to the total number of 2 vs. 2 tests. Since this is a binary classification task, the chance accuracy is 50%.

The 2 vs. 2 tests were repeated for the 5 CNN matrix C_I independently, and the scores were subsequently averaged to get a single score for a given layer of CNN. This was done to account for variability across images for a single concept in ImageNet. The whole process is then repeated for other layers in CNN, and a line graph was plotted which represents the variations of 2 vs. 2 test accuracy through layers of the network (plotted and discussed in the *Results and Discussions* chapter).

4.3 Study of Misclassification by CNN

During the experiments, we found that for a given set of w concepts, the CNN misclassified m concepts and predicted $w-m$ concepts correctly. This prompted us to search through the hierarchical layers of the CNN to pinpoint the layer where the misclassifications emerged. Such a study could help us to understand problem areas in CNN architecture and might provide a roadmap to debug and improve CNNs in general.

For a given misclassified concept i, there is a true class t_i and a predicted class

p_i. A misclassified concept is defined as a concept where $p_i \neq t_i$. From the set of **correctly predicted** $w - m$ concepts, we sampled 100 concepts randomly and extracted hidden layer representation using single image per concept. This resulted in the matrix $cnn_{correct} \in \mathbb{R}^{100*k}$ where k is the dimension of the flattened CNN layer. Similarly, word vectors corresponding to these 100 concepts were extracted from the Skip-gram model. Lets call this matrix $wv_{correct} \in \mathbb{R}^{100*p}$ where p is the dimensions of the Skip-gram word vector.

Next, for each misclassified concept i in m, hidden layer representations were extracted for each CNN layer, and Pearson correlation were computed with every concept in $cnn_{correct}$ resulting in the vector $i_{misclassified}$. The word vectors corresponding to true class t_i and predicted class p_i were also extracted and Pearson correlations computed with every concept in $wv_{correct}$ resulting in two vectors w_{true} and $w_{predicted}$. The vector $i_{misclassified}$ represents the correlation of concepts in CNN vector space whereas w_{true} and $w_{predicted}$ represents correlation of concepts in word vector space. We then check to see if the $i_{misclassified}$ is closer to w_{true} or $w_{predicted}$ (we compute Pearson correlation to study similarity) as below:

$$corr(i_{misclassified}, w_{true}) > corr(i_{misclassified}, w_{predicted})$$

The above test is defined as 1 vs. 2 test, and the chance accuracy is 50%. The test is then repeated for all m misclassified concepts. The whole process is repeated for other layers of a CNN, and a line graph plotted to study the variation of 1 vs. 2 scores through the layers for a given CNN. The results of these tests are discussed in the section *Results and Discussions*.

4.4 Statistical Significance Tests

The 2 vs. 2 and 1 vs. 2 tests were designed to study and compare the relationship between concepts in different vector spaces. The chance accuracy for both these tests is 50%. The p-value is calculated as the percentage of permutation accuracies which are greater than accuracies returned by our 2 vs.2 tests or 1 vs. 2 tests. It is calculated by conducting 1000 permutation tests by randomly shuffling the rows and columns of the correlation matrix C_D and re-running the 2 vs. 2 test for each permutation. In the 1 vs. 2 test, we have the w_{true} and $w_{predicted}$ vectors which represent the correlation of true and predicted concepts in word vector space for a misclassified concept i. The

1 vs. 2 tests are repeated 1000 times for each misclassified concept after randomly selecting the word vectors for w_{true} and $w_{predicted}$.

However, in our experiments, we are comparing the test-scores across multiple layers of a CNN. The probability of false discoveries increases with the number of experiments conducted. To minimize the false discoveries in our experiments, we apply BenjaminiHochbergYekutieli correction (BHY) to control the false discovery rates (FDR) associated with our tests [11]. This method ensures that we control the proportion of the false discoveries below a threshold Q which is usually set at 5%.

The steps followed in this method are listed below:

Step1: Sort the individual p-values p from permutation tests in ascending order

Step2: Assign ranks to the p-values, where the smallest p-value will have rank 1, second smallest rank 2 etc.

Step3: Calculate the Benjamini-Hochberg critical value for each p-value and check for,

$$p_{(k)} <= \frac{k}{m.c(m)}Q$$

Where k is the rank, m is the total number of experiments, Q is the false discovery rate. If the experiments are independent of one another or positively correlated, then $c(m) = 1$. Since the 2 vs. 2 tests for each layer of CNN is independent of other layers of CNN, we have set $c(m) = 1$. The scores which satisfy the condition above are considered to be true discoveries or statistically significant results.

4.5 Results and Discussions

In this section, we report the results of our experiments studying the semantic representation through the layers of Convolutional Neural Networks. We studied three popular CNNs using four popular word vectors utilizing the methodology described in this chapter. The results of our experiments are summarized in Figure: 4.3. The permutations tests were run for all possible pairs of CNN layers and DS models. The results reported here were found to be statistically significant after BHY correction. To make an effective comparison with the initial layers of the CNN, we also calculated the pixel-level correlation with the word-vectors using 2 vs. 2 tests. The raw image pixels of the images were flattened, and their correlation matrices were created. Then

the 2 vs. 2 tests were conducted against the correlation matrices of word-vectors. The pixel only scores are also shown in the Figure: 4.3.

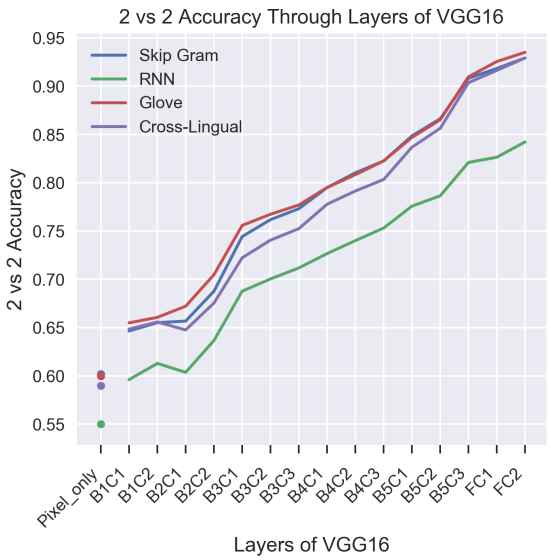
4.5.1 CNNs Learn Semantics from Images

We can see from Figure 4.3 that there is an upward trend in the growth of semantic information through the layers of all the three convolutional networks. The patterns of the graphs for Skip-gram, Glove and Cross-lingual are nearly identical for VGG16 with each of the word-vectors having a score of 0.65 for the first convolutional block and reaching a maximum 2 vs. 2 score of 0.94 at the fully connected layer just before the soft-max. The growth of semantic information through the layers of VGG16 could be better visualized from the annotated architecture diagram (Figure: A.3).

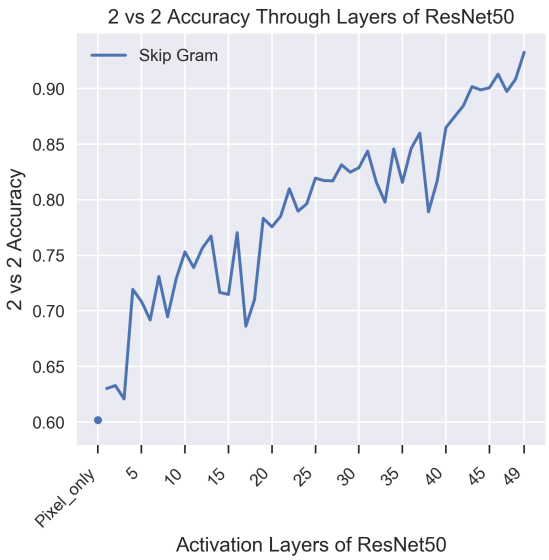
In the case of ResNet50 and Inception-v3, the study of the semantic representations was done only using Skip-gram. It should be noted that Skip-gram had the highest coverage of concepts from ImageNet. Moreover, the performance of Glove, Skip-gram and Cross-Lingual were nearly identical in our experiments with BrainBench (discussed in the previous chapter). These reasons prompted us to study ResNet50 and Inception-v3 using only Skip-gram.

The curves depicting the growth of semantic information for Inception-v3 and ResNet50 is noisy as compared to that of VGG16. Recall that the VGG16 has a homogeneous architecture and performs only 3*3 convolutions and 2*2 max-pooling throughout the architecture. The homogeneous nature of VGG architecture could have contributed to a relative smoother semantic growth curve for VGG16 as shown in Figure 4.3a.

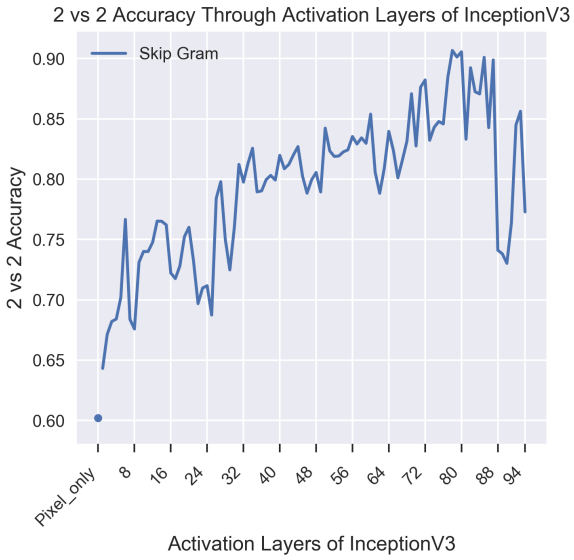
ResNet50 architecture is divided into residual blocks which causes the flow of semantic information to be non-uniform. It is also interesting to observe from the ResNet50 architecture diagram (Figure: A.4) that the 2 vs. 2 accuracy for the activation layer immediately before a residual block (or after an *add layer*) is always greater than the 2 vs. 2 accuracy for activation layers inside a residual block. According to the theory of residual learning, each residual block could be considered as an identity module calculating a small change $F_{(x)}$ for an input x (input to the residual block). The *add* layer at the end of residual blocks combines the information $F_{(x)}$ computed by a residual block with x. Combining $F_{(x)}$ with x facilitates access to comparatively higher semantic information to the activation layer after a residual block.



(a) 2 vs. 2 Accuracy through the layers of VGG16



(b) 2 vs. 2 Accuracy through the activation layers of ResNet50.



(c) 2 vs. 2 Accuracy through the activation layers of Inception-v3.

Figure 4.3: The Study of Semantic Representation Through Layers of Various CNNs. This figure summarizes the results obtained in the study of semantic representations through the layers of CNNs. To make a better judgment on the performance of the first convolutional block of each network, we also studied the 2 vs. 2 accuracy for each word-vectors directly against pixel values from the same images shown as a dot in the figures above.

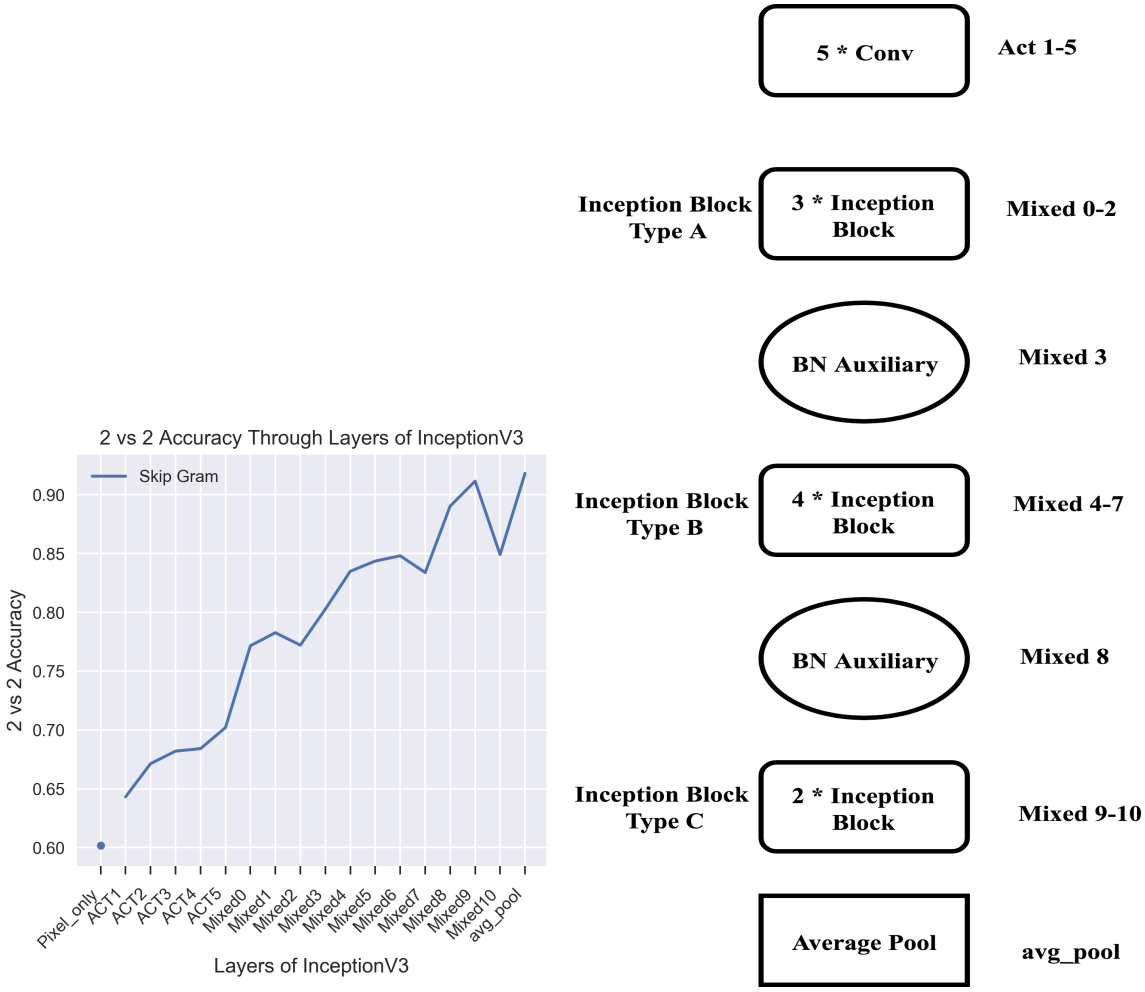
Moreover, as per the authors of ResNet architecture, the layers deep inside a residual block could face a *"degradation problem"* [35] which could explain the comparatively lower 2 vs. 2 accuracies for activation layers inside a residual block. This *"degradation problem"* is mitigated by using skip-connections between residual blocks. We believe that the ups and down in the semantic information flow for ResNet50 as shown in the Figure 4.3b is justified as per the theory of residual learning which forms the basics of ResNet architecture.

The graph for activations layers of Inception-v3 is noisier than ResNet50. An inception block consists of $1 * 1$, $3 * 3$ and $5 * 5$ convolutions happening in parallel along with a max pooling layer. Moreover, in the Inception-v3 variant, there is a further factorization of a $n * n$ convolutions into a $1 * n$ convolution followed by $n * 1$ convolution. The activations of these different convolutions operations do not hold the same information and could account for the wide variability in the 2 vs. 2 scores as shown in the Figure: 4.3c.

The Inception-v3 architecture consists of three different types of inception blocks stacked upon one another. Between, each type of inception blocks, a heavily batch normalized axillary block (BN-auxiliary) is inserted to mitigate performance degradation issues which generally arises in deep neural networks. The authors of Inception-v3 claims that BN-auxiliary layer acts as a regularizer and contributes to a 0.4% improvement in the top-1 accuracy of the network [73]. An overview diagram of Inception-v3 could be seen in the Figure: 4.4b.

In the Figure: 4.4, the 2 vs. 2 accuracy through the filter concatenation (Mixed) layers along with the average pooling layer of the Inception-V3 is shown. The concatenation layer joins the information from various parallel network operations happening within an inception block. 2 vs. 2 accuracies through these concatenate layers shows a smoother upward trend as compared to the activation layers inside inception blocks (Figure: 4.3c). Another important observation is that there is a dip in the 2 vs. 2 accuracies between layers Mixed 1-2, Mixed 6-7, and Mixed 9-10.

Moreover, BN auxiliary layers (Mixed3 & Mixed8) immediately after Mixed2 and Mixed7 show a considerable improvement in the 2 vs.2 scores. It should be noted that these BN-Auxiliaries were introduced between type A/type B and type B/type C inception blocks with the sole purpose of mitigating the problems of over-fitting using heavy batch normalization. This effect could be directly visualized in the Figure: 4.4 as well as the annotated inception-v3 architecture diagram (Figure: A.5). It should also be noted that the average pooling at the end of the last inception block



(a) 2 vs. 2 Accuracy for inception blocks in Inception-v3.

(b) Reduced architecture diagram for Inception-V3 annotated with x-ticks from the graph on the left.

Figure 4.4: Semantic information flow through the Inception blocks of Inception-v3 The graph on the left shows the growth of semantic information through the layers of Inception-v3. We can see that between layers Mixed 1-2, Mixed 6-7, and Mixed 9-10 there is a dip in the 2 vs. 2 accuracy. The BN auxiliary layers (Mixed 3 & 8) were introduced to provides a substantial boost to the information flow in the network. Authors of the Inception-v3 claimed that a 3rd BN auxiliary after the last inception block (Mixed 10) did not provide any improvement to the classification accuracy of the network and were not included as a part of the architecture [73]. However, the use of average pooling, in the end, does boost up the information flow in the network. Type A, type B and type C are different types of inception block as described by in the Figure: A.2 respectively.

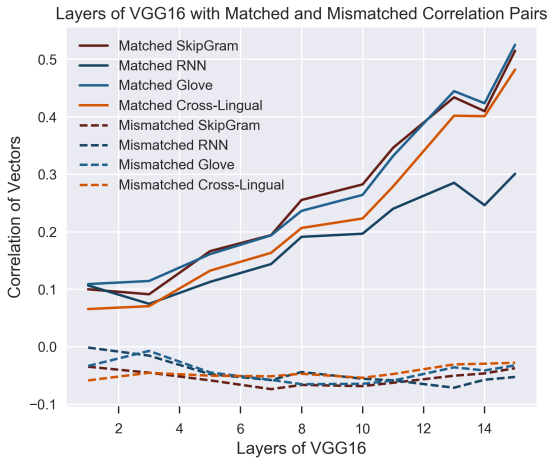


Figure 4.5: Confidence of the 2 vs. 2 tests.
The graph depicting the confidence of the 2 vs. 2 tests. We can clearly see that in the initial layers, the correlation of matched pairs are really small and therefore the test is considered to have passed with low-confidence.

provided a tremendous boost in the network performance while reducing the number of parameters in the network drastically.

4.5.2 First Convolutional Layer Itself Learns Semantics

The purpose of the experiments described in this chapter was to confirm our belief that deep convolutional neural networks could represent the semantics of an image. Our results indicate that the semantic information grows in an upward trend from the first convolutional layer and peaks at the layer before the classification layer. An important observation from the results in Figure: 4.3 is that even the first convolutional layer had a statistically significant correlation with the word-vectors. Also, first layer scores are comparatively higher (7% to 10%) compared to the pixel-only 2 vs. 2 scores which are directly computed from images without extracting features from a CNN. However, we found that the confidence of the 2 vs. 2 tests for the initial layers of CNN are comparatively lower as compared to later layers (Figure: 4.5).

In a 2 vs. 2 test, the correlation of the matched pair of vectors should be greater than the mismatched pair of vectors. We define 2 vs. 2 test confidence as the difference between the sum of correlation of matched pairs and sum of correlations of mismatched pairs. As we can see from the Figure: 4.5, the correlation of the matched pairs increases with the depth of the network whereas the correlation of the mis-

matched pair remains close to zero.

It has been shown that the first layer of CNN does learn some low-level features such as edges, angles, patches of color etc. in an image [81, 78]. Therefore, some semantic information could be decoded by the first layers of CNN directly using low-level features in the image. For example, the presence of deep blue color patches in an image could activate convolutional filters in the first layer of a CNN. This could, in turn, inform the CNN, there could be a concept like an ocean or sky in the image. If there is an ocean, then there could also be other concepts like boats, ships, beach etc. in the same image. Another argument is that most of the concepts found in nature (animals, fruits etc.) are curvy as compared to man-made concepts (vehicles, buildings, tools etc.) which are usually made of straight lines. The feature representation in the CNN layers increasingly correlates with the semantic representation as we move through a CNN's hierarchical layers.

4.5.3 Misclassifications in CNN

During our experiments, we found that the CNNs misclassified some images. This prompted us to search through the hierarchical layers of the CNN to check if the information required to make the correct prediction exist in any layer of the CNN. This was done using the 1 vs. 2 tests described by the section 4.3 of this chapter. We found that, for the VGG16 network, the information required to make the correct prediction does indeed exist in its intermediate layers.

In Figure: 4.6, the average 1 vs. 2 accuracy for each block of VGG16 is plotted. The 1 vs. 2 scores for convolutional layers which are separated by the max-pooling layers are averaged and grouped as blocks. The layers which passed the statistical significance tests (with FDR control) are marked as significant. We see that the semantics of the true class is present in the block3, block4 and the fully connected layers even if the CNN misclassified the image. Another surprising observation was that for the ResNet50 and Inception-v3, the 1 vs. 2 tests did not produce any statistically significant results after BHY correction. This implies that the information required to make the correct classification decision does not exist within any layers of ResNet50 or Inception-v3 but does exist in VGG16.

To explain these anomalous results, we decided to perform a qualitative analysis by looking into the mistakes made by these three networks. We found that VGG16 makes more egregious classification mistakes as compared to ResNet50 and Inception-

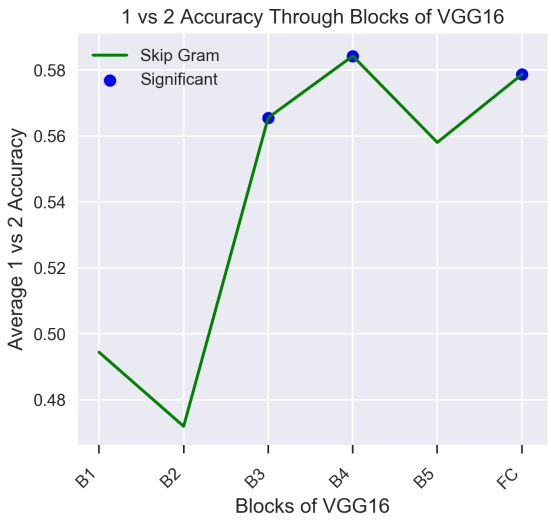


Figure 4.6: 1 vs. 2 accuracy through layers of VGG16
This figure shows the 1 vs. 2 accuracy through layers of VGG16 using the activation patterns of misclassified images.

v3. A small number of mistakes made by these CNNs were selected manually and are shown under Figure: 4.7. We compute the cosine similarity of word-vectors of true and predicted class. If the cosine similarity between true and predicted class is smaller, then we consider the mistake to be bigger and vice-versa. It is easy for the network to mistake between similar looking concepts. The similar looking concepts usually have high cosine similarity between them and we consider such mistakes to be genuine and less serious.

In the figure, misclassification of images 5-8 could be considered as more serious as compared to 1-4. Concepts which are used in the same context in text corpora often have similar semantic and visual properties. In image 1, both Catamaran and Trimaran are quite similar looking concepts and are used in same contexts in text corpora (sailing). This might account for a high cosine similarity between true and predicted class for image 1. When there is a high similarity between true and predicted concepts, it becomes difficult for the 1 vs. 2 tests to segregate the semantics of true class and predicted class.

On the other hand, a *swing* classified as a *prison* by VGG16 in image 8 could be considered as a bigger mistake as compared to misclassification of *Catamaran* as *Trimaran* in image 1. A *swing* and *prison* are concepts which are rarely used in the

same context and have a very small similarity score of 0.02. A weaker correlation between true class and predicted class could help the 1 vs. 2 test to identify the semantics of true class from the predicted class. Our analysis indicates that mistakes made by VGG16 could fall more into the category of images 5-8 (bigger mistakes).

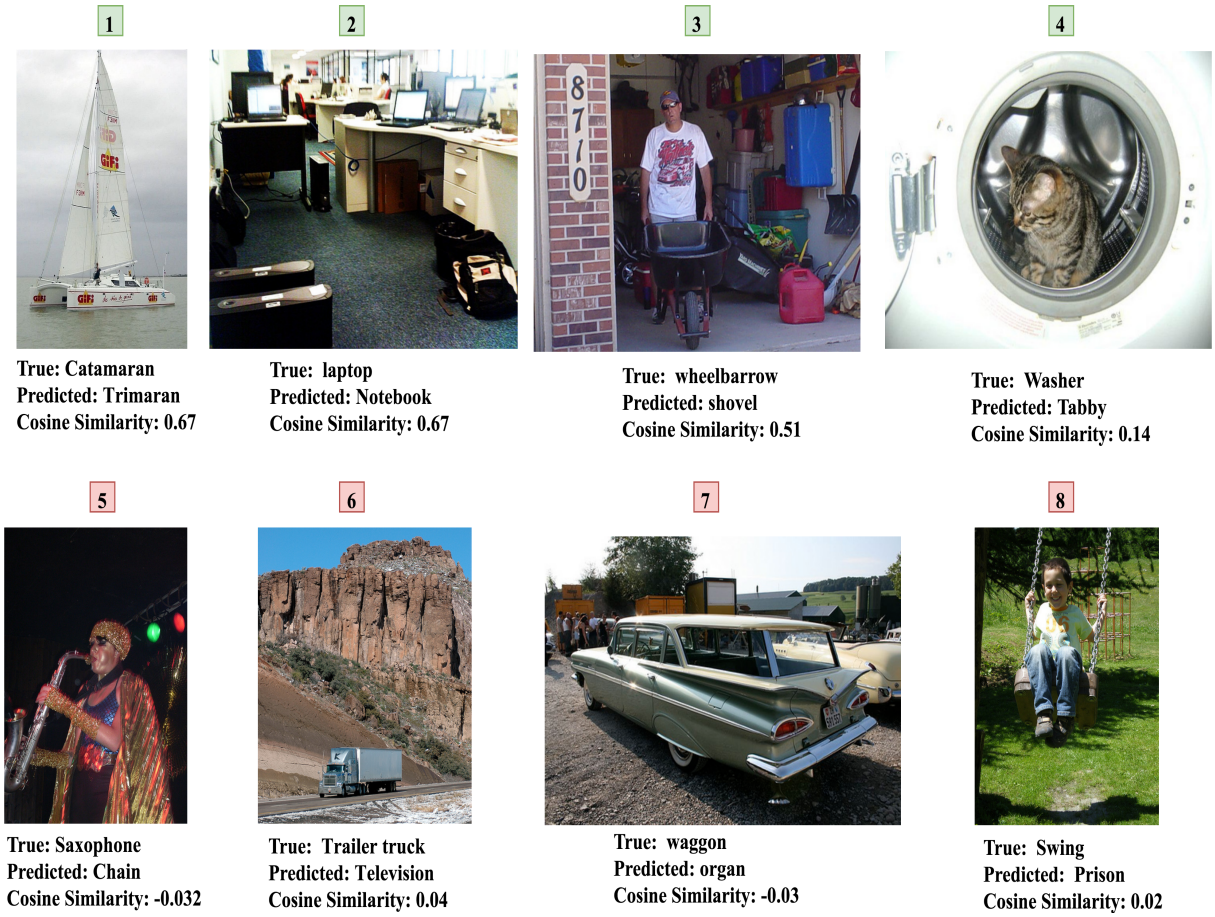


Figure 4.7: A qualitative analysis of the classifications mistakes of CNNs
The images above were manually selected to explain the results of 1 vs. 2 tests on VGG16, Inception-v3, and ResNet50. We consider a classification error to be serious if the cosine similarity between the true, and predicted class is small in word-vector space. Images (1-4) are considered to be small mistakes due to high cosine similarity between true, and predicted class, and is sampled from mistakes made by Inception-v3 and ResNet50. Images (5-8) are considered to be be more serious, egregious mistakes, and is sampled from mistakes made by VGG16.

Some examples of mistakes made by ResNet50 and Inception-v3 are shown under images 1-3. It should be noted that 1 vs. 2 tests only takes into account the top-1 accuracy which may not be a good measure for CNNs [64]. For example in image 3, CNN classified *wheelbarrow* as a *shovel*. However, there is a *shovel* present in the

image along with the *wheelbarrow* and CNN does indeed predict *wheelbarrow* correctly in its top-5 predictions. This could mean that semantics related to both *wheelbarrow* and *shovel* could be present in the CNN activations at the same time, and this could explain why 1 vs. 2 tests fails for these images. Moreover, both the concepts have a high similarity to each other making it even more difficult for the 1 vs. 2 tests to segregate between the semantics information of these two concepts.

We use this reasoning to explain why VGG16 has statistically significant points in the 1 vs. 2 tests whereas ResNet50 and Inception-v3 makes smaller and acceptable top-1 mistakes and have no significant points. Another type of scenario typically seen in the misclassification by VGG16 is shown in image 4 (*washer* vs *tabby*). Here the top1 predicted, and actual class are dissimilar but both the concepts are present in the image and also predicted by CNN in its top-5 predictions. This could mean that semantics related to both *washer* and *tabby* could be present in the CNN activations at the same time but since these two concepts are dissimilar in word vector space, the 1 vs. 2 could still identify semantics of *washer* from *tabby*. This might explain why the fully connected layers of VGG16 is still statistically significant for 1 vs. 2 tests.

4.6 Summary

This chapter focused on our methodology used to study the CNNs using Distributional semantic models *contribution B*. The preliminary section of the chapter discussed various CNN models and DS models that were included in our experiments. The 2 vs. 2 tests designed to study the semantic representation through the various layer of the networks were also elaborated. The 1 vs. 2 tests was designed to investigate misclassifications in CNN. The results and discussion section elaborated our findings and observations. Our results indicate that CNNs do indeed learn semantics. There is an upward trend in the growth of semantic information from the initial layer to the final layer of all the three CNNs that we have studied in this thesis. In case of misclassifications, the results from the 1 vs. 2 tests indicate that the semantics for the true class is present in some layers of VGG16.

Chapter 5

Conclusions and Future Work

Distributional Semantic Models (DS) or word vectors that are based on text corpora are vital for various Natural Language Processing (NLP) related tasks. Because of the immense applications of these models, there is a sheer need to perform evaluation and comparison of different DS models. BrainBench is a system designed to evaluate and benchmark word vectors using brain data published by Xu et al. in 2016 [34]. In this thesis, we publish the second iteration of BrainBench incorporating two new Italian brain datasets collected using fMRI and EEG technology. Doing so, we improved the coverage of the number of words supported by BrainBench from 60 to 190.

Another important add-on to the BrainBench suite is the ability to evaluate word-vectors against anatomical regions of interest (ROIs) in human brain. This add-on subsequently could help the computational linguistics community studying language and visual representation in the human brain using word vectors. We also conducted experiments to investigate the performance of abstract concepts in word-vectors against BrainBench. The results of this study indicate that abstract concepts show lower correlation to brain data as compared to the concrete concepts. We also provide evidence to show that DS models exhibit correlation to EEG brain signals. EEG data is much cheaper and convenient to collect as compared to fMRI and MEG. Our results with the EEG dataset, therefore, should encourage the scientific community to build more EEG based tools to evaluate and benchmark word vectors.

Although BrainBench is a robust and effective tool to evaluate word-vectors, it only has a coverage of 190 concepts even with the contributions by this thesis. Moreover, it still does not contain concepts related to other parts of speech such as adjectives, verbs, pronouns etc. The coverage of abstract nouns is only 15% of the total words. Therefore, we propose that more brain datasets need to be added to

BrainBench suite to improve the coverage of concepts and effectively making it more feasible for practical evaluation of word vectors.

Another significant contribution of this thesis is the study of semantic representation in Convolutional Neural Networks (CNN). CNNs are a computational model that has become the state of the art technology in object recognition from images. However, there is an apparent lack of understanding on why various CNN architectures perform better than the other. We used the same word vectors evaluated by BrainBench to study CNNs. In short, we asked the question: Do a CNN learn semantics?

Our results indicate that CNNs do indeed learn semantics. The semantic information in these networks grows in an upward trend from the first convolutional layer and peaks at the layer before the classification layer. We demonstrated that our methodology could explain complex architectures such as VGG16, ResNet50, and Inception-v3 by studying the semantic representations through the hidden layers of these networks. In the case of misclassifications by a CNN, we observed that the semantic information required to make the correct classification decision does exist in the intermediate layers of some networks, even if the classification layer makes wrong predictions. We hope our methodology and results could potentially pave the way for improved design and debugging of CNN.

In this thesis, we focused our study on only three CNNs: VGG16, ResNet50 and Inception-v3. There are many diverse CNNs released everywhere that could be studied with our methodology. Another exciting area of interest among the computer vision community is the study of adversarial attacks on CNNs. The techniques proposed in this work could be modified to detect adversarial attacks on CNNs. Since the hidden layers of CNNs correlate with the semantic information in word-vectors, we could potentially train CNNs using both images and word-vectors jointly to learn a shared semantic representation. Such methods might improve the generalization of these networks and could make them more robust to adversarial attacks.

Appendix A

Additional Information

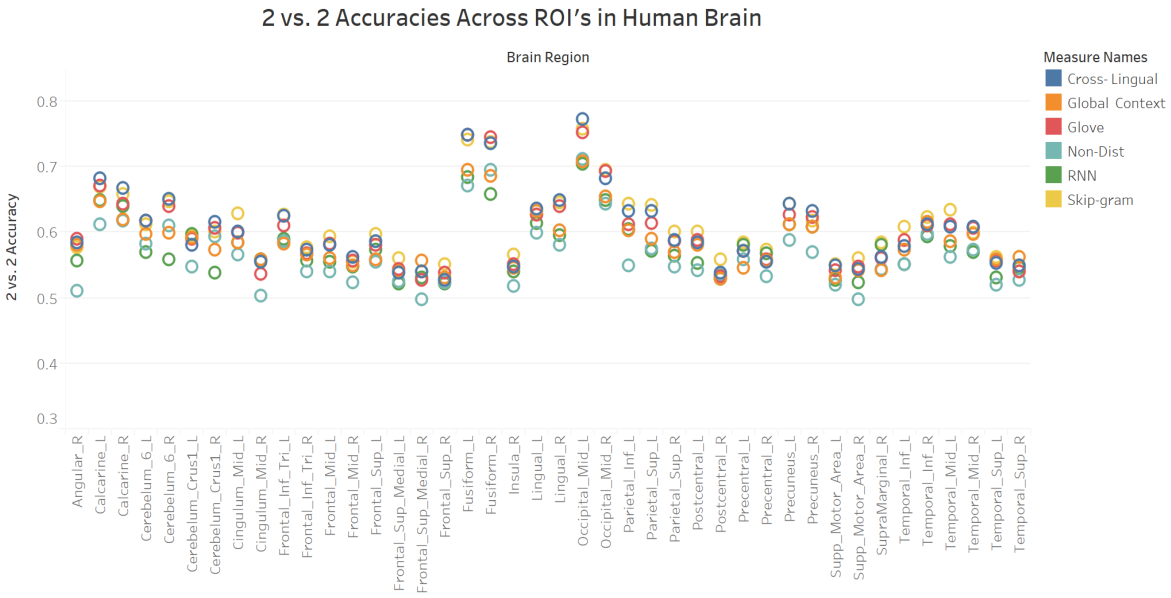
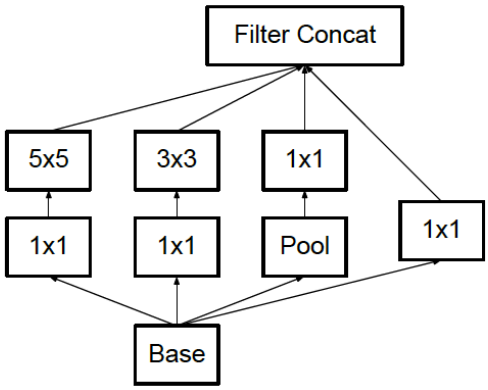
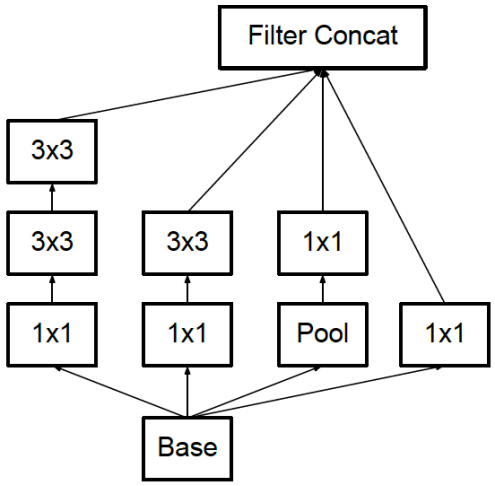


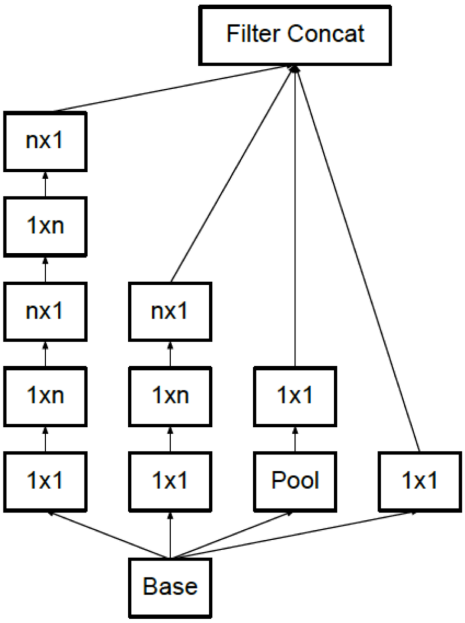
Figure A.1: BrainBench 2 vs. 2 accuracies across all 43 anatomical brain regions.



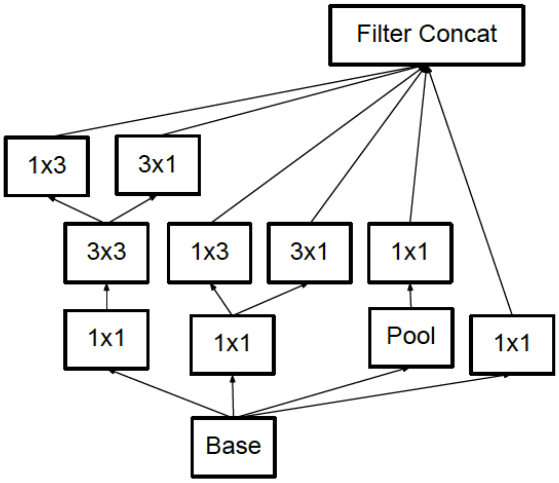
(a) The original inception module from *GoogLeNet*.



(b) $5 * 5$ convolutions broken down into two $3 * 3$ convolutions. This inception module would be referred to as *Type A*.



(c) An $n * n$ convolution is completely factorized into $1 * n$ and $n * 1$ convolutions(here n=7). This inception module would be referred to as *Type B*.



(d) Intermediate factorization of the inception module for extracting high dimensional representations.This inception module would be referred to as *Type C*.

Figure A.2: Factorizations into smaller convolutions in Inception-v3
This figure summarizes the changes in the inception module from the original GoogLeNet. Here factorization of convolutional layers are done to make the computational cost cheaper. The diagrams above are extracted from the original paper by Szegedy et al. [73].

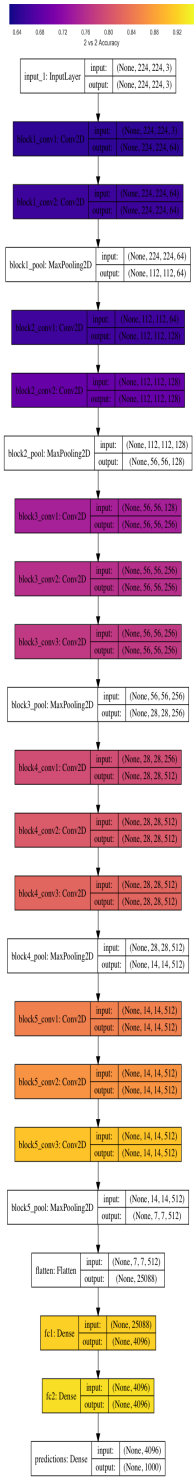


Figure A.3: 2 vs. 2 accuracy through architecture diagram of VGG16
The architecture diagram of VGG16 is annotated with 2 vs. 2 accuracy of layers against Skip-gram word-vectors [66]. This is a high resolution image and could be zoomed and viewed in a pdf.

Figure A.4: 2 vs. 2 accuracy through architecture diagram of ResNet50
The architecture diagram of ResNet50 is annotated with 2 vs. 2 accuracy of layers against Skip-gram word-vectors [35]. This is a high resolution image and could be zoomed and viewed in a pdf.

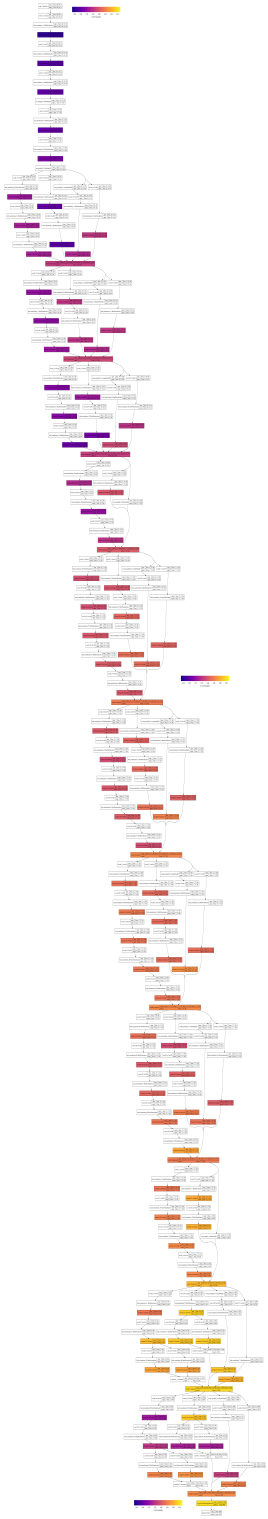


Figure A.5: 2 vs. 2 accuracy through architecture diagram of Inception-v3
The architecture diagram of Inception-v3 is annotated with 2 vs. 2 accuracy of layers against Skip-gram word-vectors [73]. This is a high resolution image and could be zoomed and viewed in a pdf.

Bibliography

[1] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27. Association for Computational Linguistics, 2009.

[2] Andrew J. Anderson, Elia Bruni, Ulisse Bordignon, Massimo Poesio, and Marco Baroni. Of words, eyes and brains: Correlating image-based distributional semantic models with neural representations of concepts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1960–1970. Association for Computational Linguistics, 2013.

[3] Andrew J. Anderson, Brian Murphy, and Massimo Poesio. Discriminating taxonomic categories and domains in mental simulations of concepts of varying concreteness. *J. Cognitive Neuroscience*, 26(3):658–681, 2014.

[4] Andrew J. Anderson, Benjamin Zinszer, and Rajeev D. S. Raizada. Representational similarity encoding for fmri: Pattern-based synthesis to predict brain activity using stimulus-model-similarities. *NeuroImage*, 128:44–53, 2016.

[5] Andrew James Anderson, Elia Bruni, Alessandro Lopopolo, Massimo Poesio, and Marco Baroni. Reading visually embodied meaning from the brain: Visually grounded computational models decode visual-object mental imagery induced by written text. *NeuroImage*, 120:309 – 322, 2015.

[6] Ben Athiwaratkun and Keegan Kang. Feature representation in convolutional neural networks. *CoRR*, abs/1507.02313, 2015.

[7] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference.*, pages 86–90, 1998.

[8] Laura Barca, Cristina Burani, and Lisa Saskia Arduino. Word naming times and psycholinguistic norms for italian nouns. *Behavior research methods, instruments, and computers: a journal of the Psychonomic Society, Inc*, 34 3:424–34, 2002.

[9] Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226, 2009.

[10] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[11] Yoav Benjamini and Yosef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

[12] Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 136–145. Association for Computational Linguistics, 2012.

[13] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[14] François Chollet et al. Keras. https://github.com/keras-team/keras, 2015.

[15] Radoslaw Martin Cichy, Aditya Khosla, Dimitrios Pantazis, Antonio Torralba, and Aude Oliva. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific reports*, 6:27755, 2016.

[16] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[17] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[18] Arnaud Delorme and Scott Makeig. Eeglab: an open source toolbox for analysis of single-trial eeg dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1):9 – 21, 2004.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255, 2009.

[20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[21] Manaal Faruqui and Chris Dyer. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*, pages 462–471, 2014.

[22] Manaal Faruqui and Chris Dyer. Non-distributional word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 464–469, 2015.

[23] Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. Problems with evaluation of word embeddings using word similarity tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35. Association for Computational Linguistics, 2016.

[24] Christiane Fellbaum. A semantic network of english: The mother of all wordnets. *Computers and the Humanities*, 32(2):209–220, Mar 1998.

[25] HuaMin Feng and Tat-Seng Chua. A bootstrapping approach to annotating large image collection. In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pages 55–62. ACM, 2003.

[26] Adriano Ferraresi. Building a very large corpus of english obtained by web crawling: ukwac. *Unpublished masters thesis, University of Bologna, Italy*, 2007.

[27] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414. ACM, 2001.

[28] Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *CoRR*, abs/1704.03296, 2017.

[29] Kunihiko Fukushima and Sei Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6):455–469, 1982.

[30] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[31] Abel Gonzalez-Garcia, Davide Modolo, and Vittorio Ferrari. Do semantic parts emerge in convolutional neural networks? *CoRR*, abs/1607.03738, 2016.

[32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016.

[33] David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4:1, 2003.

[34] Brian Murphy Haoyan Xu and Alona Fyshe. Brainbench: A brain-image test suite for distributional semantic models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Austin, USA, November 2016. Association for Computational Linguistics.

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[36] Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

[37] Tomoyasu Horikawa and Yukiyasu Kamitani. Generic decoding of seen and imagined objects using hierarchical visual features. *Nature communications*, 8:15037, 2017.

[38] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 873–882, 2012.

[39] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160(1):106–154, 1962.

[40] Wanjun Jin, Rui Shi, and Tat-Seng Chua. A semi-naive bayesian method incorporating clustering with pair-wise constraints for auto image annotation. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 336–339. ACM, 2004.

[41] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[42] N Kriegeskorte, M Mur, and P Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2(5):4, 2008-11-24 00:00:00.0.

[43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 1106–1114, 2012.

[44] Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pages 396–404, 1989.

[45] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[46] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[47] Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *Int. J. Comput. Vision*, 120(3):233–255, December 2016.

[48] Scott Makeig, Anthony J. Bell, Tzyy-Ping Jung, and Terrence J. Sejnowski. Independent component analysis of electroencephalographic data. In *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27-30, 1995*, pages 145–151, 1995.

[49] Stéphane Mallat. Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 374(2065), 2016.

[50] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[51] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048, 2010.

[52] Tom M. Mitchell, Svetlana V. Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L. Malave, Robert A. Mason, and Marcel Adam Just. Predicting human brain activity associated with the meanings of nouns. *Science*, 320(5880):1191–1195, 2008.

[53] Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[54] Brian Murphy, Marco Baroni, and Massimo Poesio. Eeg responds to conceptual stimuli and corpus semantics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, EMNLP '09, pages 619–627, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[55] Brian Murphy, Partha Talukdar, and Tom Mitchell. Selecting corpus-semantic models for neurolinguistic decoding. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 114–123, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[56] Anna C Nobre, Truett Allison, and Gregory McCarthy. Word recognition in the human inferior temporal lobe. *Nature*, 372(6503):260, 1994.

[57] Donna E Norton, Saundra E Norton, and Amy A McClure. *Through the eyes of a child: An introduction to children's literature*. Pearson Merrill Prentice Hall Upper Saddle River, NJ, 2007.

[58] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018.

[59] Charles E Osgood. Cross-cultural comparability in attitude measurement via multilingual semantic differentials. *Current studies in social psychology*, pages 95–107, 1965.

[60] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014.

[61] Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. Multiwordnet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*, January 2002.

[62] Daniel Preotiuc-Pietro, PK Srijith, Mark Hepple, and Trevor Cohn. Studying the temporal dynamics of word co-occurrences: An application to event detection. In *LREC*, 2016.

[63] Arjun Punjabi and Aggelos K. Katsaggelos. Visualization of feature evolution during convolutional neural network training. In *25th European Signal Processing Conference, EUSIPCO 2017, Kos, Greece, August 28 - September 2, 2017*, pages 311–315, 2017.

[64] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, December 2015.

[65] Chi-Ren Shyu et al. Relevance feedback decision trees in content-based image retrieval. In *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on*, pages 68–72. IEEE, 2000.

[66] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[67] John Simpson, Edmund SC Weiner, et al. Oxford english dictionary online. *Oxford: Clarendon Press. Retrieved March*, 6:2008, 1989.

[68] Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman, and William T. Freeman. Discovering objects and their localization in images. In *ICCV*, pages 370–377. IEEE Computer Society, 2005.

[69] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *CoRR*, abs/1507.06228, 2015.

[70] Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society. Series B (Methodological)*, pages 111–147, 1974.

[71] Gustavo Sudre, Dean Pomerleau, Mark Palatucci, Leila Wehbe, Alona Fyshe, Riitta Salmelin, and Tom Mitchell. Tracking neural coding of perceptual and semantic features of concrete nouns. *NeuroImage*, 62(1):451 – 463, 2012.

[72] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[73] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[74] Samu Taulu and Matti Kajola. Presentation of electromagnetic multichannel data: The signal space separation method. *Journal of Applied Physics*, 97(12):124905, 2005.

[75] M. A. Uusitalo and R. J. Ilmoniemi. Signal-space projection method for separating meg or eeg into components. *Medical and Biological Engineering and Computing*, 35:135–140, 1997.

[76] Haiguang Wen, Junxing Shi, Yizhen Zhang, Kun-Han Lu, Jiayue Cao, and Zhongming Liu. Neural encoding and decoding with deep learning for dynamic natural vision. *Cerebral Cortex*, pages 1–25, 2017.

[77] Wikipedia contributors. Noun — Wikipedia, the free encyclopedia, 2018. [Online; accessed 17-April-2018].

[78] Jason Yosinski, Jeff Clune, Anh Mai Nguyen, Thomas J. Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *CoRR*, abs/1506.06579, 2015.

[79] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535, June 2010.

[80] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[81] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars,

editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing.