Evaluation of Intra-set Clustering Techniques for Redundant Social Media Content

by

Jason Jubinville

B.Eng., University of Victoria, 2013

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Evaluation of Intra-set Clustering Techniques for Redundant Social Media Content

by

Jason Jubinville

B.Eng., University of Victoria, 2013

Supervisory Committee

Dr. Thomas E. Darcie, Co-Supervisor

(Department of Electrical and Computer Engineering)

Dr. Stephen W. Neville, Co-Supervisor

(Department of Electrical and Computer Engineering)

ABSTRACT

This thesis evaluates various techniques for intra-set clustering of social media data from an industry perspective. The research goal was to establish methods for reducing the amount of redundant information an end user must review from a standard social media search. The research evaluated both clustering algorithms and string similarity measures for their effectiveness in clustering a selection of real-world topic and location-based social media searches. In addition, the algorithms and similarity measures were tested in scenarios based on industry constraints such as rate limits. The results were evaluated using several practical measures to determine which techniques were effective.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

my family and friends, for all the continued support and inspiration.

the Echosec team, for the unprecedented opportunity to push boundaries.

Dr. Stephen Neville, for the advice and guidance along the way.

DEDICATION

I’d like to dedicate this to the friends, family, and colleagues that were by my side throughout this journey. You know who you are.

Chapter 1

1 Introduction

This chapter reviews the background technology and need for effective social media filtering tools. The problem this thesis explores will be introduced along with the thesis goals. The background of the problem will be reviewed with a brief discussion of the current industry toolset.

1.1 Problem Statement

This thesis evaluates different filtering techniques for the intelligent clustering and filtering of content rich social media for industry applications. Today, more than 100 million social media posts are generated daily [1]. For end users to effectively find and understand what is important, requires methods for reducing content volumes. An intelligent social media clustering and filtering system is extremely useful in reducing daily work effort.

This thesis extends works in the social media clustering, analysis and filtering space. As a rapidly growing phenomena, techniques and methods to understand high-volume social media is a strong area of industry and academic interest. Social media is generated on scales infeasible for manual inspection. Social media is used not only on a personal basis to communicate, but on an industry level to understand those conversations and the people behind them. Unfortunately, many tools that are currently available to industry do not adequately provide the level of filtering and classification that is required to provide truly effective platforms. This thesis was completed with the support of an industry partner, Echosec Systems Ltd [2], that understands these

industry limitations and is well positioned to provide access to both data and qualitative and quantitative understanding of research implications.

1.2 Social Media Background

Social media is a growing phenomenon and represents critical infrastructure for communication in the modern age [3] [4]. As of 2015, more than 75% all internet users in the United States were on at least one social media network, which is a ten-fold increase over the last decade [3]. There are a number of common uses for social media network technology including text-based communication, photo sharing, real-time video and more. Each social media user has his or her own application for the technologies ranging from brand development, day-to-day communication, event organization, breaking news consumption, advertising and seeking job opportunities, promoting products and more. Social media usage for news consumption and general information sharing is so ubiquitous it is understood to be one of the greatest influences in democratic systems, commonly accepted to affect the outcome of national elections [5] [6]. The applications for clustering social media content and the subsequent filtering of redundant or irrelevant information is universal across all use cases.

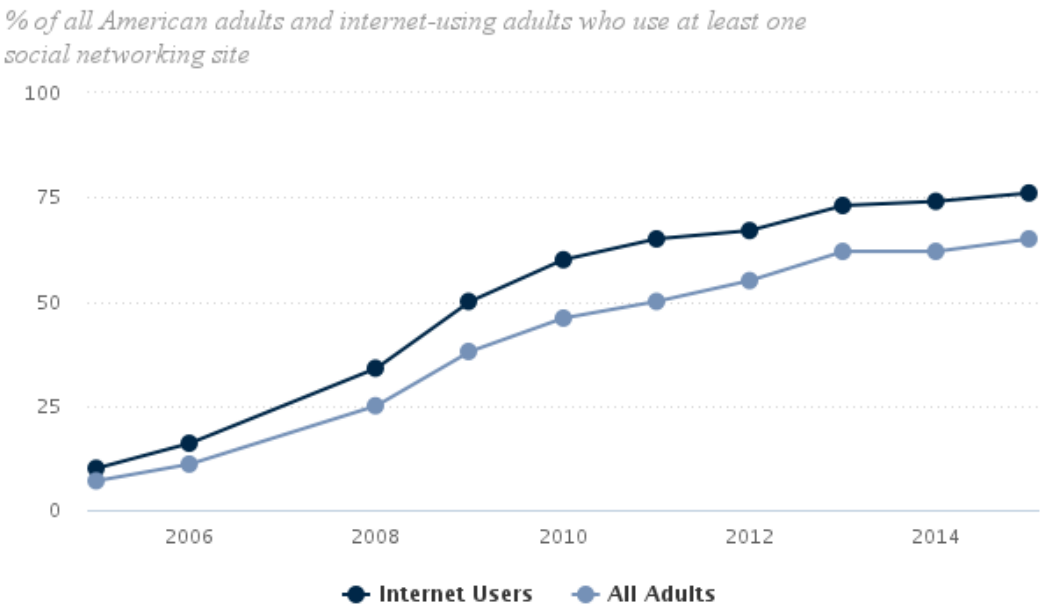

Figure 1.1: Adult Adoption of Social Media in the US [3]

1.2.1 Use of Social Media in Industry

In industry, understanding of how people engage, share, and consume information online is of significant interest to a multitude of third-party organizations. Marketing teams, security organizations, journalists, advertisers and other information professionals all have a vested interest in understanding the social conversation. Industry tool sets, typically known as social media monitoring platforms, or social media analytics platforms [7] provide insight into topics, trends and other social media phenomena that is relevant to their customers.

Despite the maturity of the social media monitoring and analytics space, many of the tagging and analysis platforms use naïve approaches to their analytics [7], such as summing hashtags, compiling word frequencies, or similar and deriving understanding from those metrics. These technologies typically focus on systems that allow marketers

to reach the largest number of consumers the quickest [8]. While this strategy may be effective for generating the highest revenue for the least input effort, it does leave a significant number of research use cases, specifically those looking for information, underserved. These tools commonly rely on their respective end users to define priority keywords, topics, and trends. This is an effective strategy in a tightly scoped scenario with a specific topic. As subject areas become global and multifaceted, more variables must be accounted for, including translation, colloquialism, synonyms, semantically similar topics and euphemisms. This list of exceptions quickly becomes unmanageable. When a use case calls for the understanding of unique one-of scenarios, it is helpful to reduce the total amount content an analyst must review.

Most industry social media applications support two core functions, a query function and an analysis function. For each industry tool, both the query function and the analysis function can vary from naïve to complex. By focusing on different types of queries or analysis functions each industry tool can differentiate itself from the others in the market. A query is the primary method by which an end user informs the platform what topics are of interest. Depending on the tool the end user is using, this query could be focused on a historical search, a real-time streaming search, or another causal time period and across any number of topics. An analysis function operates on the data returned by each social media provider and processes it for end user consumption. Different presentation mechanisms are also included as part of the analysis component. The analysis function could be as naïve as normalization and export, and as complex as a proprietary technology or algorithm. An example tool is Dataminr [9]. Dataminr is a social media alerting platform that allows its end users to receive alerts about critical events around the world, such as earthquakes or terrorist attacks. Dataminr takes a simple query of topics each user is interested in and outputs an email or text message alert when that event occurs. Dataminr’s difference, and thereby their value, is the speed and accuracy at which they can send an alert before any standard media outlet picks up on the story [10]. Regardless of a tool’s sophistication, end users will always have to manage high levels of social media noise.

1.2.2 Noise in Social Media

Noise, defined to be confounding social media information not relevant to the current analysis(es), can take many forms and is inherently subjective. We take the definition of social media noise to be: social media content that distracts from or does not contribute or distracts to the ultimate purpose for a search carried out by a social media user, marketer, investigator, or other person interested in social media. For example, a user looking to understand social sentiment around political candidates may consider bot traffic to be noise. However, at the same time a different end user that is looking to understand how bots are influencing elections could be interested in both the bot traffic and more standard social media data, but consider extraneous data collected in the same search to be noise. For Dataminr end users can use the tool to understand and respond to high urgency events as soon as possible [10]. End users would, therefore, care about all high urgency content that is relevant to them. In this scenario, noise would be classified as a false positive result (Dataminr detects a high urgency event that is not urgent), or a high urgency event that was misclassified as something you would need to know about (Dataminr alerts you of an earthquake halfway around the world). Noise can also be content that does not give an end user new information. For example, additional social media posts that convey the exact same message do not directly contribute additional information to the topic in question. Additional sources can, however, contribute to the veracity of the original post. However, the loudest or most popular topic may not reflect current affairs. For Dataminr's end users, a second alert about an earthquake may be noisy, but it supports the narrative that an earthquake is occurring. Ultimately, the frequency of occurrences provides additional information, where such information augments the contents of the messages themselves.

Developing the ability to accurately and consistently filter out such noise is an important capability to support effective social media searches. Noise reduction in a social media application allows end users to save time in processing information and make better decisions.

1.2.3 Implications of Social Media Noise In Industry

Today, many industry tools, for example Dataminr, Hootsuite, Sysomos, and Echosec [9] [11] [12] [2], are focused on directly finding and analyzing social media content collected via end-user searches for different use cases. However, these technologies, and the industry at large, lack effective methodologies for reducing noise. To reduce noise effectively a platform must implement a technology that can identify, tag, groupor remove redundant or irrelevant social media posts, where redundant content denotes social media content that provides little or no new information to the subject matter or topic. Irrelevant content, by comparison, is understood to be social media content that does include unique or novel in information but does not provide valuable information for the end user.

Even when an end user searches for a specific topic, redundant or irrelevant content will be present. Commonly, this noise appears either as social media content that is effectively the same post, or content that contains the hashtag the user was looking for but represents a different topic. This occurrence is more frequent for hashtags that are broad or can represent more than one topic, for example #blues can represent both the music and the Southend United Football Club, as seen in Figure 1.2 and Figure 1.3, respectively.

Figure 1.2: #Blues Music Example

Figure 1.3: #Blues Soccer Example

1.2.4 Social Media Analysis Techniques

Social media represents an unprecedented method for researchers to understand human interaction and intent through text, image, and video sharing. There has been a significant amount of research completed in both academia and industry that attempts to understand social media content, social media users and their interrelations. These works commonly address many topics, across similar datasets that are made available to the academic community. Amongst this research, three highly prevalent topics are sentiment analysis, recommendation engines, and clustering.

1.2.4.1 Sentiment Analysis

Sentiment analysis is the process of attempting to learn and autonomously understand what sentiment an individual is trying to convey when they use a particular word or phrase [13]. Automated or computer implemented sentiment analysis, also commonly called opinion mining, has been heavily researched on the advent of social media as content generation has quickly outstripped the ability to manually read and analyze every post for meaning.

A naïve approach for sentiment analysis uses a defined lexicon of word-sentiment score pairs to establish a baseline [13]. Unknown word-sentiment score pairs can be then generated by setting the sentiment of an entity to the average the score of the known pairs in the piece of content. Typically, stop words are hard coded to a neutral value. Sentiment analysis algorithms suffer from the multi-tone and ambiguous nature of human language [14]. As a result new processes are included to improve successful classification. More sophisticated approaches employed in industry now include various natural language processing techniques, machine learning algorithms, and statistical methods [13].

1.2.4.2 Recommendation Engines

Another major topic of interest in the industrial and academic communities is serving interesting, engaging and relevant content to targeted social media users. These systems are commonly known as a recommenders or recommendation engines[15]. Recommenders are algorithms that allow social media networks to identify content based on any user's interests. These recommenders can then be used to promote relevant content in a news feed or serve a targeted advertisement. These systems often implement a solution based on the hashtag or keywords present in the social media content [16]. There are various techniques that allow recommenders to tune the system for better recommendations as the naïve approach can be improved on. These systems commonly include natural language processing, entity recognition and other feature analysis including influence of the original poster, virality, and time relevance. Ultimately, however, they are significantly based on predefined interests or user input that assists a learning model to promote appropriate content.

1.2.4.3 Clustering

Clustering social media content is a topic area that has been closely researched by both industry and academia. In social media clustering, the purpose of an algorithm is to separate distinct topics, gather lexically similar posts, cluster semantically similar posts, and other categorization and classification operations. It is common in academia for clustering algorithms to distinguish distinct topics from a unified pool of social media content, for example searches of #NBA and #Trump that have been combined into a larger set. In academia, these search datasets are sourced through publicly available datasets, other researchers, or through cost effective Twitter APIs. These data sets are then merged to create a more significant original set for training and testing. This practice, however, does not align with industry requirements. Social media monitoring platforms operate on a single search term, or multiple search terms, where the end results are not combined with other end user searches on the platform. In industry, searches are commonly far more significant in size and closer in topic than two

disparate sets available to researchers. For example, publicly accessible research set containing #JeSuisCharlie and #Trump, as used in the current state of the art [17] would logically be easier to separate than three Superbowl 2018 related searches of #Eagles, #Patriots, and #SuperBowl .

In academia, it is common to build a classification system that detects the differences between two or more data sets, or inter-set clustering. By way of example, an inter-set clustering problem would be classifying posts belonging to #MeToo and #Food. Inter-set clustering problems in the social media space are a manufactured problem, as many industry tools would merge the results of multiple queries and then perform clustering to separate out the query results.

Intra-set clustering, however, is the practice of identifying the sub-clusters that may exist within the results returned by a single search query. This is a highly useful process that allows end-users to consume, monitor, or make decisions in a data environment with less noise. Intra-set clusters have several important characteristics that are of interest to industry. Firstly, in many cases, an intra-set cluster represents redundant or repeated data that is not important to the end user. Reposts or similar posts to an original tweet do not contain new information for a social media researcher. Further, the size and density of an intra-set cluster may indicate the popularity a topic area. Likely due to data constraints and accessibility, intra-set clustering has not been heavily researched, despite being an important industry topic area.

Ultimately, there has been a significant exploration of methods for evaluating algorithms for understanding social media content especially sentiment analysis, recommenders and clustering. There appears, however, to be a lack of work in intra-set clustering for the purposes of smart social media data filtering despite its value in industry.

1.3 Twitter

Started in 2006, Twitter is a social media data company. Twitter allows its users to broadcast short posts, called Tweets, online for others to consume and interact with. Users can follow their favourite influential people, friends, and topics. Twitter posts can range in content from breaking news to informal update on daily affairs. Until recently, Tweets were limited to at total of 140 characters. As of November 2017, this was increased to a 280 character limit [18] , but this limit does not include any URLs added to the posts.

1.3.1 Twitter Data Products

Twitter makes its data available to third party developers for a wide variety of purposes [19]. This allows Twitter to develop a secondary market of applications and a development community built around the Twitter network. To do this, Twitter has several API endpoints that can be programmatically accessed by their third-party developer community. These API endpoints are bucketed into tiers of access and vary in their sophistication, data volume, data type and price. At the time of writing, these tiers are Standard, Premium and Enterprise [20], where the Premium data tier is a flexible pricing between the free Standard API and costly Enterprise API.

Twitter's Standard API is a free, heavily rate-limited source that can be appropriately used for simple applications and learning how to develop in the Twitter ecosystem. The Standard API is commonly used in academia to gather social media datasets due to its accessibility. The Standard API, however, does not guarantee data fidelity for most industry applications [20]. Twitter will also serve pseudo-cached content to its Standard API Endpoint that may affect clustering results as it may have already been influenced by user interest or engagement.

Twitter's Enterprise API data products are colloquially referred to as the 'Twitter Firehose' and are targeted at sophisticated industry organizations. Twitters Enterprise products guarantee data fidelity and unlimited throughput. Twitter Enterprise products

can offer unlimited access to both historical and real-time Tweets on any topic, keyword or search [21]. For commercial reasons, Twitter has several products that fit into the Enterprise API bucket, but importantly they cover two basic search types - historical search, and real-time streaming.

Twitter's Historical search can provide publicly available content all the way back to the first Tweet in 2006. However, due to its design Twitter Historical search is only capable of returning content in 500 Tweet buckets. An Enterprise API consumer can make multiple queries on a single test, but each package that gets returned will only contain 500 Tweets. In addition, requests are gated by a rate limit to effectively 2 requests per second. Twitter's API documentation assumes that each response takes up to 2 seconds to respond, which would depend on various factors.

Twitter's Real-Time streaming products also come in various tiers, ranging from the deca-hose at 10% of potential throughput to full fidelity Firehose access called the PowerTrack API. Different from the Historical Search APIs, PowerTrack serves content to customers as soon as it is available as a single Tweet.

Both the Historical and Real-Time streaming products are based off a query system that retrieves results matching the end users request. These requests are available across many of Twitter's features, but commonly are hashtags/keywords, usernames, and locations. Consequently, requests are not overly broad in nature and require separation.

Twitter's Terms of Service restricts the ability to analyze the performance of the API to determine specific numbers for this performance. Twitter also reserves the right to restrict any company from accessing their content based on how an organization proposes that to use the social media content. Many academic papers and data sets, Arin et al [17] and Thaiprayoon et al [22] for example, are based on the Standard API access. This may be a result of the financial barrier to entry for academic institutions to engage with Twitters Enterprise products.

1.3.2 Consequences of Twitter Product APIs

As a result, industry products must be built around Twitter's API constraints. It is also possible to leverage these constraints to build intelligent solutions. For example, a solution that analyzes historical results does not need to look at one million rows at a time, but manage a throughput of 500 per second, with a best-case scenario 2-second delay [23]. Similarly, a real time streaming system need only be able to classify a single post as compared to the previous set. Further, as query selection is facilitated by end users, a separation or classification operation is not required to categorize topics.

1.3.3 Twitter and Bots

Due to API endpoint accessibility, Twitter has had its struggles with various accounts automatically posting content to the platform [24]. Commonly called 'bots,' these programs have always existed on the Twitter platform despite the company's efforts to remove them [25]. Bots degrade Twitter end user experiences by posting significant amounts of irrelevant content. In some situations, bots can be used to sway perception on important topics including elections. In Q2 2018, Twitter increased its efforts to remove bots in response to the findings that Russian coded bots may have influenced the US 2016 election [24]. To compound this issue, there are social marketing software packages that allow users to semi-automate content distribution, such as Hootsuite or BufferApp [11] [26]. These scheduling applications allow social media marketers to reduce the overhead of posting content frequently. While these tools themselves are not bots, there are also bots that emulate industry posting habits and semi-automated tools to evade detection. It is can be challenging to differentiate an intelligently coded bot and a user posting and liking content through a semi-automated posting platform [25]. It is technically against Twitter's Terms of Service develop a tool for detecting bots, as the resource requirements could affect end user experience [27]. As a direct results of those terms, this thesis will assume content is either generated by a Twitter end user or by an end user leveraging a semi-automated scheduler.

1.4 Industry Partner Echosec Systems

Echosec Systems Ltd is a social media aggregation and analysis platform [2]. Echosec provides their industry clients real-time social media content across multiple platforms for the purposes of understanding real-world breaking news scenarios. Echosec's technology primarily focuses on a unique geo-tagged or location-based social media, but also provides standard keyword and hashtag search queries and analysis. Echosec's clients range from small news networks to Fortune 500 companies. Many of their customers rely on Echosec to help them understand the social media landscape and to react to changes in that landscape by making better informed decisions.

Over the course of their daily use of the Echosec platform, customers regularly encounter redundant and irrelevant information. For each customer, what comprises a noisy, redundant or irrelevant post is unique to their usage and use case. As a result, Echosec has a strong interest in developing a system for tagging and filtering content that can be driven by an end user and reduces the amount of information a user must process. This thesis and research focus on addressing this need.

1.5 Chapter Summary

The purpose of this research is to review and evaluate techniques for the smart filtering of social media content for industry applications. Many current clustering tools in academia are built on the problematic approach of clustering artificial datasets by merging multiple datasets than seeking to develop technique to separate these constructed data sets. This practice does not accurately represent industry's need for effective intra-set social media clustering, where the overall data contains far more similarity than exist in conglomerates of disparate data sets. More important in industry is the reduction of irrelevant or redundant data in search results by intra-set clustering similar content. Ultimately, the thesis' goal is to evaluate techniques for clustering similar social media posts to filter redundant or irrelevant content in industry-relevant social media applications.

1.6 Thesis Outline

This section outlines the subsequent contents of the thesis.

- Chapter 2 discusses existing research relevant to the smart clustering and filtering of text based social media content.
- Chapter 3 reviews the process and methods that were applied to develop and effective clustering techniques
- Chapter 4 analyzes the results of the work for various experimental setups.
- Chapter 5 summarizes the thesis, presents the thesis' conclusion and suggests recommendations for future work.

Chapter 2

2 Literature Review

Social media content clustering and filtering, as well as other document types, is a well researched field in academia and industry. The purpose of this chapter is to review and present some of the relevant works in the space of intelligent clustering and smart social media data filtering.

2.1 Content Similarity in Social Media

Twitter content is primarily comprised of text and images. While most posts contain text, only about 45% of tweets containing an image [28]. As result, work has been put into understanding the information contained in text based social media content. Many different methods have been explored for the detection of content similarity in social media including Hamming and Levenshtein Distances, Jaccard Tanimoto Similarity, Bag of Word analysis, lexical and semantic similarity, as well as suffix trees.

2.1.1 Conventional Similarity Hamming and Levenshtein Distances

Hamming and Levenshtein distances are two measures of the edit distance between two strings. The edit distance between two strings is defined as how many characters in a string would need to be changed to exactly recreate a second string [29]. Hamming and Levenshtein subject strings can be binary, alphanumeric characters, or entire strings. The measures have applications in communications, coding, and general similarity determination [30] [31]. Hamming and Levenshtein have been used as conventional baseline comparisons for more sophisticated similarity measures [32].

Hamming Distance is restricted to strings of the exact same length. While in theory, the Levenshtein distance allows for insertions at the end of strings. When applied at larger scales Levenshtein performs better for strings of the same length [32].

Both Hamming and Levenshtein Distances operate from left to right on an attribute by attribute basis. As a result, both measures are dependent on ordering and are susceptible to the omissions or small edits that are commonplace in social media content.

2.1.2 Jaccard/Tanimoto Similarity for String Similarity

The Jaccard Similarity, or Tanimoto Similarity, measure has been heavily explored in academia to help identify similar sets of numbers, strings, and other attributes. At its core, Jaccard is a comparison of the number of shared attributes as a function of the total number of attributes across two datasets [33]. Similar to Hamming and Levenshtein Distances, the comparison can be made across different attributes, whether they are binary, character, or strings. The Jaccard Similarity has the added benefit of being order agnostic.

Shameem et al found that Jaccard could be used to improve a k-Means document clustering algorithm [34]. In the research, a standard vector space model was used to translate documents into multi-dimensional Euclidean space to cluster using a standard k-Means approach [34]. Jaccard was used to identify and remove significantly dissimilar results in the k-Means algorithm to improve the initial means selected for analysis. Shameem et al, apply Jaccard as a secondary, supportive measure to the original vector space model [34].

Jaccard Similarity also suffers from high computational complexity [35]. MinHash is hashing algorithm that simulates Jaccard Similarity invented by Andrei Broder and can be used to improve the performance of Jaccard Similarity [35]. MinHash is in online algorithm and improves on the memory performance of a standard Jaccard

implementation and is appropriate for usage at large numbers of attributes. MinHash however is not necessary for smaller sets such as social media content.

2.1.3 Bag of Word Content Similarity and Classification

Initial work for understanding text in social media has been using Bag-of-Word classification methods for Twitter content [36]. Bag-of-Word (BoW) analysis involves considering each word in the sentence or document as an unordered set and attempts to draw meaning from the set. BoW implementations for Twitter content have significant limitations to their reliability due to the limited number of words present in any given social media post [37]. In BoW classifications, its is common to use these systems to group tweets into broad categories like News, Opinions, Deals, and Private Messages, but BoW classifications perform worse for increasingly specific topics. In their work to improve popularity detection based on a similarity analysis and identify meaningful tweets, B. Sriram and D. Fuhry [37] used additional features contained in the user profile to add appropriate weights to the semantic understanding of the text content such as user, retweets, replies, time and date. Bag of Word systems suffer from the limited amount of content available in any given Tweet and can only associate Tweets with broad predefined categories.

2.1.4 Suffix Trees Clustering of Twitter Content

Suffix Tree similarity algorithms have been explored for the purposes of detecting and grouping similar documents. The foremost research into Suffix Tree clustering of Twitter content was performed by I. Arin et al's Interactive Twitter Clustering Tool (I-TWEC) [17] [38]. I-TWEC is two phase clustering tool that leverages both lexical and semantic similarity to cluster a static data set comprising of sixty thousand Tweets across four primary topics including #NBA, #Trump, #jesuischarlie, and #christmas (2016). The first phase included a suffix tree clustering system that grouped Tweets by lexical similarity. The second phase took user input to group chosen clusters by semantic similarity. A suffix trees implementation of the first phase allowed the ITWEC team to build a clustering tool that worked exploited Twitter's character limit to operate in linear time.

Additional suffix tree algorithms were used on Twitter data with different focuses. Poomagal, Visalakshi, and Hamsapriya (2015), Thaiprayoon, Kongthon, Palingoon, and Haruechaiyasak (2012) and Fang, Zhang, Ye, and Li (2014) all leverage suffix tree based clustering of Twitter content to group similar Tweets [39] [22] [40]. In each case, they focus on a subset of the most popular clusters, and disregard smaller clusters based on a defined threshold.

2.2 T-Codes as a Similarity Measure

T-codes, a variable length, prefix-free code invented by Titchener [41], have been used for various applications including error detection, malware detection, cryptography, data compression and basic information classification [32]. T-Codes provide an advancement in string similarity detection by using string complexity measures and have been made to allow for strings of unfixed or unequal lengths and for performance increases across large strings. When applied, T-Codes compress a given string to subsequences which represent the basis vectors of the string. The T-Codes then can be used to determine an overall complexity measure for the string. Strings basis vectors can be compared in both an information distance and a complexity distance to determine similarity. Information distance compares the total information in strings, whereas the complexity is determined by the comparing a string to the complexity of a large random string. Yang and Speidel's work in String Parsing-based Similarity Detection determined that using Lempel and Ziv's 1976 similarity measure for string randomness [42] is a similarly effective technique for measuring the string complexity and similarity for relatively short strings. Further, Yang and Speidel found that Titchener's T-complexity measure could be effectively applied in similar situations with the added benefit of higher performance for longer strings [32] [43]. N. Rebenich et al, developed a fast T-code decomposition FLOTT, increasing performance over previous implementations of T-codes in both speed and memory utilization [44]. Rebenich also found that T-Codes have a firm basis in information theory and proved that T-complexity is not a measure [45]. T-Codes may provide an effective method for clustering Twitter content that is agnostic to language, small omissions, and other variations common in the Twitter content.

2.3 Other Document Clustering

Document clustering is the process by which an algorithm can group a set of documents into similar clusters and can be carried out in supervised or unsupervised manners. One common method for document clustering is Term Frequency - Inverse Document Frequency (ti-idf). H. Tu and J. Ding use TF-IDF and a cosine similarity measure to effectively cluster tweets into ‘hot topics,’ based on web article popularity [46]. However, the ‘hot topic’ categories were trained using a non-Twitter data set, specifically web articles, to achieve the required accuracy. H. Tu and J. Ding did not cluster all Tweets in their dataset. Any post that didn’t fit a ‘hot topic’ was discarded using a Bayesian classification filter.

Many effective algorithms exist to cluster documents such as k-Means, naïve Bayes or Gaussian mixture models, DBScan and others [47]. It is common for these clustering algorithms to categorize entities such as webpages, articles, and other relatively large documents. However, these traditional document clustering algorithms commonly operate on larger documents than a standard social media post, and as a result, breakdown for large data sets that are predominantly shorter word counts with an unknown number of clusters [17].

2.4 Recommendation Systems

Similar research to document and topic clustering in the social media space, is the recommendation systems, commonly referred to as recommenders. A recommender is a system that identifies social media content that might interest a user and promotes that content to their news feed [15]. In their work, Ramesh et al. [15] used a collaborative filtering approach to generate content recommendations. These filtering approaches use several features to appropriately recommend content including historical activity, content rank, indexing, trending, common interest and other features. Recommenders also leverage semantic and lexical similarity to suggest posts similar to an ideal suggestion. Recommenders differ from document clustering algorithms in that

they are only looking for a few interest indicators to suggest viability for it to be recommended and would be most similar to the tagging of the most popular cluster.

2.5 Spam Bot Detection

Another area of relevant research for smart filtering of social media content has been the exploration of the prevalence of spam and bot traffic on social media sites, specifically on Twitter. The detection of spam and bots has predominantly leveraged an entropy component, text-based spam component, and an account features component [48] [25]. To measure entropy, the Tweeting interval was measured. Periodic and regular timing was used as in indicator for bot activity. Other automation indicators included spam words, URL type, and Tweet composition. Further, these techniques looked at the frequency and pattern of posting in correlation with the content and its similarity to other Tweets to provide a prediction or probability that the original poster is one of three classes, Human, Cyborg, or Bot [25]. The difference between the classifications being entirely human, computer assisted posting, and computer automated. A dominant feature in Chu's [25] research is Account Reputation, which attempts to measure the likelihood that an account is considered a bot. Account Reputation is defined by the following equation, Equation 2.1, and measured between zero and one. Follower Count is the number of followers of a Twitter account and Friend Count is the number of Twitter accounts a user follows:

$$Account\ Reputation = \frac{Follower\ Count}{Follower\ Count + Friend\ Count} \qquad \text{(Eq. 2.1)}$$

A famous person, with a high follower counter and low friend count, would score relatively high on Account Reputation. By comparison, A bot would have a high friend count and fewer followers. According to Chu’s findings, bot accounts rarely have a greater reputation than 0.5. Chu also asserts that a semi-automated account, or Cyborg, will generate a larger volume of Tweets than a Human based account. Perhaps unexpectedly, a Bot may generate fewer Tweets than a Human account [25] over its total lifetime. It was shown that a Bot will show more activity during its window than a

Human account, take longer hiatuses, and is more subject to Twitter suspensions and removal.

2.6 Research Opportunities

As reviewed in previous sections many methods have been explored for social media filtering and clustering. However, these methods fail to effectively evaluate intra-set clustering and denoising methods in a social media context. Clustering systems are either built on non-Twitter training sets [46], or from small original datasets that are sourced through Twitter's Standard API [17] [22]. In addition, many clustering algorithms focus on inter-set clustering instead of intra-set clustering [17] [40]. Furthermore, research was commonly carried out on readily available dataset instead of industry relevant ones. For example, I-TWEC [17] tests against a set of 60k Tweets accessed through the Twitter Streaming API which does not guarantee a representative sample of Twitter content. In addition, intra-set clustering was not extensively explored for geo-based searches, nor were clustering options explored in the context of an industry-relevant search, which can serve at most 500 posts for a single search or in a real time streaming environment. Finally, there have been advancements in the string similarity space, specifically T-codes and T-information, that have not yet been explored for its effectiveness in clustering Twitter content.

2.7 Chapter Summary

Several methods have been explored for effective social media data clustering and filtering. There also appears to be a lack research into industry-relevant dataset and subject to industry constraints.

Ultimately, there is a need for research to be carried out to understand the effective clustering of industry-relevant searches from a large, high fidelity sample using both conventional and more recent string similarity measures.

Chapter 3

3 Methodology

This chapter discusses the methods used to build and characterize the datasets used in this research, the data sanitation operations, the analysis toolset and metrics, the data characterization, the clustering methodology, and the industry-based constraints that were tested against.

3.1 Social Media Data Acquisition

Before pursuing clustering methodologies, a suitable corpus of social media content needed to be created. Twitter content was selected as the primary social media content for this evaluation due to its accessibility, industry relevance, and depth of content. To effectively research clustering techniques in an industry relevant context, an industry relevant data set was developed. Twitter's free Streaming API does not guarantee data fidelity [20], therefore, access to it's Enterprise Data API was required.

Through industry partner, Echosec Systems [2], access to search content from the Enterprise API was possible. Echosec's data access is representative of an industry organization that requires high fidelity Twitter content. While Echosec's specific relationship with Twitter is confidential, it is not exclusive in nature and can be recreated by other organizations.

3.1.1 Data Acquisition Method

The Echosec platform allows users to define search queries and retrieve Twitter and other social media content for consumption. Echosec has several different search capabilities that search across different features common to social media including location, keyword and username. For the purposes of this research, both location and keyword searches were used.

Echosec's location-based search gathers content from any region in the world based on a user defined geo-fence. Using standard drawing tools, user can input a location and Echosec will format an API query to each of its social media partners then collate the results returned. Alternatively, users can input an address, city, or landmark into Echosec's search bar. The Echosec platform will then interpret the location using a geocoder and draw a suitable boundary around the specified region format the social media query. The automated geo-fencing capability was used to standardize search sizes for the purposes of this research. An example Echosec search of Vancouver, Canada is shown in Figure 3.1.

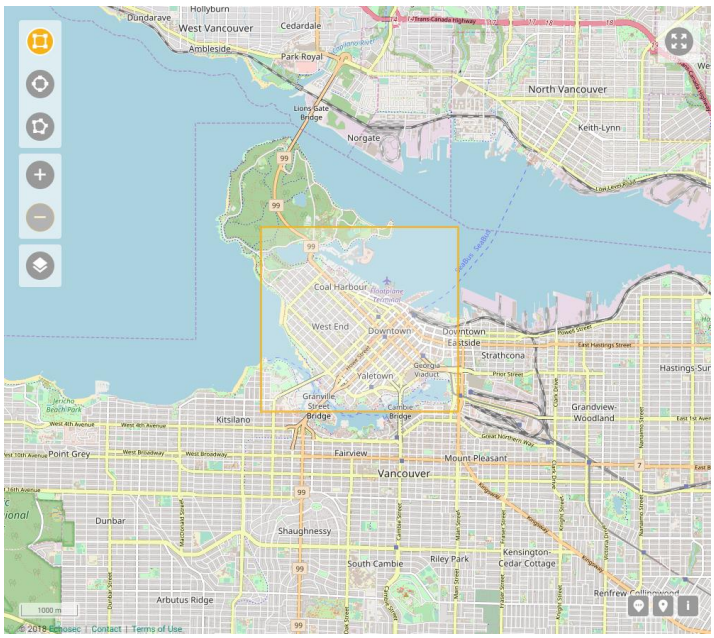

Figure 3.1: Example Vancouver Location Search on Echosec Map

Echosec's keyword search, similarly, gathers content that matches a user defined search term. Importantly, a keyword will return both the keyword and the matching hashtag. For example, the keyword search for 'food' will return posts that contain the word 'food' as well as posts containing the hashtag '#food.' However, the keyword search for '#food' will only return content that contains the hashtag. For the purposes of this research, queries did not include a hash (#) in the search queries and the results include content containing the keyword, the hashtag, or both. Example posts for the keyword 'hiking' are shown in Figure 3.2 and Figure 3.3.

Figure 3.2: Hiking Example with Hashtag

Figure 3.3: Hiking Example Without Hashtag

The Echosec platform can translate both location and keyword searches into either historical searches or real-time streaming searches for Twitter's Enterprise Data API. Over the duration of a real-time data search, Echosec retains content that can then be used for analysis and export. To build the research content corpus, Echosec real-time searches were generated then exported after each event or an appropriate amount of time. Importantly, Echosec does not return Re-Tweeted (RT), so the corpus will only contain original tweets.

Through Echosec Systems Ltd, Twitter content was aggregated from the high-fidelity Enterprise API. A number of saved search queries were constructed to retrieve and record various datasets that represent common corporate security, marketing, and journalism searches. These search queries included both keyword-based searches and location-based searches. Search queries were generated using the Echosec User interface, pictured below in Figure 3.4. Specifically, the search bar was used for queries, as can be seen in Figure 3.5.

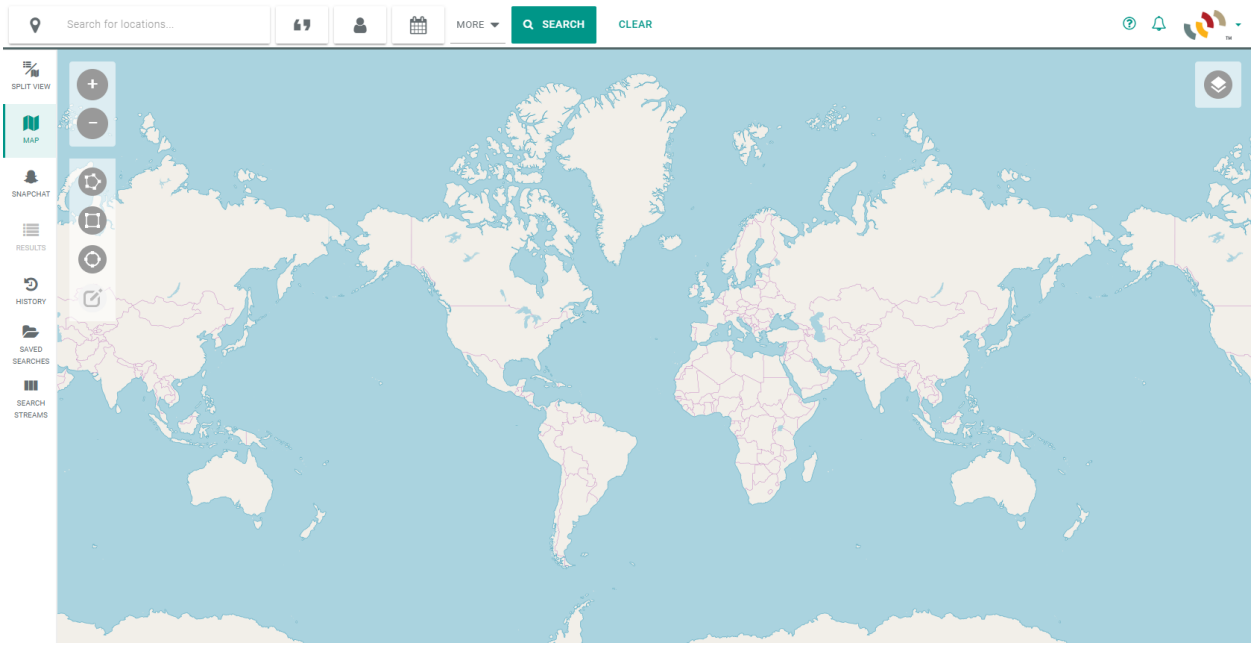

Figure 3.4: Echosec User Interface

Figure 3.5: Echosec Search Bar

3.1.2 Data Acquisition and Selection

Each query was run and then exported from the Echosec system. Searches represented a wide range of datasets including sporting activities, cultural events, and various metropolitan areas. Table 3.1 lists each of the searches recorded including the search topic, search type, and the number of posts retrieved.

Table 3.1 Table of Searches

Search	Seach Type	# of Tweets (Thousands)
Worldcup	Keyword	6793
Superbowl	Keyword	1625
RoyalWedding	Keyword	1105
Eagles	Keyword	332
Patriots	Keyword	236
StanleyCup	Keyword	161
MeToo	Keyword	126
Vancouver	Keyword	80
Memorial Day	Keyword	51
Florida	Location	5895
Seattle	Location	1283
London	Location	982
Chicago	Location	308
New York	Location	79
Vancouver	Location	29
Longbeach	Location	15

To focus research efforts, a selection of searches was made to best represent keyword and location-based searches and to draw meaningful comparisons between each. To that end, the location-based searches selected were London and Vancouver. Each search represents a location of interest with differing data throughputs and in different regions of the world. London represents a high data throughput and varied content, whereas Vancouver is lower throughput and less varied content. For keyword searches, Royal Wedding and World Cup were chosen. Both the Royal Wedding and World Cup events happened at an international scale, however, each had a distinct audiences and different period of relevance. The RoyalWedding search has a high volume of data over a short period of time and the WorldCup search exhibits longer, moderate to high volume over a longer period.

3.2 Data Composition

Each dataset was downloaded from Echosec's database as a SQL query. All personally identifiable information was removed at the time of download to avoid any privacy or legal considerations and to remain compliant with Echosec's terms of service.

For each search, the resulting social media posts comprised four components. A Post ID, a Timestamp, a User ID, and the Tweet content. The Post ID and User ID are randomized ID's used by the Echosec Platform. As a result, the posts and users are independently and uniquely referenceable but not reversible to identify a Twitter user. The time stamp corresponds to when Twitter acknowledged the creation the specific Tweet. Finally, the Post content contains the raw content of the Tweet including text, emoji's, and URLs. An example plain-text data schema and an associated content is seen in Figure 3.6:

Schema: “PostID,” “Timestamp,” “User ID,” “Post Content”

Content: "6341968195","2018-05-28 02:54:20","6343661122","'Back on the scene y\\'all #NHL #StanleyCup happy to be here https://t.co/CHLPnjcC42'"

Figure 3.6: Data Schema and Content

While the majority of content followed this format, outliers in the dataset required a data sanitation operation.

3.3 Data Ingestion and Sanitation

Different data ingestion, sanitation, and manipulation methods were tested for overall effectiveness and ability maintain data fidelity. Tools that were testing comprised native Python Comma Separated Value (CSV) manipulation tools [49], native Python data structures [50], the Python Data Analysis Library (*pandas*) [51] and various export methods from MYSQL [52].

3.3.1 Newlines and Punctuation in Social Media

Several social media content characteristics do not allow for the naïve manipulation of social media content. A naïve approach for data ingestion, using Python’s CSV read method, caused many row entries to fail. It is common practice among Twitter users to inject newline characters in a post for emphasis. Newline characters are a programmatic character that instructs a computer to start on the next line down and is commonly equivalent hitting the ‘enter key’.

Newline characters are also how a computer knows to read the next line in a CSV document. These newline characters were causing issues for the CSV file reader implemented by Python. A CSV reader would incorrectly assume it had read in the

complete Tweet and fail to read the full content. In addition, the reader would then ‘start’ ingesting the remaining content as a newline, representing a new Tweet. This remaining content, however, did not match the intended schema and caused a cascade of errors. This compounding effect was often not recoverable, and the rest of the file would be read erroneously. This newline effect was also experienced, with less severity, with other punctuation and emojis. For example, the following Tweet in Figure 3.7 caused Python’s native CSV reader to fail.

Figure 3.7: Tweet with Newline Emphasis

In another example, seen below in Figure 3.8, a newline character in the Tweet caused the CSV reader to treat a single Tweet as multiple Tweets. Figure 3.9 is the properly handled tweet by comparison and contains the PostID, Date, UID and Content in appropriate order.

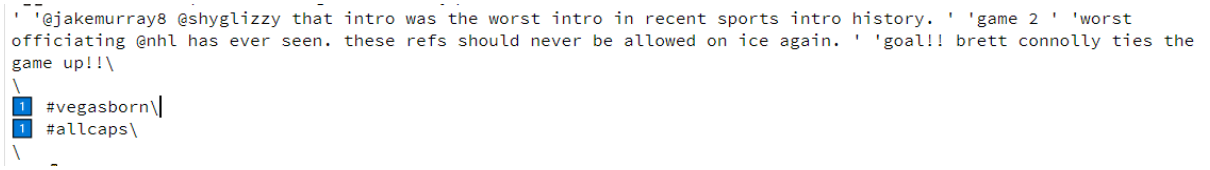

Figure 3.8: Example of Incorrectly Read Tweet CSV

```
"6346260073","2018-05-29 03:07:43","4912100777","'Worst
officiating @nhl has ever seen. These refs should never be
allowed on ice again. #StanleyCup'"
```

Figure 3.9: Example of Correctly Read Tweet CSV

3.3.2 Ingestion and Sanitation Tools

Several different methods were used to combat the newline and punctuation issues including command line operations, MYSQL CSV Export, Pandas data frames, and MYSQL databases.

Initially, an open source command line tool called CSV Master (CSVM) [] was used to export content from SQL files to a CSV. This was found to fail for escaped quotes and newlines. A second method was attempted by searching and replacing characters in the text file, specifically, \" and "" were exchanged in order to sanitize the files before reading it into a CSV format. This method was found to cause real Tweets containing those characters to be misrepresented in the analysis and consequently discarded. Finally, MYSQLs 'TO OUTFILE' command was used, with additional operations to appropriately transform the Echosec based MYSQL queries into appropriately formatted CSVs. The MYSQL operations in Figure 3.10 were appended to each query to format the CSVs.

```
INTO OUTFILE '/var/lib/mysql-files/filename.csv' FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n';
```

Figure 3.10: MYSQL Into Outfile Code

Finally, a local recreation of the Echosec database was explored to manage data. A MYSQL Docker [53] container was built and the corresponding Echosec searches were loaded into the container.

3.3.3 Data Manipulation for Analysis

For data manipulation and analysis, both Python's native data structures and Pandas (Python Data Analysis Library) were explored. Python's native structures offered a significant amount of flexibility but were cumbersome. Conversely, Pandas and its library of tools were well-aligned with the analysis requirements.

It was determined that a simple MYSQL to CSV export, followed by ingestion into Pandas removed the majority of erroneous data points caused by newlines and other punctuation. Even with this improvement, content rows still existed that consisted of a null character and were incorrectly represented in the data frame. To sanitize these null characters, a Pandas method was used to drop rows containing null characters from the table. An example Pandas data frame row that properly deals with an escaped character is shown in Figure 3.11.

14973	6345906106	2018-05-29 01:05:55	3222236228	'RT @NHLonNBCSports: THIS GAME IS CRAZY\\n#...

Figure 3.11: Pandas Data Frame Content

While this form of data sanitation was encouraging and offered a simple method for producing results, the removal of content reduced the overall data fidelity of the

analysis. As a result, the Pandas data library was tested with a direct MYSQL read from a local database. This method offered the least number of errors and provided the highest data fidelity. No further data sanitation operations were necessary using this method.

3.4 Data Analysis Toolset

Once a method for data ingestion had been determined, the core toolset for analysis was developed. All infrastructure was maintained in Docker containers [53] to ease recreation on alternate systems. Search data was housed in a MYSQL database within its own Docker container. A second, linked Docker container housed the analysis code. Scripting was accomplished using Python3 in a Jupyter Notebook [54], which allowed for the use of *Pandas*' data analysis library. All plots were drawn using the Python library Matplotlib [55]. This tool set was found to have a quick setup and maintained high data fidelity.

3.5 Data Characterization

Each data set was characterized using several techniques to best understand what was represented in each original search.

3.5.1 Primary Hashtags

For each search, the top ten hashtags by frequency were determined to confirm the data sets matched the expectations of conducted queries. The top hashtag for each search could later reviewed to understand how it may have biased the clustering results. For each dataset, a random sample of fifteen thousand Tweets was analyzed for the ten most frequent hashtags. The built-in P*andas* method *sample* was used to generate a random sample of Tweets. To find each hashtag in a post, a regular

expression was utilized. Each occurrence of a hashtag in a post was recorded along with its count. The top ten hashtags and their frequencies were then displayed using a Matplotlib histogram.

Example plots of the top ten hashtags from the Vancouver, Worldcup, RoyalWedding , and London searches can be seen in Figure 3.12, Figure 3.13, Figure 3.14, and Figure 3.15, respectively.

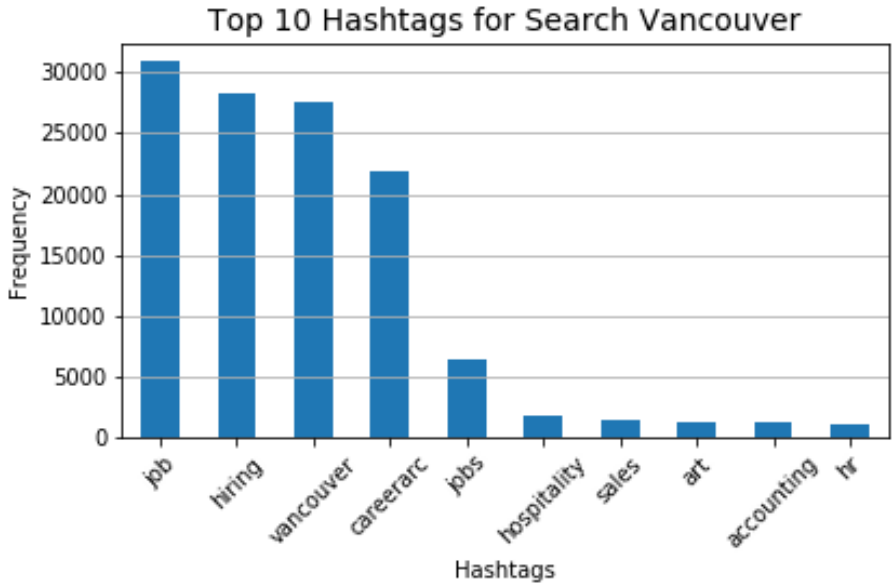

Figure 3.12: Top ten hashtags for Vancouver

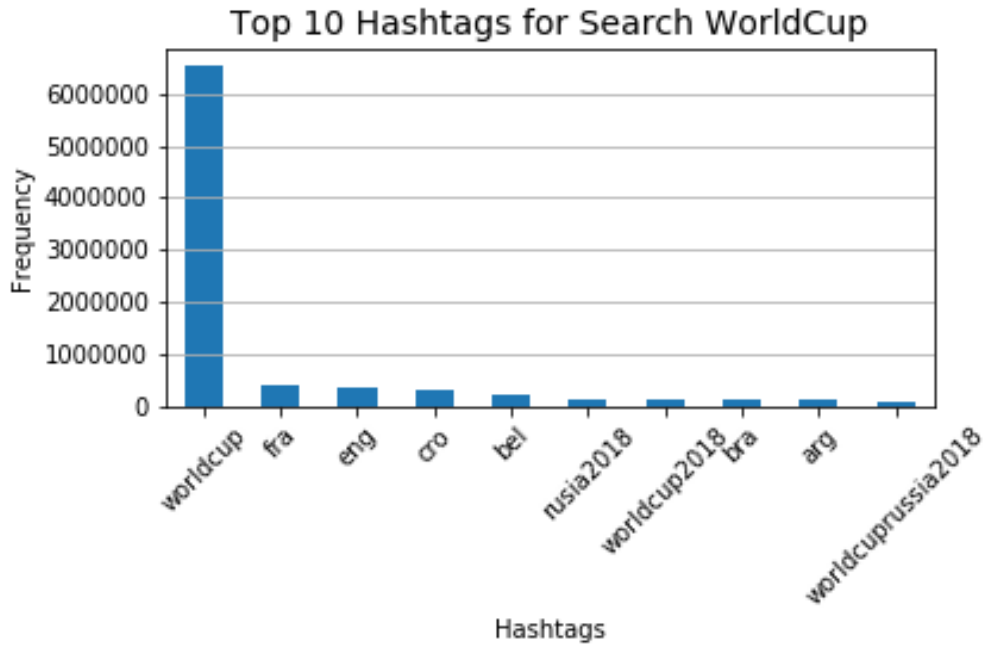

Figure 3.13: Top ten hashtags for WorldCup

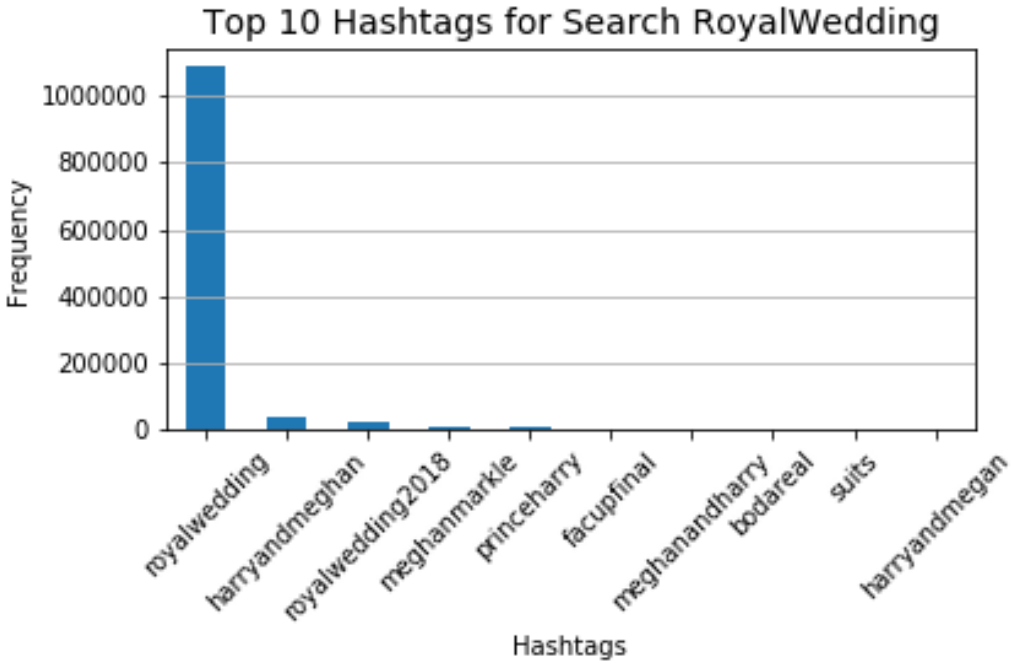

Figure 3.14: Top ten hashtags for RoyalWedding

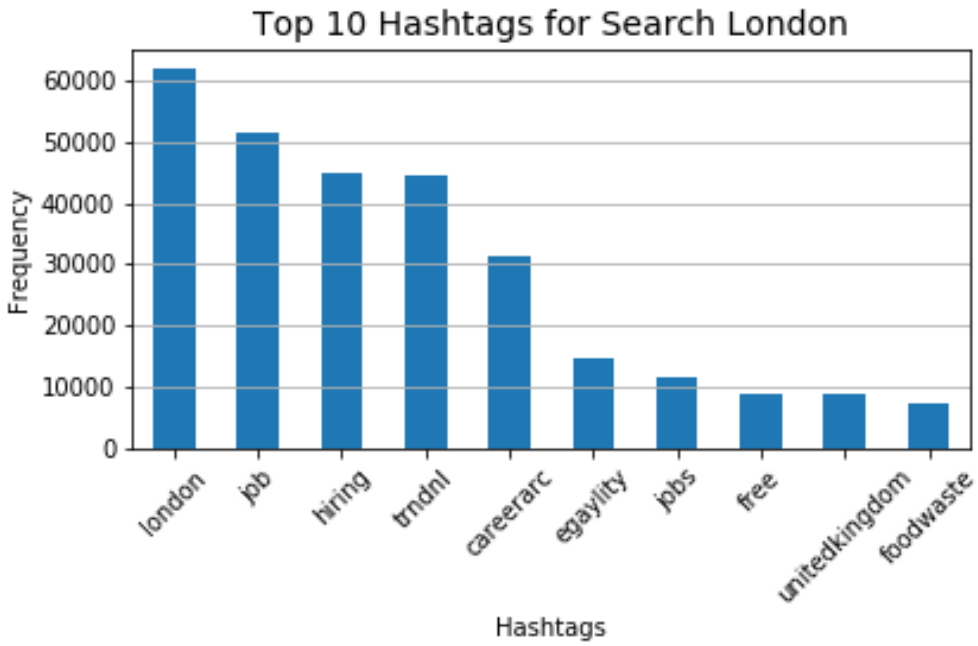

Figure 3.15: Top ten hashtags for London

3.5.2 Primary Term Composition

Word clouds of Tweet content were generated to further understand the general composition of each search. To do this, samples were taken of each search comprising 10% of the total number of tweets. The content of all the tweets in the sample set were then aggregated into a large string. The string was then passed to a Python based word cloud generator [56]. In addition to stop words, common Twitter link strings “https” and “co” were removed from the word clouds. For RoyalWedding and Worldcup searches the primary hashtag was also removed. Word clouds for Vancouver, London, RoyalWedding, and WorldCup can be seen in Figures 3.17, 3.18, 3.16, and 3.19, respectively.

Figure 3.16: Vancouver Word Cloud

Figure 3.17: London Word Cloud

Figure 3.18: RoyalWedding Word Cloud

Figure 3.19: WorldCup Word Cloud

3.5.3 Length of Tweets by Term and by Characters

Each dataset was also characterized by the number of terms and the number of characters present in each of its Tweets. During the data acquisition, the Echosec Twitter Enterprise API only supported 140 characters, despite the application natively supporting 280 for the majority of 2018, as a result each Tweet is capped at 140 characters [18] [21]. Very short and very long Tweets can both bias clustering results, therefore, it was necessary to understand the statistical distribution of Tweet lengths. Similar to the hashtag analysis, the number of terms and the number of characters were counted using native Python methods for each post in the data set then plotted as a Matplotlib histogram. Figures 3.20 - 3.23 show string token Tweet lengths for Vancouver, London, RoyalWedding, and WorldCup.

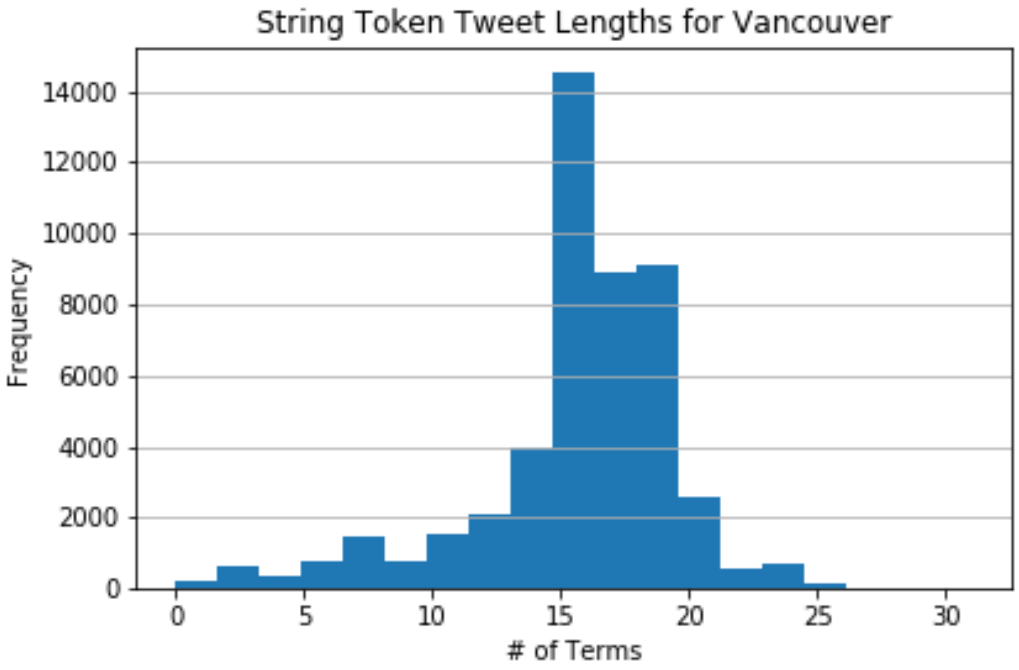

Figure 3.20: String Token Tweet Lengths Vancouver

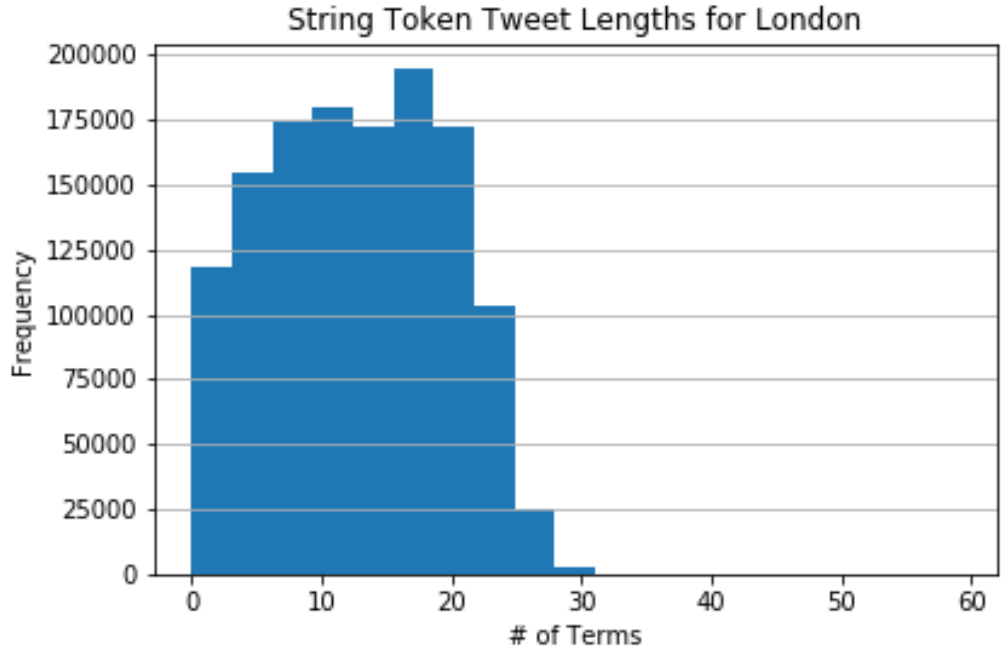

Figure 3.21: String Token Tweet Lengths for London

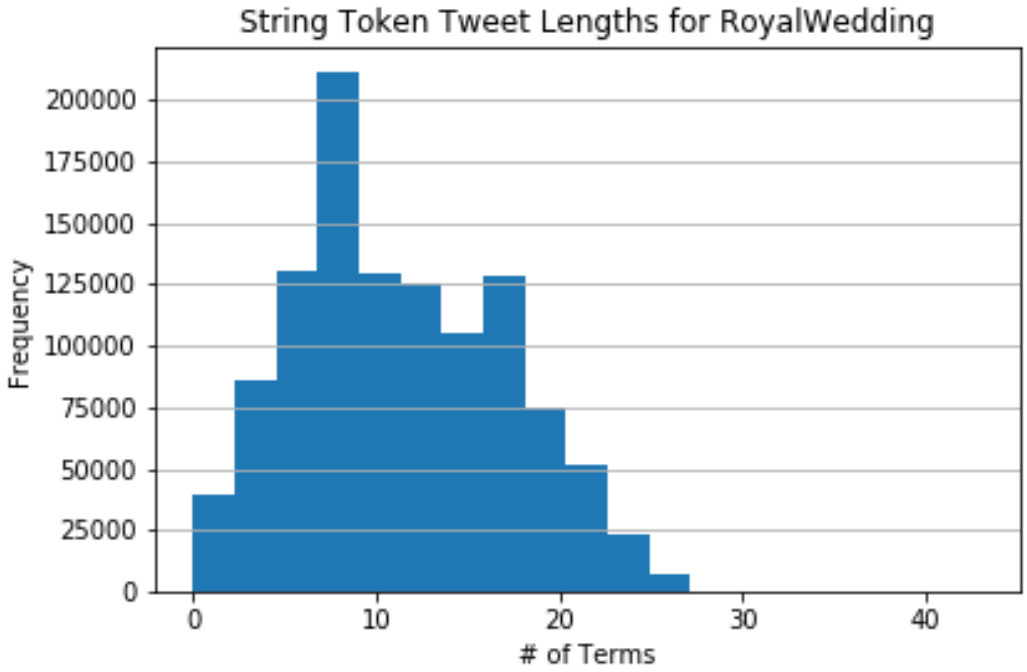

Figure 3.22: String Token Tweet Lengths RoyalWedding

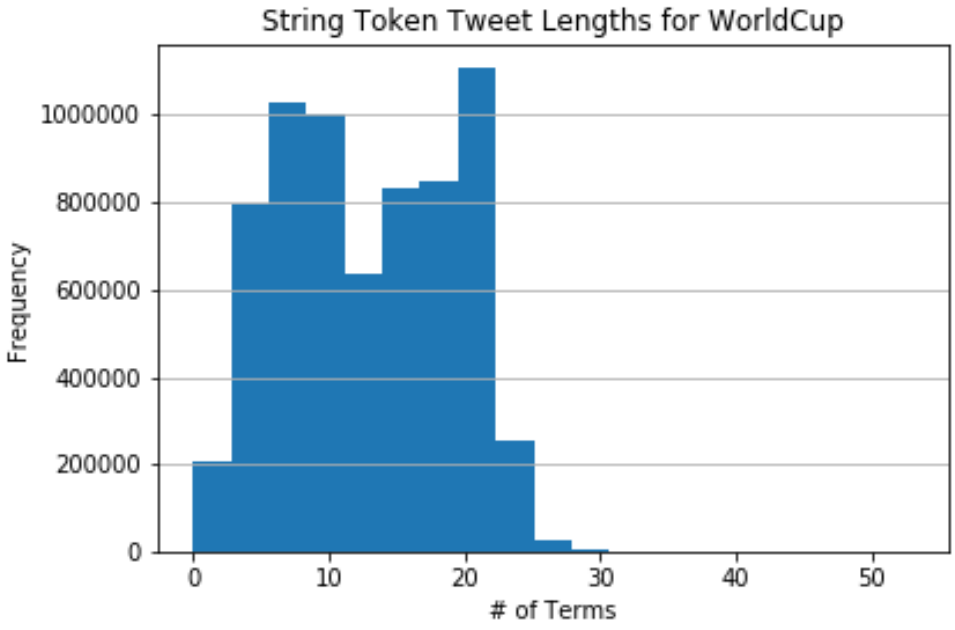

Figure 3.23: String Token Tweet Lengths World Cup

Figures 3.24, 3.25, 3.26, and 3.27 show character-wise Tweet lengths for Vancouver, London, RoyalWedding, and WorldCup, respectively. As can be seen in the figures, there is a truncation at the 140-character mark, which may bias the clustering results. As the purpose was to find similar tweets, truncation or no, it is expected that the bias will not significantly affect the results.

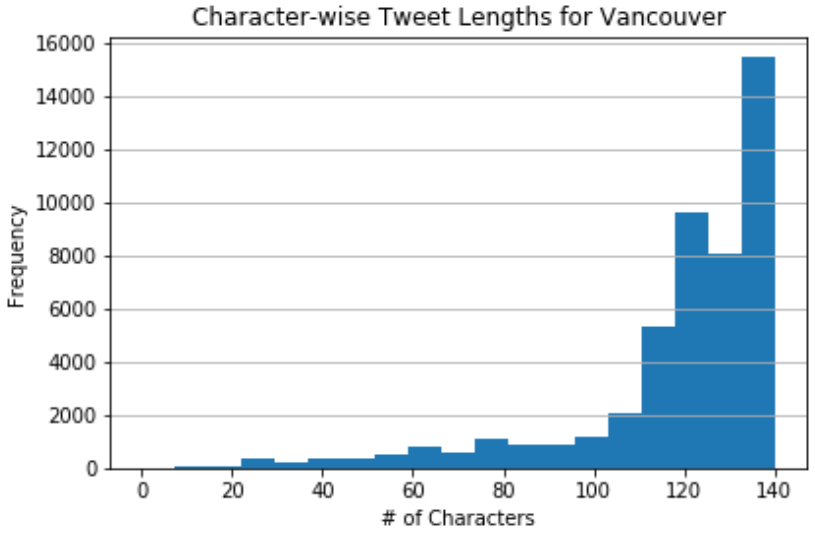

Figure 3.24: Character-wise Tweet Lengths Vancouver

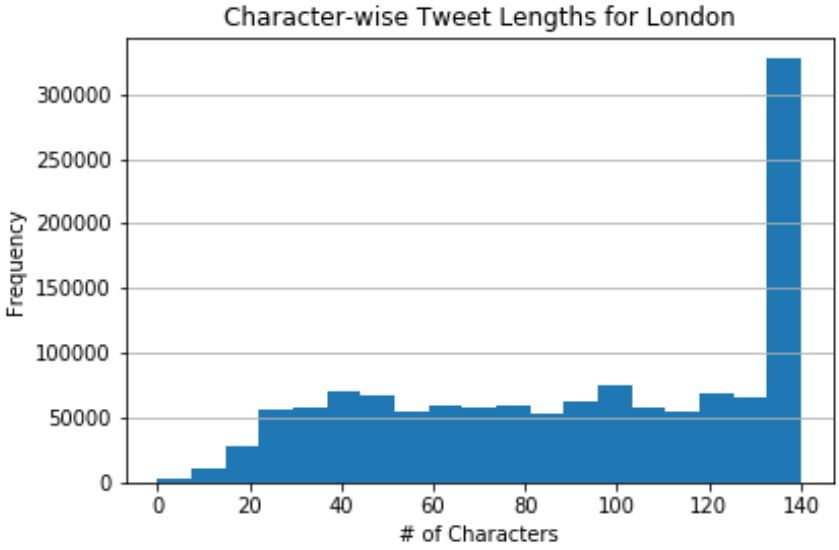

Figure 3.25: Character-wise Tweet Lengths London

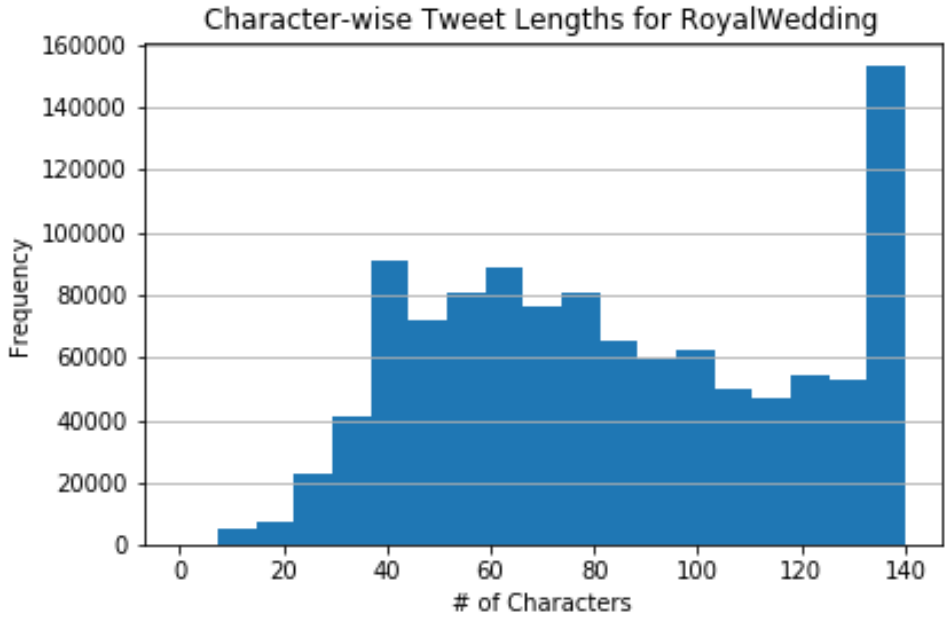

Figure 3.26: Character-wise Tweet Lengths RoyalWedding

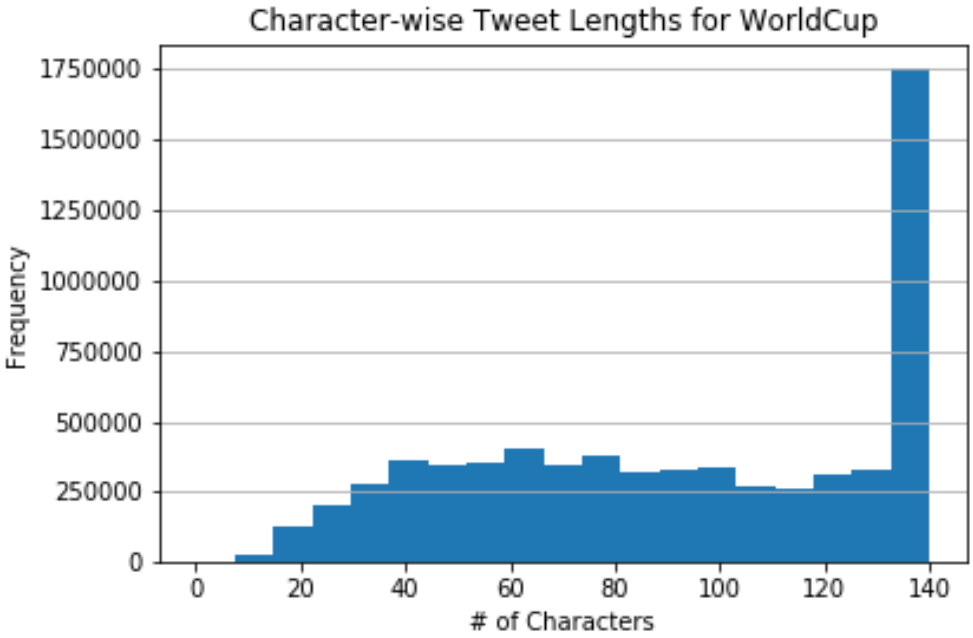

Figure 3.27: Character-wise Tweet Lengths WorldCup

Table 3.2 lists statistics for the String Token lengths of each search.

Table 3.2 String Token Statistics by Search

Search	Mean	Median	Std	*Z* = -1	*Z* = -2	*Z* = -3
Vancouver	15.63	16.0	3.84	11.79	7.95	4.11
London	12.86	13.0	6.55	6.31	-0.24	-6.79
RoyalWedding	11.44	11.0	5.72	5.72	0.0	-5.72
WorldCup	12.85	13.0	6.3	6.55	0.25	-6.05

Table 3.3 lists statistics for the Character-wise lengths of each search.

Table 3.3: Character-wise Statistics by Search

Search	Mean	Median	Std	*Z* = -1	*Z* = -2	*Z* = -3
Vancouver	118.41	125.0	24.33	94.08	69.75	45.42
London	91.35	96.0	40.35	51.0	10.65	-29.7
RoyalWedding	84.13	80.0	35.06	49.07	14.01	-21.05
WorldCup	92.11	93.0	38.89	53.22	14.33	-24.56

3.5.4 Dataset Time Period and Post Frequency

Finally, each dataset was analyzed for the post frequency during the time that it was active. Post frequency across common searches allows for the understanding of how standard industry search queries behave.

The Post Frequency characterization was accomplished by binning all Tweets according to the date they were posted. Posts were binned on a per day basis using 0:00 hours UTC as the start and end of each day. The count for all posts in each bin was then displayed as a histogram. Figures 3.28 - 3.31 show the post frequencies of Vancouver, London, RoyalWedding, and WorldCup. RoyalWedding exhibits unique behaviour in that the vast majority of activity occurred on a single day.

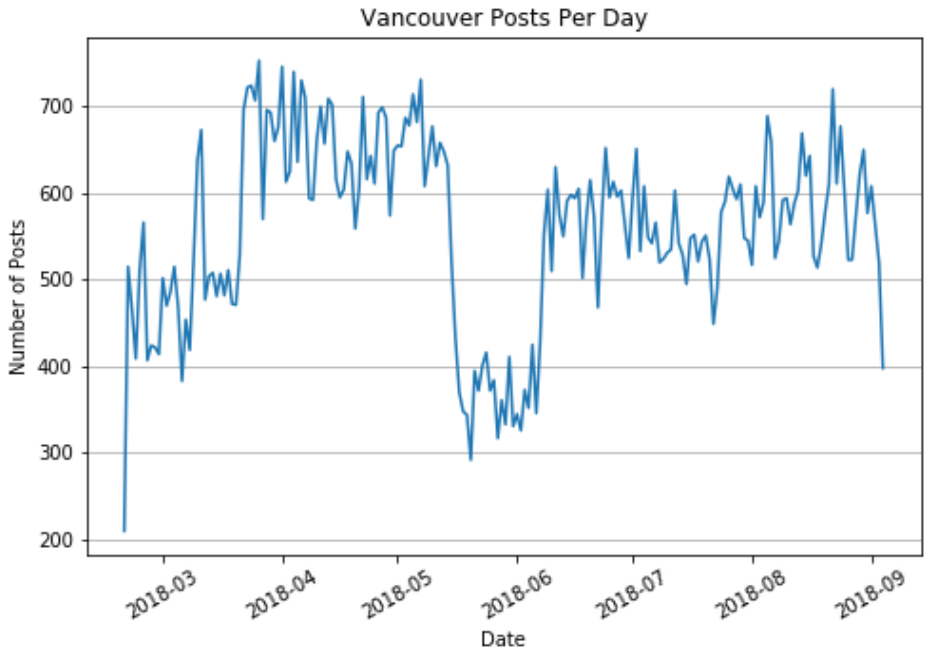

Figure 3.28: Vancouver Post Frequency

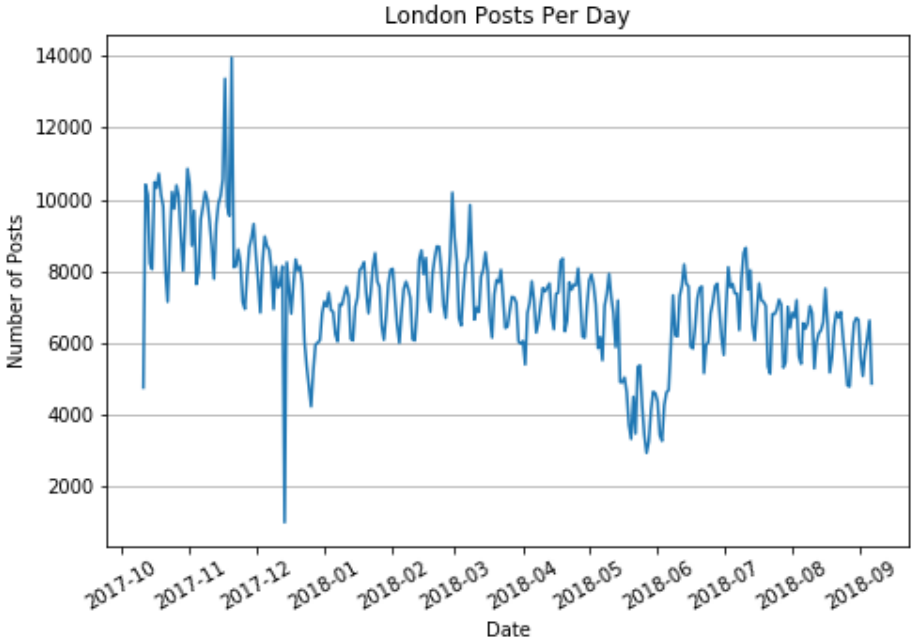

Figure 3.29: London Post Frequency

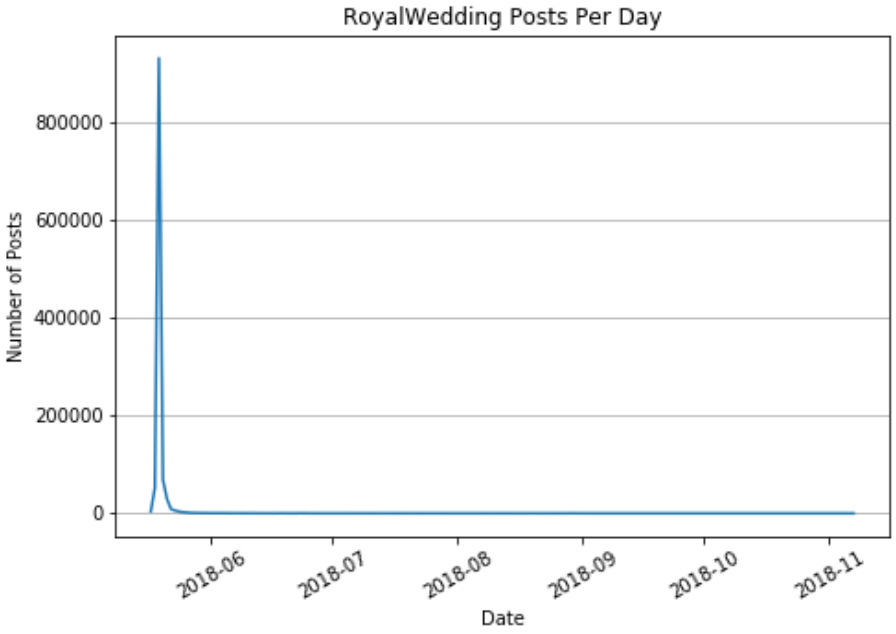

Figure 3.30: RoyalWedding Posts Frequency

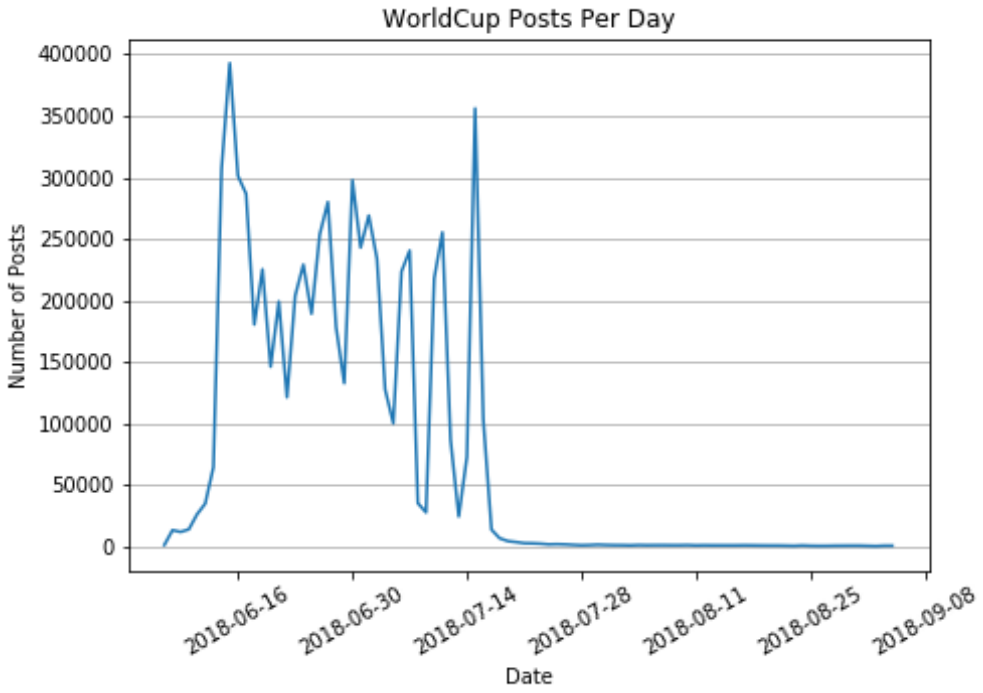

Figure 3.31: WorldCup Posts Frequency

Table 3.4, Table 3.5, Table 3.6 list Maximum, Mean, and Minimum number of posts per day shows for each search, respectively. As can be seen in Table 3.4, the highest daily rate of posts occurred in the topic searches of WorldCup and RoyalWedding at 109 and 258 posts per second, respectively.

Table 3.4: Maximum Tweets Per Day

Search	**Maximum Tweets Per Day**	**Maximum Daily Rate (Posts Per Second)**
WorldCup	392944	109
RoyalWedding	930116	258
London	13955	3.88
Vancouver	753	0.21

Table 3.5: Mean Tweets Per Day

Search	Mean Tweets Per Day	Mean Daily Rate (Posts Per Second)
WorldCup	457	21.0
RoyalWedding	9908.0	2.75
London	7174.0	1.99
Vancouver	558.0	0.16

Table 3.6: Minimum Tweets Per Day

Search	Minimum Tweets Per Day	Minimum Daily Rate (Posts Per Second)
WorldCup	457	0.15
RoyalWedding	1	0
London	1010	0.28
Vancouver	210	0.06

3.5.5 Similarity Measure Testing

Similarity measures are the core measurement used to cluster Tweets. Each measure compares two Tweets and outputs a result that represents how similar or how different the Tweets are from each other. Commonly a similarity score is between 0 and 1, where 0 is exactly similar and 1 is dissimilar. Clusters are then formed by grouping Tweets that are similar. The efficacy of similarity measures on two Tweets will affect the quality of the final clusters.

An initial test was developed to observe the efficacy of each proposed similarity measure. The purpose of this test is to disqualify similarity measures that do not perform for basic variations of a Tweet. For this test, a single Tweet was modified with obvious and common social media alterations that could change the similarity score. The original Tweet was then compared to each new, modified Tweet using the similarity measures. For this test, a Tweet was selected that comprised many standard content features including words, hashtags, and a link. The modifications made to this example

Tweet, while in context of the Tweet, are independent of the Tweet content. As a similarity measure will only be removed from consideration for gross failure, similar results would be obtained if different example Tweets was used. The modifications and justifications for each are found in Table 3.7.

Table 3.7: Tweet Modifications and Justificitations

No.	Modification Type	Justification
0	None	Establish a baseline
1	Observably Different Tweet	Establish a baseline
2	Additional Username	Send a Tweet to a Friend
3	Add a hashtag	Personalize a Tweet.
4	Link Modification	Shortened URLs are commonly different, but the Tweet and landing page are the same.
5	Deletion	Delete a hashtag or username you do not want to promote
6	Emoji Addition	Reacting to another Tweet with Emojis
7	Adding Quotations	Add quotes to reference another Tweet
8	Typographical Error	Error in re-writing Tweet content
9	Common Autocorrect	Common Errors as a result of auto correct
10	Abbreviated Speak	Shortened Text that means similar

The Tweet in Figure 3.32 was selected at random from the Vancouver search data set to use for the similarity testing.

Figure 3.32: Similarity Test Example Tweet

Modifications were made to the example Tweet as outlined in Table 3.7 and the resulting Tweets are seen in Table 3.8.

Table 3.8: Modifications and Resulting Tweets

No.	Modification	Resulting Content
0	None	Can you recommend anyone for this #job in #Vancouver, BC? bit.ly/2Kby8sG #security #Hiring
1	Different	A real privilege to meet courageous Ben and an incredibly proud moment to receive the World Cup golden boot. All focus now on the bigger, team prize in 2020. #ThreeLions #England
2	Add User	**@jimmyp** Can you recommend anyone for this #job in #Vancouver, BC? bit.ly/2Kby8sG #security #Hiring
3	Hashtag	Can you recommend anyone for this #job in #Vancouver, BC? bit.ly/2Kby8sG #security #Hiring **#seriously**
4	Link	Can you recommend anyone for this #job in #Vancouver, BC? bit.ly/**13asXy1** #security #Hiring
5	Deletion	Can you recommend anyone for this #job in #Vancouver, BC? bit.ly/2Kby8sG ~~**#security**~~ #Hiring
6	Emoji	Can you recommend anyone for this #job in #Vancouver, BC? 😂😂😂bit.ly/2Kby8sG #security #Hiring
7	Quotations	**"**Can you recommend anyone for this #job in #Vancouver, BC?**"** bit.ly/2Kby8sG #security #Hiring
8	Typo	Can you recommend anyone for **thsi** #job in #Vancouver, BC? bit.ly/2Kby8sG #security #Hiring
9	Autocorrect	Can you **reconnect** anyone for this #job in #Vancouver, BC? bit.ly/2Kby8sG #security #Hiring
10	Abbreviated	Can **u** recommend **any1** for this #job in #Vancouver, BC? bit.ly/2Kby8sG #security #Hiring

The similarity measures tested were Hamming distance, Levenshtein distance, Jaccard similarity, and T-Information distance.

3.5.5.1 Hamming Distance

Hamming distance was the first measure explored for Tweet similarity. The Hamming distance measures the edit distance between strings of an equal length [30]. An element-wise difference at the same position in a string is counted as ‘1.’ Each difference is summed to calculate the Hamming distance between two strings. The Hamming distance requires strings to be of the same length, and consequently, does not account for omissions, deletions, or additive differences in elements between two strings unless one of the two strings is padded. Two element types were tested with Hamming distance, both characters and string tokens.

For character-wise Hamming distance, each element was compared in with the corresponding element in a second string from the first position, position 0, to the final position of the string when reading from left to right. For every difference between corresponding elements, a counter was iterated and ultimately summed to finalize the Hamming distance between the strings. A Hamming distance of zero indicates an exact character-wise match of two strings. As the Hamming distance increases it indicates an increasing variation between strings, maximizing at the total string length. An example of a character-wise Hamming distance calculation for several examples can be seen in Figure 3.33.

Hamming(‘abc’, ‘aba’) = 1

Hamming(‘kings’, ‘rinse’) = 3

Figure 3.33: Character-wise Hamming Distance Examples

A second method of Hamming distances was implemented using string tokens. To generate the comparison, the content of each social media post was broken into space separated tokens. Each Tweet was broken into N substrings commonly comprising words, usernames, hashtags and links. Importantly, each string token was subsequently treated as an individual, unique element. When each element was compared to the corresponding string token from a different social post, a determination was made as to whether it was exactly matching.

Hamming('This is a string.', 'This is a sentence.') = 1

Hamming('The quick red fox.', 'This lazy brown dog.') = 3

Figure 3.34: Example String Token Hamming distance

In both cases, the Hamming distance was normalized by the length of the Tweets to get a distance from 0 to 1, where zero represented an exactly similar Tweet. The resulting equation is seen in Equation 3.1, where dHamming is the normalized Hamming distance, and the Hamming function calculates either the character-wise or string token Hamming distance, TweetA and TweetB are Tweets of the same, non-zero length, and |.| is the standard set cardinality operator.

$$dHamming(TweetA, TweetB) = \frac{Hamming(TweetA, TweetB)}{|TweetA|} \qquad \text{(Eq 3.1)}$$

3.5.5.2 Levenshtein Distance

The Levenshtein distance also measures the edit distance between two strings. The Levenshtein distance, however, works for cases involving strings of a dissimilar length. Specifically, The Levenshtein distance finds the number the number of character replacements, insertions, or deletions it would require making a second string match the first.

Similar to the Hamming distance, the Levenshtein distance measure was used to calculate the distances between character elements and token elements between social media posts to determine its similarity. Levenshtein distance has a maximum distance equal to the length of the larger string. As a result, the distances measured were normalized to the length of the largest string to get a value between 0 and 1 as can be seen by Equation 3.2 below. In Equation 3.2, TweetA and TweetB are Tweets of non-zero length, |.| is the set cardinality operator, max() calculates the maximum of two values and levdist() calculates the character-wise or string token Levenshtein distance between two tweets. It's noteworthy that normalization by maximum tweet size could bias the results to smaller distance values.

$$dLevenshtein = levdist(TweetA, TweetB) \,/\, \max(|TweetA|, |TweetB|) \quad \text{(Eq. 3.2)}$$

3.5.5.3 Jaccard Distance

The Jaccard Similarity measure, Jaccard Coefficient, or Jaccard Distance compares the similarity of two unordered sets using the intersect and the union of the sets [33]. Jaccard Similarity tested using both characters and string tokens for effectiveness. The Jaccard Distance is calculated using Equation 3.3. In Equation 3.3, A and B are sets comprising all characters or all string tokens from TweetA and TweetB, and |.| is the cardinality operator:

$$dJaccard(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} \qquad \text{(Eq 3.3)}$$

Effectively, the Jaccard distance is measured from 0 to 1, where 1 represents two dissimilar sets and 0 represents two exactly similar sets. The Jaccard distance allows for the measure of similarity between two sets of unequal size.

3.5.5.4 T-Information Distance

T-Information is a measure of string complexity generated by decomposing strings into base T-Codes [44]. T-information measures how much information is contained in each string as compared to the information in another string. The T-information for two strings, A and B, can be computed by effectively compressing String A using the basis strings generated by a T-code decomposition of another string, String B. The information distance then becomes a function of how well one string's base strings compress the other string. This compression system is robust to small variations and modifications in the strings, hence, T-information is robust to common issues in Tweets such as character level misspellings. T-information occasionally produces a result outside of the expected 0,1 bounds when measuring the similarity of two Tweet strings. It is suspected this is a result of how T-information handles emoji's and special characters.

3.5.5.5 Similarity Measure Implementation

Hamming Distance and Levenshtein Distance were implemented using an open source distance library [57]. Jaccard Distances was implemented natively in Python3. To analyze T-Information, N. Rebenich's FLOTT C-code implementation was used [44] [58], with Python bindings written by Michael Anderson [59].

3.5.5.6 Similarity Measure Independence and Performance

Each similarity measure was tested for performance based on both character-wise similarity and, where appropriate, string token similarity. Character-wise distance is measured using character-by-character analysis of string similarity. String token

distance is measured using string tokens instead. String tokens are generated by splitting the string into space separated tokens commonly comprising words, hashtags, links etc. All measurements are carried out using a left to right processing of the strings.

Each similarity measure was tested independently. Despite being measured from 0 to 1, each similarity distances exists within a different and distinct measure space and hence cannot be compared directly. Practically, this means given an example Tweet in a large sample of data, each similarity measure would resolve a different subset of Tweets for the sample set as similar, based on the original example tweet for the same set distance threshold.

The following figures were generated comparing the original tweet in Figure 3.32 to each Tweet in Table 3.8 for each similarity measure. Figure 3.35, Figure 3.36, Figure 3.37, Figure 3.38, Figure 3.39, Figure 3.40, and Figure 3.41 plot the similarity distance from 0 to 1 for each modification type for each similarity measure.

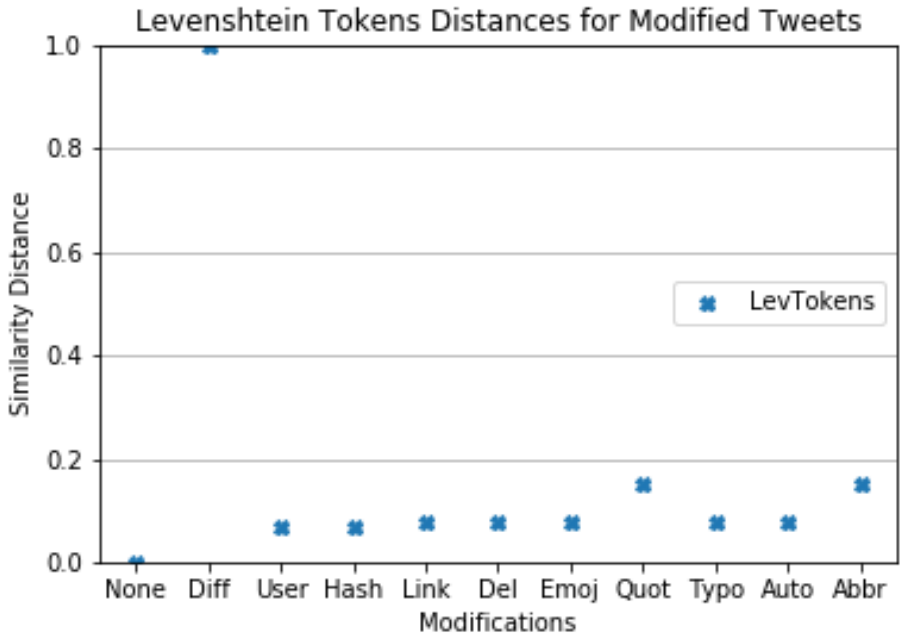

Figure 3.35: String Token Levenshtein Distances for Modified Tweets

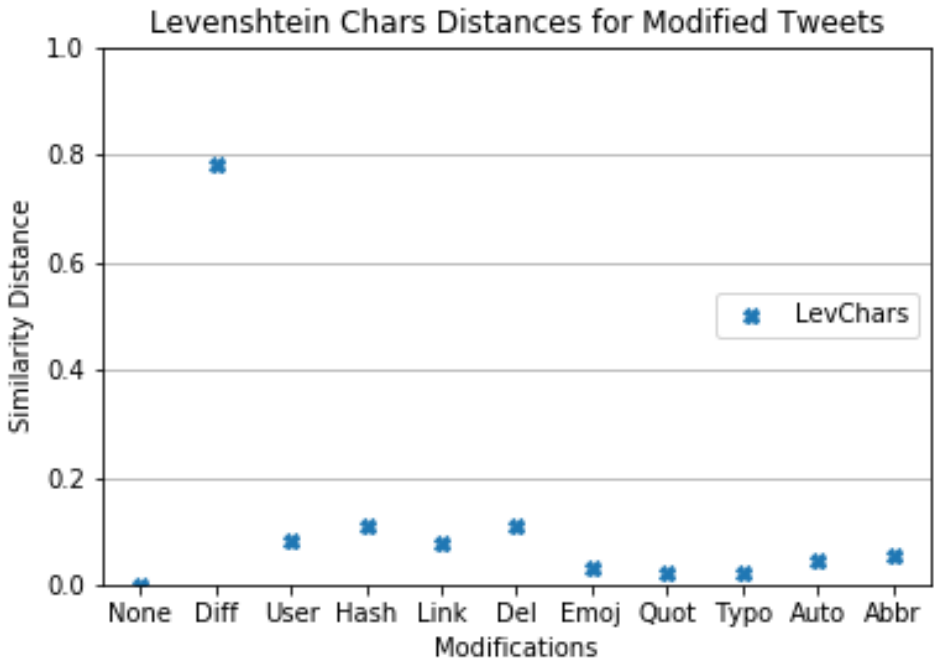

Figure 3.36: Character-wise Levenshtein Distances for Modified Tweets

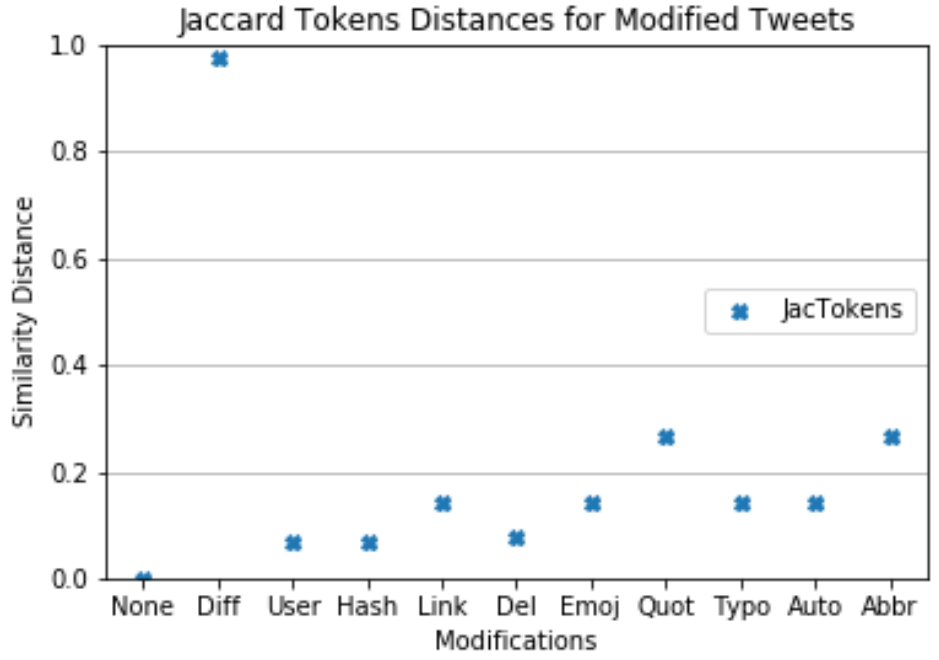

Figure 3.37: String Token Jaccard Distances for Modified Tweets

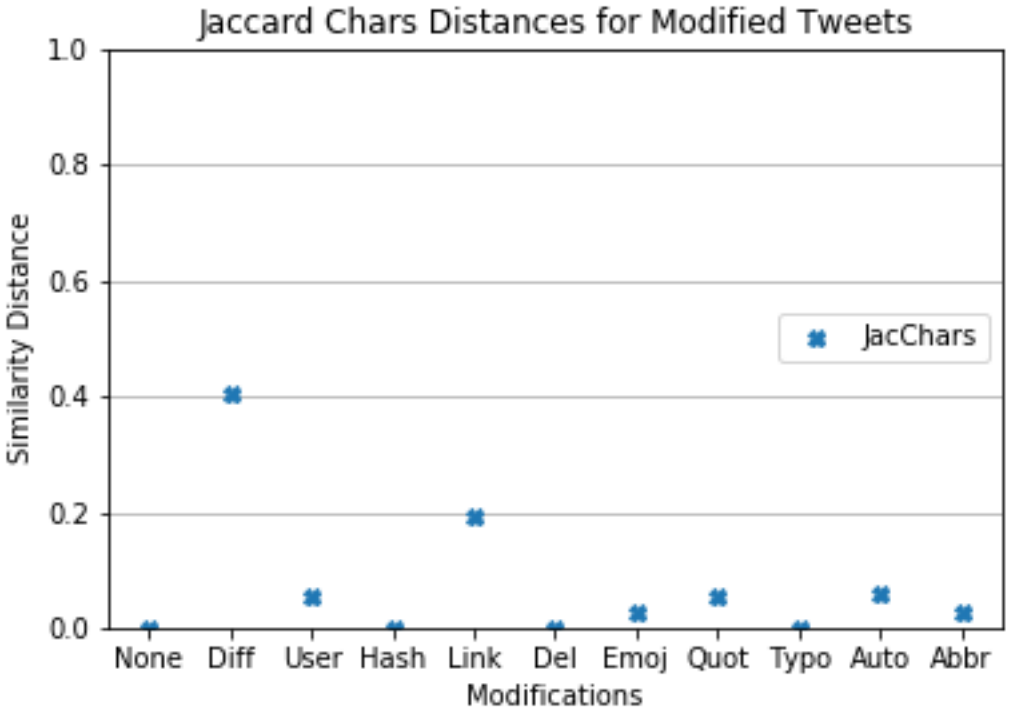

Figure 3.38: Character-wise Jaccard Distances for Modified Tweets

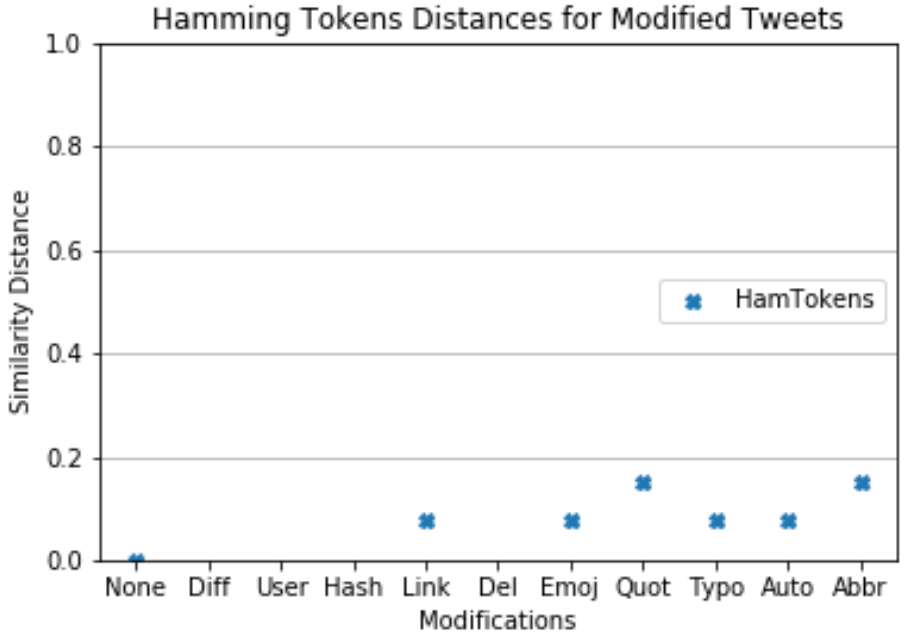

Figure 3.39: String Token Hamming Distances for Modified Tweets

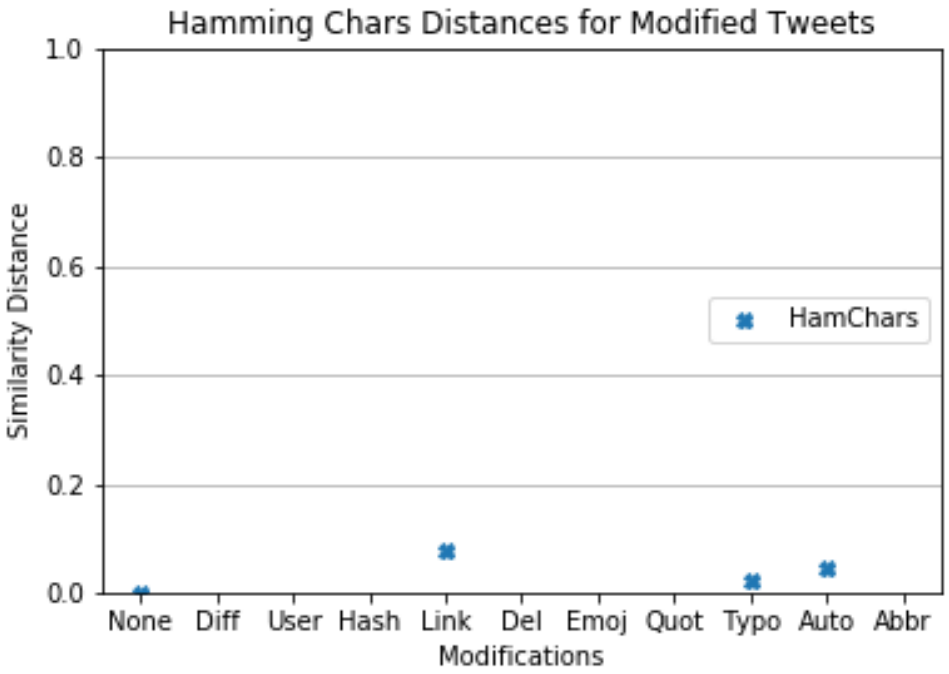

Figure 3.40: Character-wise Hamming Distances for Modified Tweets

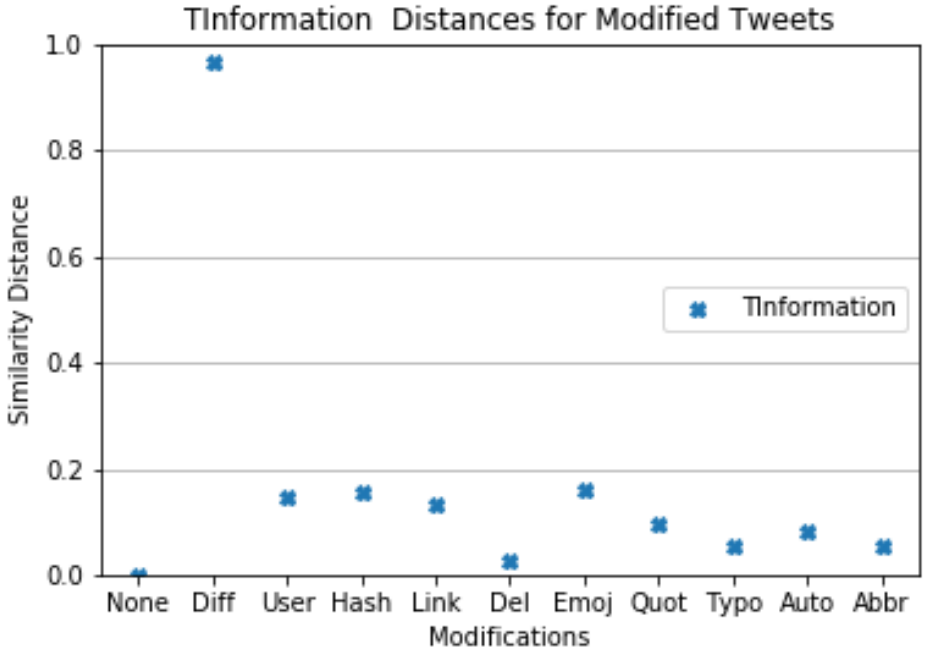

Figure 3.41: T-Information Distances for Modified Tweets

Figure 3.35 is the results for the string token Levenshtein similarity measure. The measure reasonably scores the unchanged Tweet and observably dissimilar tweet. It also scores minor modifications as mostly similar. Figure 3.36 is the results for the character-wise Levenshtein similarity measure. The measure reasonably scores the unchanged Tweet. It measures the observably dissimilar tweet as distant, but not as distant as might be expected. It also scores minor modifications as mostly similar.

Figure 3.37 is the results for the string token Jaccard similarity measure. The measure reasonably scores the unchanged Tweet and observably dissimilar tweet. It also scores minor modifications as mostly similar. Figure 3.38 is the results for the character-wise Jaccard similarity measure. The measure reasonably scores the unchanged Tweet. It does not score the observably dissimilar tweet as sufficiently dissimilar. It also scores minor modifications as mostly similar.

Figure 3.39 is the results for the string token Hamming similarity measure. Where the Tweets were of the same length, the Hamming distance measured minor modifications and the same Tweet reasonably. It failed for all Tweets that were not the same length. Failed tests are shown as missing data points in the figure. Figure 3.40 is the results for the character-wise Hamming similarity measure. Where the Tweets were of the same length, the Hamming distance measured minor modifications and the same Tweet reasonably. It failed for all Tweets did not contain the same number of string tokens.

Figure 3.41 is the results for the T-Information similarity measure. The measure reasonably scores the unchanged Tweet and the observably dissimilar tweet. It also scores minor modifications as mostly similar.

Some of the similarity measures are not appropriate for clustering Tweets due to failing the initial basic similarity test. Both the character-wise and string token Hamming distances failed all tests where the modified and original Tweets had an unequal number of characters or tokens, respectively. As many social media posts do not have the same number of characters, Hamming distance was discarded as an option for

further testing. The character-wise Jaccard distance effectively measured the common alphabet used between Tweets as a function of the length. As a result, it failed to adequately distinguish a unique Tweet and therefore was discarded from evaluation. Consequently, the remainder of the research focused on Jaccard string token, Levenshtein character-wise, Levenshtein string token, and T-Information distances.

3.5.5.7 Similarity Measure Complexity

Each of the similarity measures explored had different computational complexities. As a result, each measure was tested to determine its fitness for the clustering research. This was accomplished by applying each similarity measure against the same data set for increasing sample size and measuring the total time taken for the operation.

For this test, two experiments were carried out. One experiment was done for small samples ranging from 10 to 100 and the second experiment for larger samples ranging from 100 to 10 000. In a worst-case clustering operation each Tweet must be compared to every other Tweet in the set, so for this experiment, the distance between each pair of Tweets in the set was calculated. Both experiments were run in Docker container on Jupyter Notebooks, running Python3 on a virtual computer system comprising at 3.60GHz and with 14 GB RAM.

For the first test, the Vancouver search data set was sampled at 10, 50, and 100 Tweets, which corresponded to 100, 2500, and 10 000 distance calculations for each similarity measure. The results for the small sample size can be seen in Figure 3.42.

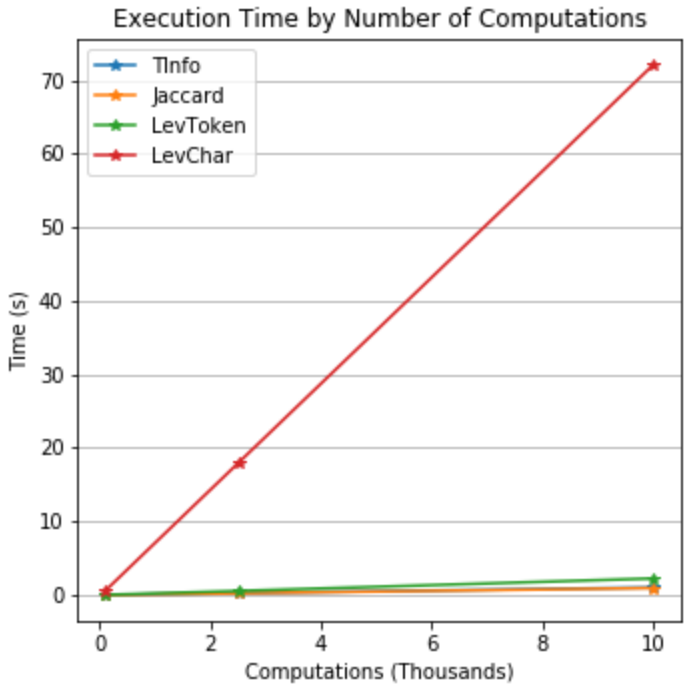

Figure 3.42: Practical Computational Complexity Small Sample

As can be seen in Figure 3.42, character-wise Levenshtein similarity edit distance was cost prohibitive for large sets and was consequently removed from the analysis pool for larger sample sizes.

For the larger sample size experiment, seen in Figure 3.43, the Vancouver search dataset was sampled at 100, 500, 1000, and 5000 Tweet which corresponds to 10K, 250K, 1M, and 25M distance calculations, respectively.

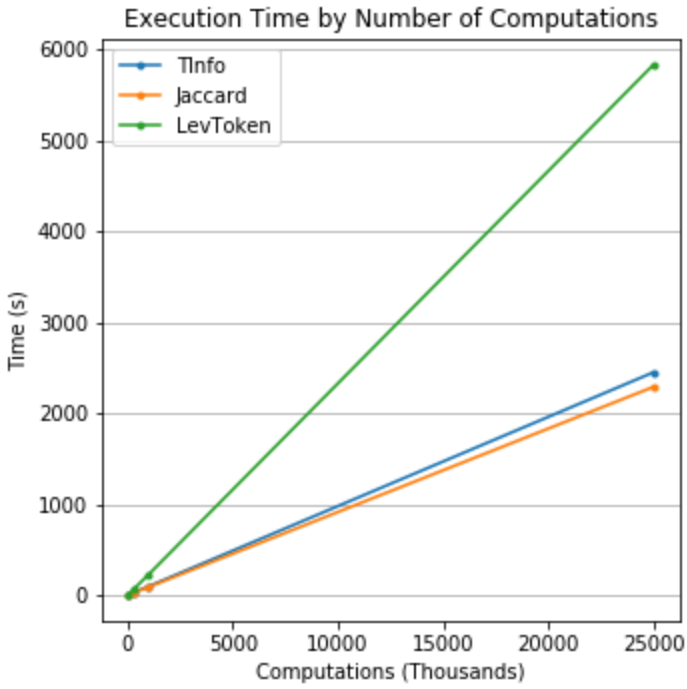

Figure 3.43: Practical Computational Complexity Large Sample

As can be seen, the practical implementation of Levenshtein Distance by Tokens is more computationally complex than T-Information and Jaccard. T-Information and Jaccard Similarity distances were found to perform similarly on this sample set.

3.5.5.8 Similarity Measure Selection

Each similarity measure was tested for robustness and computational cost. It was determined that Hamming distance and Jaccard character-wise distance were not robust enough to small variations and were excluded. Levenshtein character-wise distance was too computationally expensive and was also excluded. The measures chosen to evaluate for effective clustering were Levenshtein distance with string tokens, Jaccard distance with string tokens, and T-Information distance.

3.5.6 Cluster Modality Testing

A final data characterization was made to understand how effectively the data would cluster. Some clustering algorithms do not guarantee convergence [33], therefore, it was necessary to understand if there were enough similar Tweets in the datasets to generate clusters.

A sample of fifteen thousand Tweets was randomly selected from each data set. From that sample, fifty Tweets were randomly selected to be analyzed. Each of the fifty Tweets was compared to all other Tweets in the original sample set. The degree of similarity between each Tweet and all other Tweets was measured using the Levenshtein Distance, Jaccard Distance, and the T-information distance. Jaccard and T-Information examples are shown here to represent a character-based measure and a string token-based measure.

The similarity distances were tabulated and graphed as a histogram. Four representative examples of T-Information and Jaccard Similarity distance histograms using Vancouver and RoyalWedding can be seen in Figure 3.44, 3.45, 3.46, and 3.47. The corresponding sample Tweets are shown in Table 3.9, 3.10, 3.11, 3.12, respectively. The samples are not necessarily representative of the entire set but are chosen based on their cluster behaviours. A qualitative estimate of each Tweet's clusterability is included in each table and classified as Similar, Dissimilar, or Ambiguous as compared to other Tweets in the fifteen thousand Tweet sample. Tweets classified as 'Similar,' represent clusterable Tweets and have a number of Tweets in the dataset more similar than the rest of the sample. This is visible as isolated group of low distance values in the histogram. 'Dissimilar' Tweets are distant from others in the sample and may have a large group of distances measured greater than 0.8. Finally, 'Ambiguous' Tweets either show multiple characteristics or no definitive characteristics in the histograms.

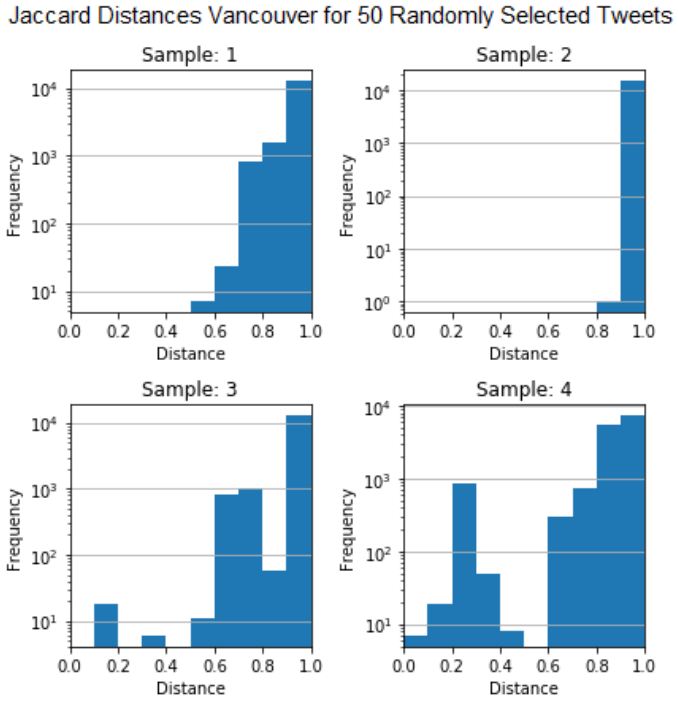

Figure 3.44: Representative Jaccard Distances for Vancouver Samples

Table 3.9: Sample Tweets and Content For Jaccard Vancouver

Sample	Content	Cluster Behaviour
1	We're #hiring! Click to apply: Engaging Brand Ambassador for a Premier Home Coffee System - Vancouver -… https://t.co/U8ZS7tQmC6	Ambiguous
2	LEZZZGOOO! #WhitecapsFC https://t.co/ajTp813TQ0	Dissimilar
3	Can you recommend anyone for this #job? Director, Human Resource Business Partner - https://t.co/chJBWENg2N #HR… https://t.co/v3nNeUX0Jx	Similar
4	Want to work in #Vancouver, BC? View our latest opening: https://t.co/LUB6UFe7ar #Database #Job #Jobs #Hiring #CareerArc	Similar

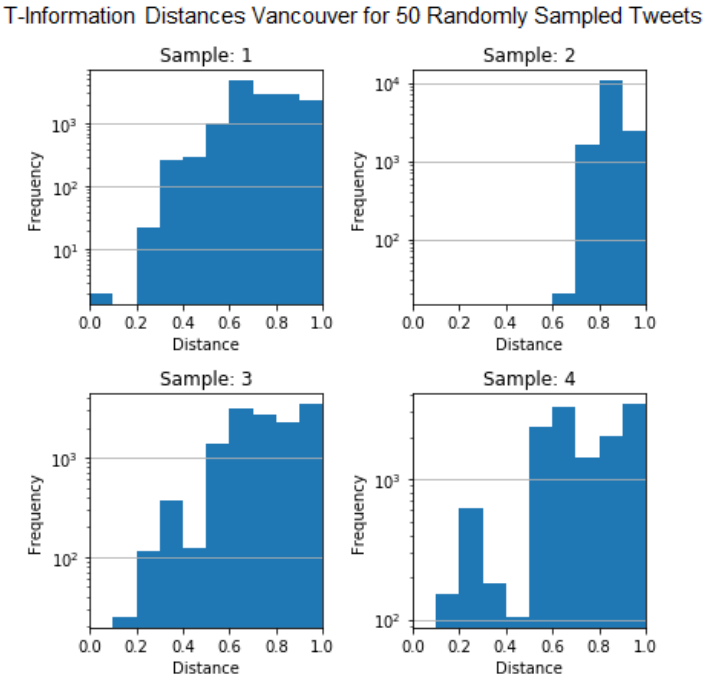

Figure 3.45: Representative T-Information Distances for Vancouver Samples

Table 3.10: TInfo Sample Tweets and Content For Vancouver

Sample	Content	Cluster Behaviour
1	We're #hiring! Click to apply: Web Content Specialist - https://t.co/17dUXLhXeZ #Writing #Vancouver, BC #Job #Jobs #CareerArc	Similar
2	Enjoying my first @CREWvancouver panel luncheon about the Multi-Family #Development Forecast! Meeting new faces AND… https://t.co/fgfzTgioGd	Dissimilar
3	Want to work at TD Bank Canada? We're #hiring in #Vancouver, BC! Click for details: https://t.co/Ypx7jpNsxo #Banking #Job #Jobs #CareerArc	Similar
4	Want to work in #Vancouver, BC? View our latest opening: https://t.co/gFBLCxoxHc #QA #Job #Jobs #Hiring #CareerArc	Similar

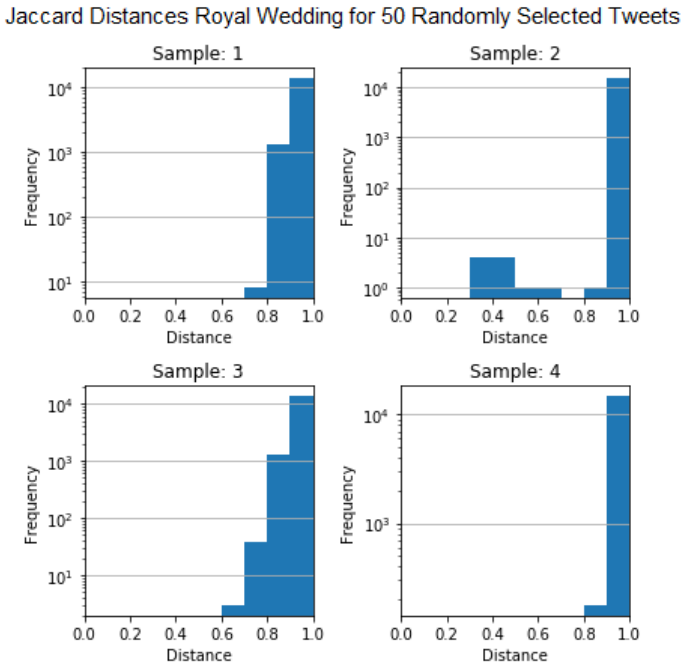

Figure 3.46: Represenative Jaccard Distances for RoyalWedding Samples

Table 3.11: Jaccard Sample Tweets and Content For RoyalWedding

Sample	Content	Cluster Behaviour
1	#RoyalWedding Beautiful horses! Very well seated.	Dissimilar
2	ابهاء #مسابقه_عبدالعزيز_الخضيري #امن_الدوله_شكرا_لكم #RoyalWedding ديكور وعازل للحرارة و البرودة والصوت ومقاوم لصت... https://t.co/5RjiJhpR29	Similar
3	I feel like I'm watching a #Suit episode!! #RoyalWedding	Dissimilar
4	#NAB does not believe in vindictive actions. #RoyalWedding #StreamFakeLoveNow #AssamNeedsProtection… https://t.co/4bF3auEp6E	Dissimilar

Sample 2 roughly translates to the following using Google Translate [60]:

Abha # مسابقه_عبدالعزيز_الخضيري #Insurance_India_ Thank you #RoyalWedding Decorative, heat insulation, cold, sound and resistor .. https://t.co/5RjiJhpR29.

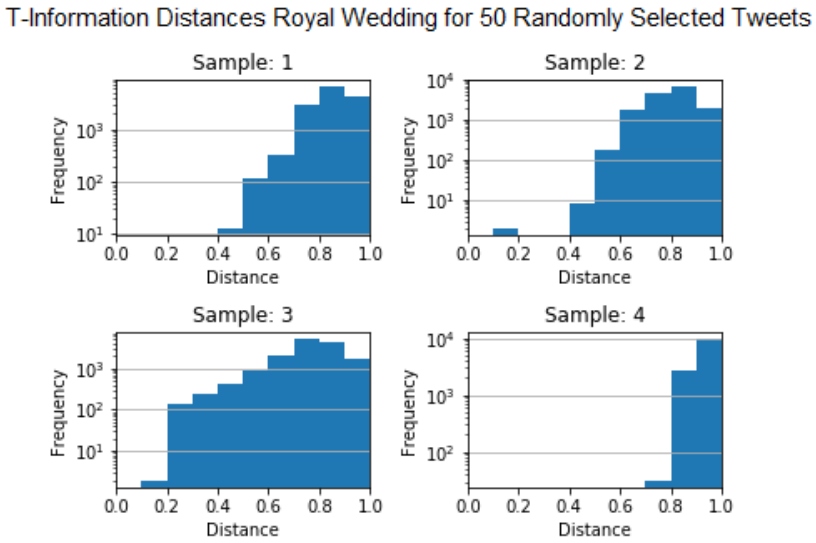

Figure 3.47: Representative T-Information Distances for RoyalWedding Samples

Table 3.12: T-Information Sample Tweets and Content For RoyalWedding

Sample	Content	Cluster Behaviour
1	The latest The Renee Everett Daily! https://t.co/rlun5VsyrW Thanks to @RichardddeNooy #seo #royalwedding	Ambiguous
2	Meghan Markle and Prince Harry shared a secret moment on the altar #RoyalWedding #HarryandMeghan https://t.co/vkQ1XfYSzN	Similar
3	#RoyalWedding https://t.co/3waerOWHuv	Ambiguous
4	Kate comunque sará pure in simil bianco ma vince sempre tutto nella vita, elegantissima stupenda è una Queen in ogni occasione #RoyalWedding	Dissimilar

As can be seen in the modality tests, each of the similarity measures could cluster the Twitter content. As a result, no further measures were excluded from the clustering research.

3.6 Data Clustering Methods

Two primary clustering methods were explored based on a standard thresholding algorithm as presented in I-TWEC [17]. Additionally, small modifications including the removal of popular hashtags and short tweets were made to each dataset to understand those factors would impact clustering.

3.6.1 Threshold Based Clustering

The basis of the threshold clustering algorithm was presented in the paper by I-TWEC [17]. The threshold clustering algorithm was tested using two methods, the I-TWEC version and a modified I-TWEC version. The algorithm operates on two primary variables; the threshold parameter determines the minimum distance a Tweet must be from another Tweet include in a cluster and the minimum cluster size parameter determines how many Tweets must comprise a cluster for the cluster to be valid. Both algorithm methods were tested across several practical threshold parameters and minimum cluster sizes.

3.6.1.1 I-TWEC Threshold Clustering Algorithm

The basic threshold clustering algorithm presented in I-TWEC [17] was tested for its applicability with the selected similarity measures. It allows for a simple method of clustering documents that has no maximum cluster size. It is dependent on a user defined threshold, *th*, and a minimum cluster size, *minC*. The theoretical computational complexity for this implementation is $O(N^2)$ in the edge case where no Tweets meet the threshold distance. The primary advantage of the ITWEC based algorithm is that it will attempt to re-cluster Tweets that do not originally find a cluster larger than the minimum cluster size. This should result in fewer unclustered tweets and slightly larger clusters. This added benefit comes at the expense of computational complexity. The pseudo-code for the I-TWEC based clustering algorithm can be seen below in Figure 3.48:

Algorithm 1 ITWEC Threshold

```
1:  Clusters = {}
2:  for i ← n do
3:      Tweet_i.isClustered ← False                ▷ All Tweets Are Unclustered
    end
4:  for i ← n do
5:      if Tweet_i.isClustered) = False then
6:          c={i}
7:          for j ← (i+1) to n do
8:              if Tweet_j.isClustered = False and
9:                  dist(Tweet_i, Tweet_j) ≤ threshold then
10:                 c ← c ∪ j
            end
11:         if |c| ≥ minC then
12:             C ← C ∪ c          ▷ If clustersize greater than minC add to set
13:             for index ∈ c do
14:                 Tweet_index.isClustered ← true          ▷ Update as Clustered
                end
    end
```

Figure 3.48: ITWEC Algorithm Pseudocode [17]

3.6.1.2 Modified Threshold Clustering Algorithm

A minor variation of the Threshold Based algorithm was also tested. Instead of checking for a max cluster size in the loop, clusters were evaluated for max size after the analysis phase. This change in operations reduces the practical computational complexity, the worst-case complexity for the Modified threshold is determined Equation 3.4 where N is the sample size and "!" is the factorial operator. The trade-off, however, is potentially missing clusterable Tweets as the algorithm then disregards Tweets that don't meet the initial threshold and does not revisit them for later clustering. The practical implications for this modification were tested. The pseudo-code for the adjusted algorithm is seen in Figure 3.49.

$$Complexity_M = \frac{N!}{2!(N-2)!} = \frac{N(N-1)}{2} \qquad \text{(Eq. 3.1)}$$

Algorithm 2 Modified Threshold

```
    Clusters = {}
2:  for i ← n do
        Tweet_i.isClustered ← False            ▷ All Tweets Are Unclustered
    end
4:  for i ← n do
        if Tweet_i.isClustered = False then
6:          c={i}
            Tweet_i.isClustered ← True          ▷ Mark Tweet i as Clustered
8:          for j ← (i+1) to n do
                if Tweet_j.isClustered = False and
10:                 dist(Tweet_i, Tweet_j) ≤ threshold then
                    c ← c ∪ j
            end
12:         C ← C ∪ c
    end
    for c ∈ C do                                ▷ Evaluate Clusters
14:     if |c| < minC then
        Tweets_c.isClustered ← False    ▷ Update small clusters as Unclustered
    end
```

Figure 3.49: Modified Threshold Algorithm Pseudocode

3.7 Analysis Metrics

To evaluate the effectiveness of the clustering algorithms, a process and evaluation measures were defined, as discussed below.

3.7.1 Clustering Computational Complexity

Both clustering algorithms clustering algorithms have a worst case computational complexity of $O(N^2)$. Focus was instead placed on the practical computational complexity of the algorithm. The number of computations required to run a complete clustering algorithm was used as a measure for the practical computational complexity. For each test, the number of computations was recorded and graphed against the worst-case maximum.

3.7.2 Unclustered Posts and Data Reduction

Unclustered posts represent unique posts in the dataset and, therefore, are of interest. The number unclustered posts also help determine how effective each algorithm was at clustering content. For both threshold algorithms the number of unclustered posts are highly dependent on the minimum cluster size and the similarity threshold selected for each test.

Ultimately, the purpose of this research is to reduce the number of posts that an analyst, marketer, or journalist would have to review and analyze. Consequently, how data could be flagged as redundant and filtered is a key evaluation metric. The overall data reduction was calculated by comparing the number of posts in the original sample dataset to the total number of clusters and unclustered posts. The underlying assumption being each cluster of similar Tweets represents a common topic or redundant information. Reduction represent the amount of content remaining after a theoretical noise filtering operation as a fraction of the starting content. The equation for data reduction given in Equation 3.5 below. In Equation 3.5, U represents the set of

unclustered posts, C represents the set of clusters greater than the minimum cluster size, O represents the set Tweets from the original experiment sample, and |.| is the cardinality operator.

$$\text{Reduction} = \frac{|U|+|C|}{|O|} \qquad \text{(Eq. 3.5)}$$

3.7.3 Total Clusters and Cluster Size

The total number of clusters and cluster size are an important metric for understanding what each search represents and its compressibility. A search that contains a large total number of clusters but has a low average clusters size indicates there are many topics and lots of information contained in the search. Searches with a lot of unique information are not compressible, in that, it is hard to reduce the information in the set to a smaller number of Tweets. A search that contains large clusters, but has relatively few total clusters indicated the search is dominated by a single common theme, hence, it is highly compressible. For the threshold algorithms, the total clusters were calculated as the number of produced clusters that equaled or exceeded the minimum cluster size. Cluster size was calculated by counting the number of Tweets contained in each cluster.

3.7.4 Cluster Root Mean Squared Distance

Another measure useful to understand is the density of each cluster. Very dense clusters indicate that each Tweet in the cluster comprises the same topic same language. Filtering out content from contained in a dense cluster can be safely assumed to not remove valuable information from the set. Low density clusters indicate a loose similarity relationship between the Tweets in the set. Cluster density was calculated using a Root Mean Square Distance (RMSD) approach as given in Equation 3.6. To get an indication of the quality of the clusters from a given experiment the average RMSD across all clusters was calculated.

$$RMSD_{cluster} = \sqrt{\frac{\sum_{i=0}^{n} dist(Tweet_i, Tweet_j)^2}{n}} \quad \text{(Eq. 3.6)}$$

3.7.5 Cluster Validation

To ensure the clustering algorithm and similarity measures were behaving as intended a select set of manual inspections were carried out. The first quantitative cluster validation technique used was measuring the exemplar Tweet distances of the largest clusters. An exemplar Tweet for a cluster was defined as the Tweet with the minimum root mean square distance to all other Tweets in the cluster, roughly equivalent to a centroid Tweet. This measure could be used to determine if the top clusters were close together or far apart in the given measure space. The second test comprised measuring the similarity distance between the exemplar Tweets of the largest clusters and five randomly selected unclustered Tweets. This test gave an indication if the unclustered Tweets were discernibly different than the top clusters. Finally, the largest clusters were aggregated into a single string and the similarity distance between each of the cluster aggregates was measured. The further apart the aggregate cluster were indicates how similar each element in a cluster is to another cluster.

To qualitatively validate different clusters word clouds were generated using strings across the largest clusters in the set. Strings were built by appending all Tweets occurring within its respective cluster similar to the aggregate measure introduced previously. The word cloud algorithm removed ‘https’ in addition to common stopwords before generating its word cloud.

3.8 Clustering With Industry-Based Constraints

An experiment was developed to evaluate social media data clustering for industry relevant applications. There exist a few industry constraints that determine the necessary computational complexity and performance requirements for effective

clustering. The industry constraints evaluated included 500 post limits and real-time streaming applications.

3.8.1 Appropriate Threshold Values

Thresholds are applied to each algorithm to determine how similar one Tweet must be to another Tweet in order to cluster both Tweets. A threshold value of 0 clusters only exactly similar Tweets, while a threshold of 1 would place the entire data set into a single cluster. Overly strict similarity thresholds cause clusters to be too tight where small variations in Tweets are classified as dissimilar. Overly generous thresholds cause clusters to include Tweets that are not similar or cause neighbouring clusters to merge. To determine which thresholds are appropriate, a sample of 5,000 Tweets were randomly selected from each search and clustered for varying thresholds. 5000 Tweets was selected as a sufficient sample size to ensure there were enough clusters to observe the intra-cluster pair-wise behaviour. The intra-cluster pair-wise distances of the Tweets were then measured and plotted using a histogram to determine if there was an ideal range of thresholds. Initial threshold values were set from 0.4 to 0.8, at 0.1 increments based on the normalized threshold values used in I-TWEC using the string token Jaccard similarity distance, the string token Levenshtein similarity distance, and the T-Information similarity distance [17]. Using thresholds may introduce bias into the results as it is not possible to normalize the distances across all similarity measures resulting in each threshold representing a different value that may be more permissive for one measure than another. As the behaviour of larger sets were primarily of interest for this experiment, a minimum cluster size of 10 was used and the algorithm implemented was the modified thresholding algorithm presented in Section 3.6.1.2.

3.8.1 Sample Size

The number of Tweets compared by clustering algorithm has direct implications on quality of the resulting clusters and the computational complexity. An experiment was developed to evaluate clustering effectiveness and quality for varying sample sizes. For this experiment, the minimum cluster size was set to 2 and the similarity distance

threshold was set to 0.4. Ten independent runs were conducted on each of Vancouver, London, RoyalWedding and Worldcup datasets. For each run, the number of tweets in the sample was incremented from 250 Tweets to 1000 Tweets, in 250 Tweet increments. These sample sizes were selected to represent the number of Tweets expected from a standard industry Twitter search [21]. For every run, the clustering algorithms and similarity measures were compared using the same sample of Tweets.

3.8.2 Minimum Cluster Size

The minimum cluster size determines how many Tweets must be grouped together before it is classified as a cluster. This has a few practical consequences and complexity considerations in the implemented algorithms. Practically speaking, the minimum cluster size allows end users to set what amount of noise is tolerated for a given search. A minimum cluster size of 2 allows for the pair-wise grouping and filtering of Tweets but is expected to have the highest computational expense. A larger minimum cluster size will still group redundant and similar Tweets but may not produce all possible clusters. To measure the impact of cluster size on performance, a sample of 2000 tweets was taken from each search and clustered using a threshold of 0.4 and modifying the cluster size from 2, 5, and 10 to represent pairwise, small, and moderate sized clusters. The sample size of 2000 was selected to help ensure there would be enough data to effectively cluster for larger minimum cluster sizes.

3.8.3 500-Tweet Search Clustering

Twitter’s Enterprise Search API only serves content in 500 post sequences [21]. As a result, the clustering techniques were evaluated on simulated 500 Tweet searches. 500 post searches were simulated by randomly sampling the original searches blocks of 500 Tweets. The clustering criteria including thresholding and minimum cluster size were set after completing the previous experiments designed to understand and select appropriate values. ITWEC and the modified thresholding algorithm perform similarly for a minimum cluster size of 2, therefore, minimum clusters size is set to 3 in order to

observe performance differences between the algorithms. The threshold was set to 0.4 to help ensure tight, accurate clusters.

3.8.4 Real-Time Streaming Simulated Clustering

Twitter's Enterprise Streaming API allows for higher throughput than the Search API. In industry, as shown by the Echosec platform [2], it is common for a real-time streaming search to begin with a simple historical search then aggregate the new streaming content as it appears. This behaviour was simulated by starting with a seed data set of 500 Tweets from the original search generated by randomly selecting a 500 Tweet sequence of content. A streaming data set was then generated by a sequence of 2000 Tweets immediately following the seed set. The seed data set was clustered, in full, using both thresholding algorithms. The streaming data set was then clustered against the seed data clusters one at a time in chronological order to simulate new data entering the system. This method of clustering was termed "quick clustering." The quick clustering method was compared to each standard clustering method using the union of the seed and stream datasets to understand the trade-offs between methods. A percent difference calculation was used to compare the results from each simulation. The percent difference calculation is shown in Equation 3.7, where A represents the quick cluster mean measure and B is the full cluster mean measure. Real-Time Streaming constraints were also compared to real-world data throughput numbers analyzed in Section 3.5.4.

$$Percent\ Difference(A,B) = \frac{|A-B|}{B} * \%100 \qquad \text{(Eq. 3.7)}$$

3.9 Chapter Summary

Chapter 3 presents the methodology for this thesis. Social media data was acquired through the Echosec platform. The data was then sanitized of personal or private information and stored in a MYSQL database. The data was ingested and manipulated in Jupyter notebooks that made use of Pandas and various open source Python libraries. The data underwent a characterization phase to best understand what each search comprised. Similarity measures were then tested for robustness against common social media characteristics. Eventually, T-Information, Jaccard string token, and Levenshtein string token similarity measures were determined to be appropriate. The modified and ITWEC threshold based clustering algorithms to be evaluated were presented. Then a set of analysis metrics were determined for the evaluation of the similarity measures and algorithms including cluster quality, computational complexity, reduction characteristics, and a qualitative validation process. Finally, several industry relevant constraints were introduced to ensure relevance.

Chapter 4

4 Results

This chapter presents the results of the methodologies presented in Chapter 3. The effects of various parameters used in the thresholding algorithms are presented. The results of the industry-relevant simulations are also presented.

4.1 Similarity Distance Thresholding

The effects of similarity distance thresholding were explored using the methodology detailed in Section 3.8.1. The histogram plots in Figures 4.1 - 4.10 show the intra-cluster pairwise distances between each Tweet in the sampled search. Histograms were plotted for clusters using the T-Information, Jaccard and Levenshtein similarity measures. For each histogram, pairwise distances were measured from 0 to 1 and collected in 50 bins of width 0.2. Each colour of the superimposed histograms represents a unique cluster. As seen in the figures, the threshold parameter clearly affects how the clusters are formed for each data set. It is also apparent the clustering behaviour for various thresholds is also dataset dependant.

4.1.1 T-Information Thresholding Performance

Figures 4.1- 4.5 are histograms of the pairwise distances for each search using T-Information similarity distance. As can be seen, T-Information similarity distance results in continuous clusters as a direct result of its character-wise similarity measurement.

As shown in Figure 4.1 and Figure 4.2, T-information forms tight, continuous clusters for the thresholds 0.4 and 0.5, respectively. Many of the clusters have a smaller

pair-wise distance than the original threshold distance which suggests that T-Information can find better clusters than the maximum allowable distance. As threshold distances increase to larger values, however, different behaviour emerges including dominant clusters and cluster merging.

The first behaviour evident in all datasets for large threshold values is a small number of clusters start to dominate the dataset. As a result, many Tweets are classified as the same cluster, even if they may not be similar in nature. While this behaviour is dependent on the original data set, it can be seen to emerge for all sampled searches at threshold values greater than 0.7 indicated by the occurrence of multi-modal histograms.

The second behaviour observed for large threshold values is multiple clusters merging into one cluster. This pattern occurs when the mean distance of two clusters is approximately the same as the threshold parameter. As a result, the cluster has two distinct peaks in the pairwise histogram. An example of this behaviour can be seen in the large blue cluster with two peaks in WorldCup for thresholds 0.7 and 0.8 from Figure 4.4 and Figure 4.5. As these peaks are likely to represent different information, it would be beneficial to have these clustered separately and have a narrower pair-wise distance distribution.

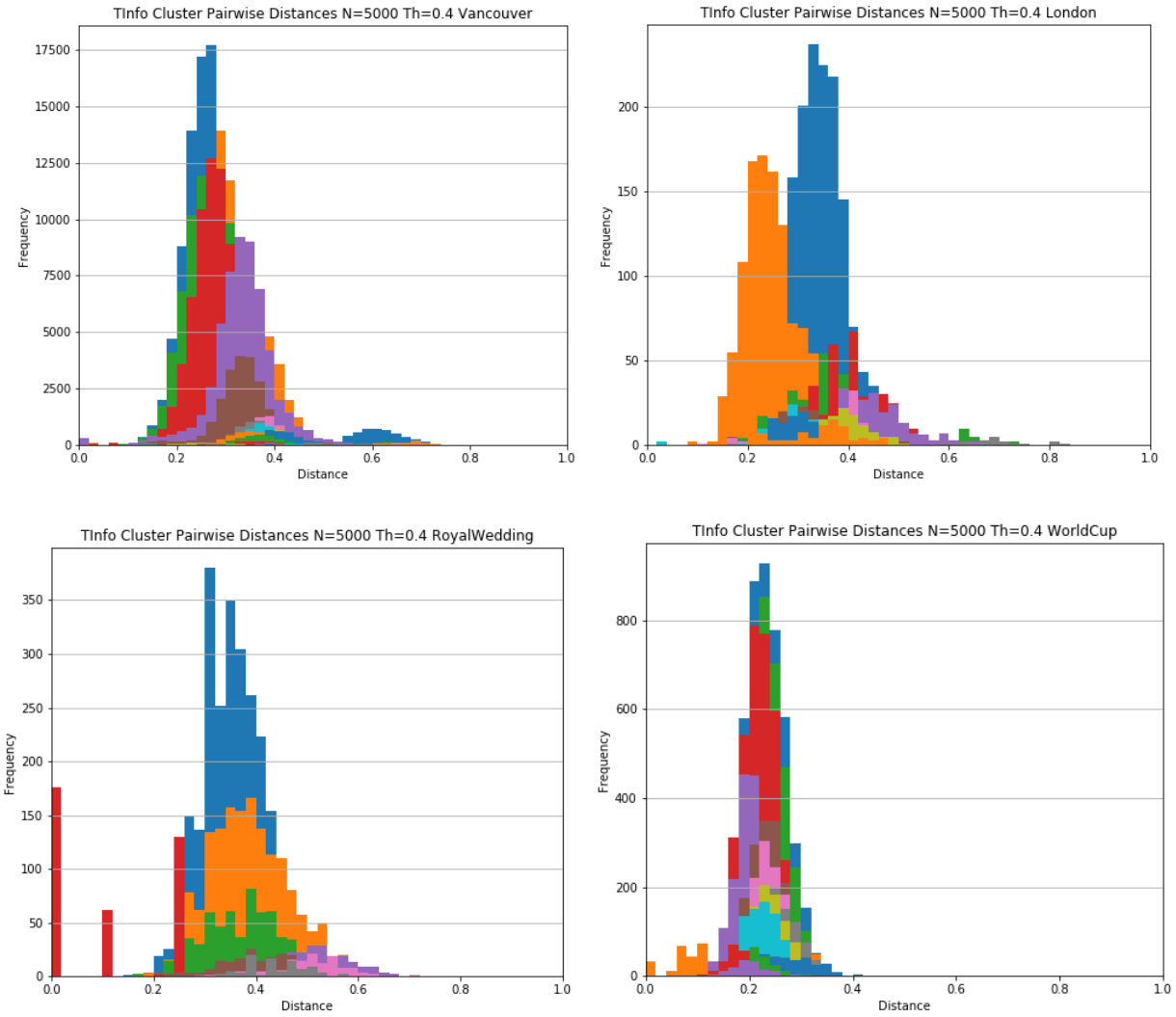

Figure 4.1: T-Information Threshold 0.4 For For All Searches

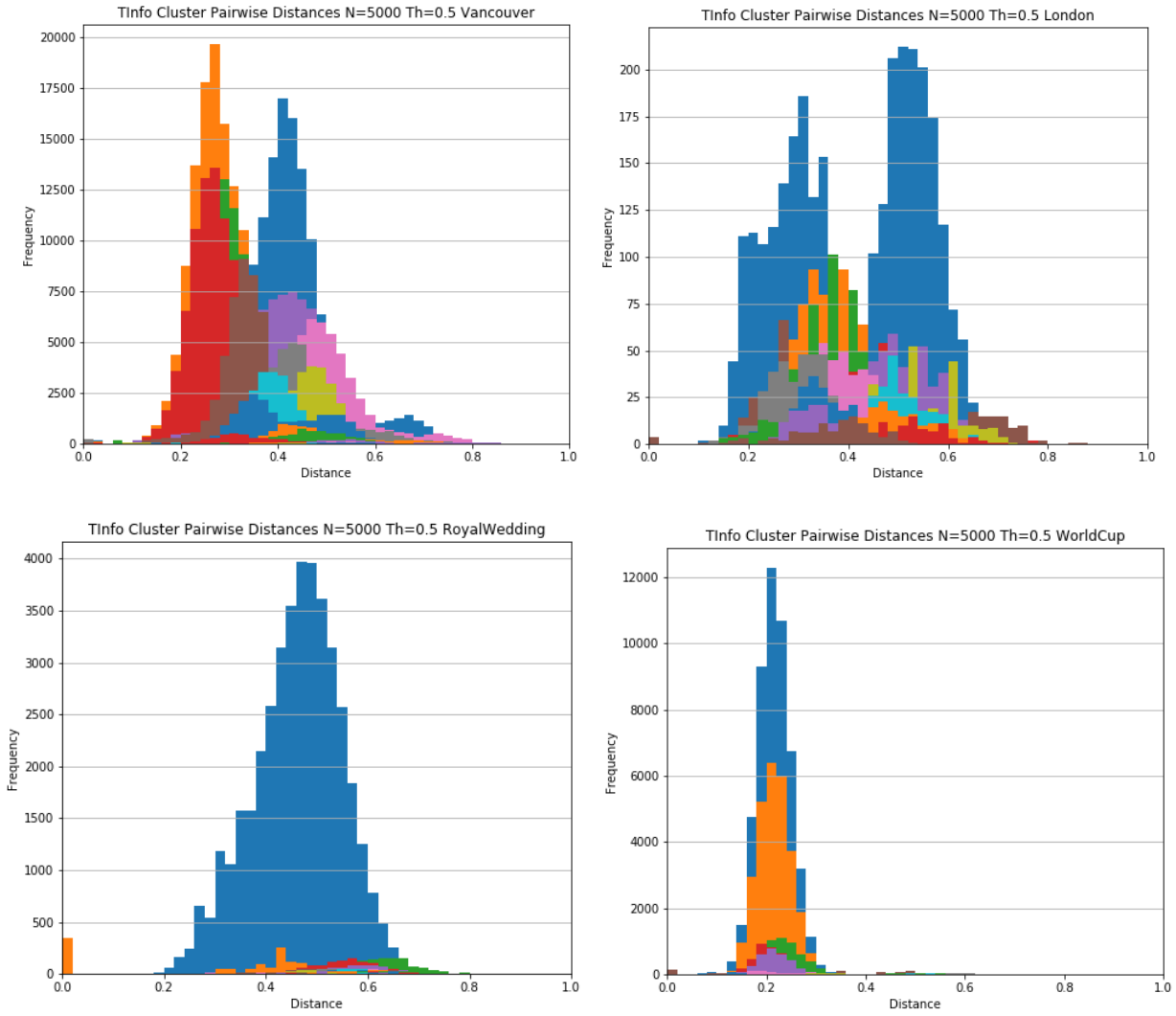

Figure 4.2: T-Information Threshold 0.5 For For All Searches

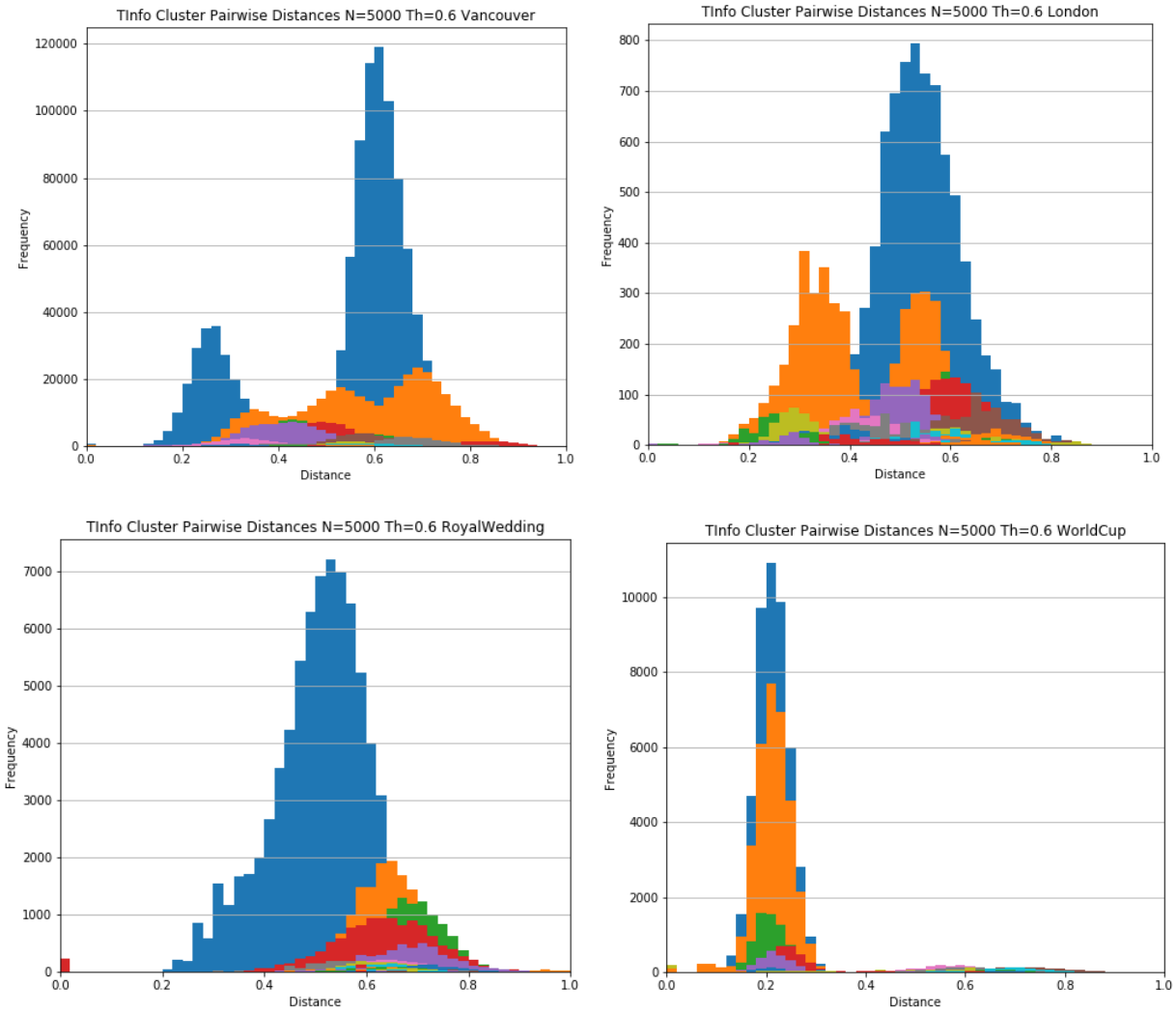

Figure 4.3: T-Information Threshold 0.6 For For All Searches

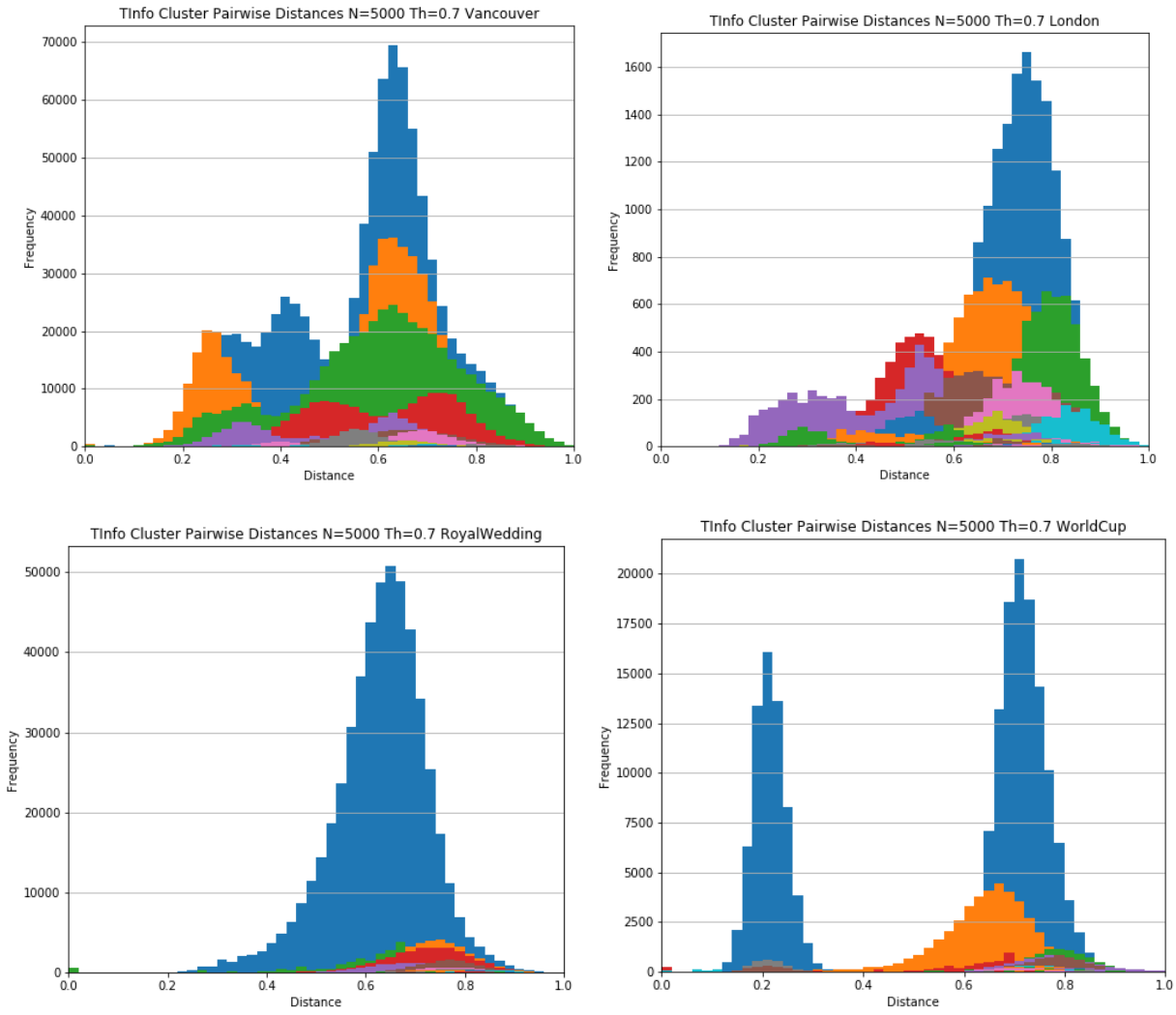

Figure 4.4: T-Information Threshold 0.7 For For All Searches

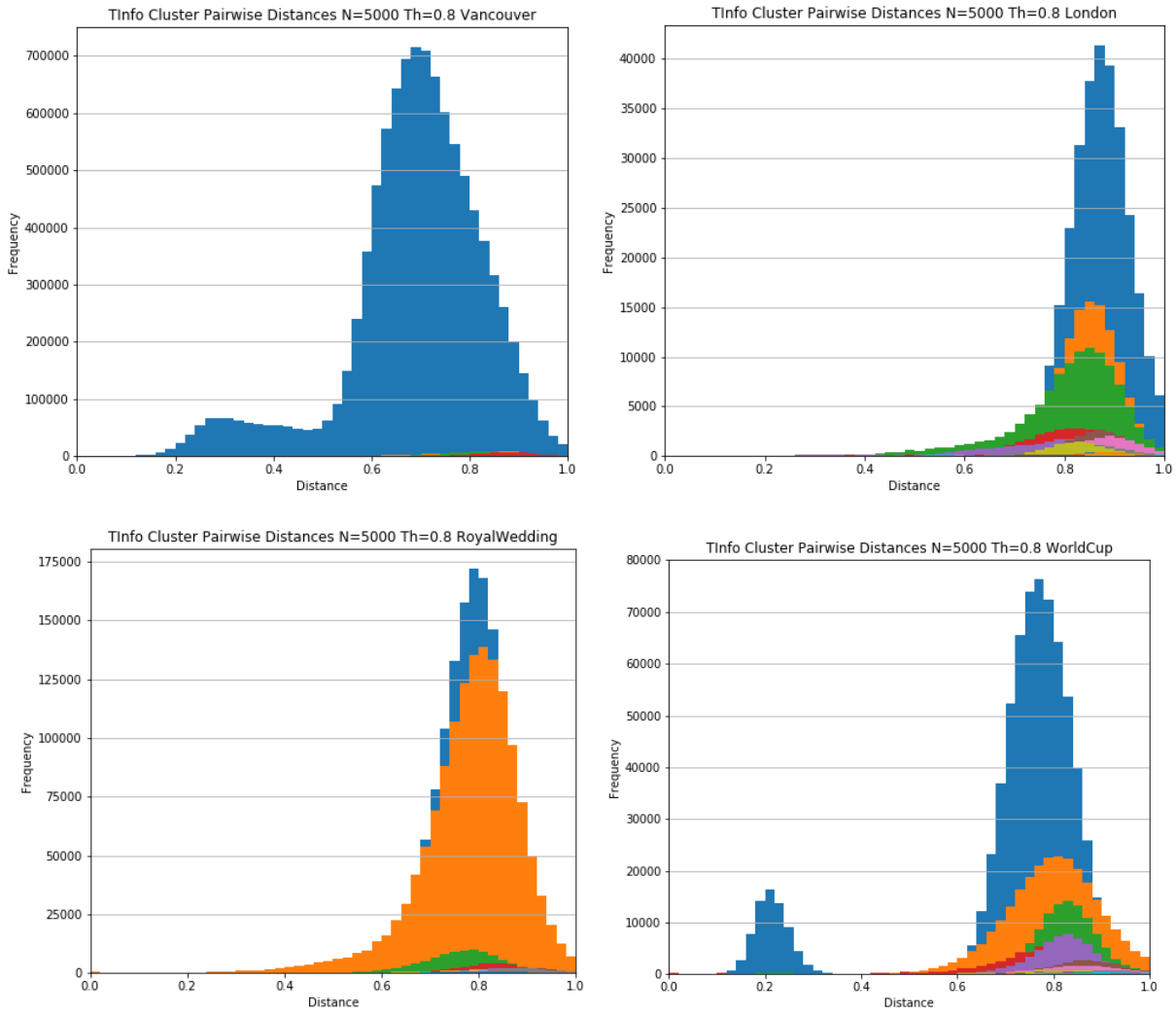

Figure 4.5: T-Information Threshold 0.8 For For All Searches

4.1.2 Jaccard Thresholding Performance

Figures 4.6 - 4.10 are histograms of the pairwise Tweet distances for each search as measured by the Jaccard similarity distance. As can be seen in the figures, Jaccard forms very narrow bands for intra-cluster pairwise distances. This is due to the Jaccard algorithm operating on string tokens, which are then also limited by the number of characters a Tweet may contain. As a result, there exist finite number of distances that can occur between two Tweets and common social media alterations cause the similarity distance to fit in a narrow very band.

As can be seen in the Figures 4.6-4.8, for the thresholds 0.4 to 0.6 many of the clusters have a smaller pair-wise distance than the original threshold distance and therefore can effectively identify similar clusters without grouping too many dissimilar tweets. As threshold size increases, clusters begin to merge and split across two peaks. The London search in Figure 4.10, for example, shows two large dominant clusters. Also in Figure 4.10, the blue cluster has two groups of peaks that indicate two or more groups of tweets are within the threshold parameter where it may have been productive to classify them independently.

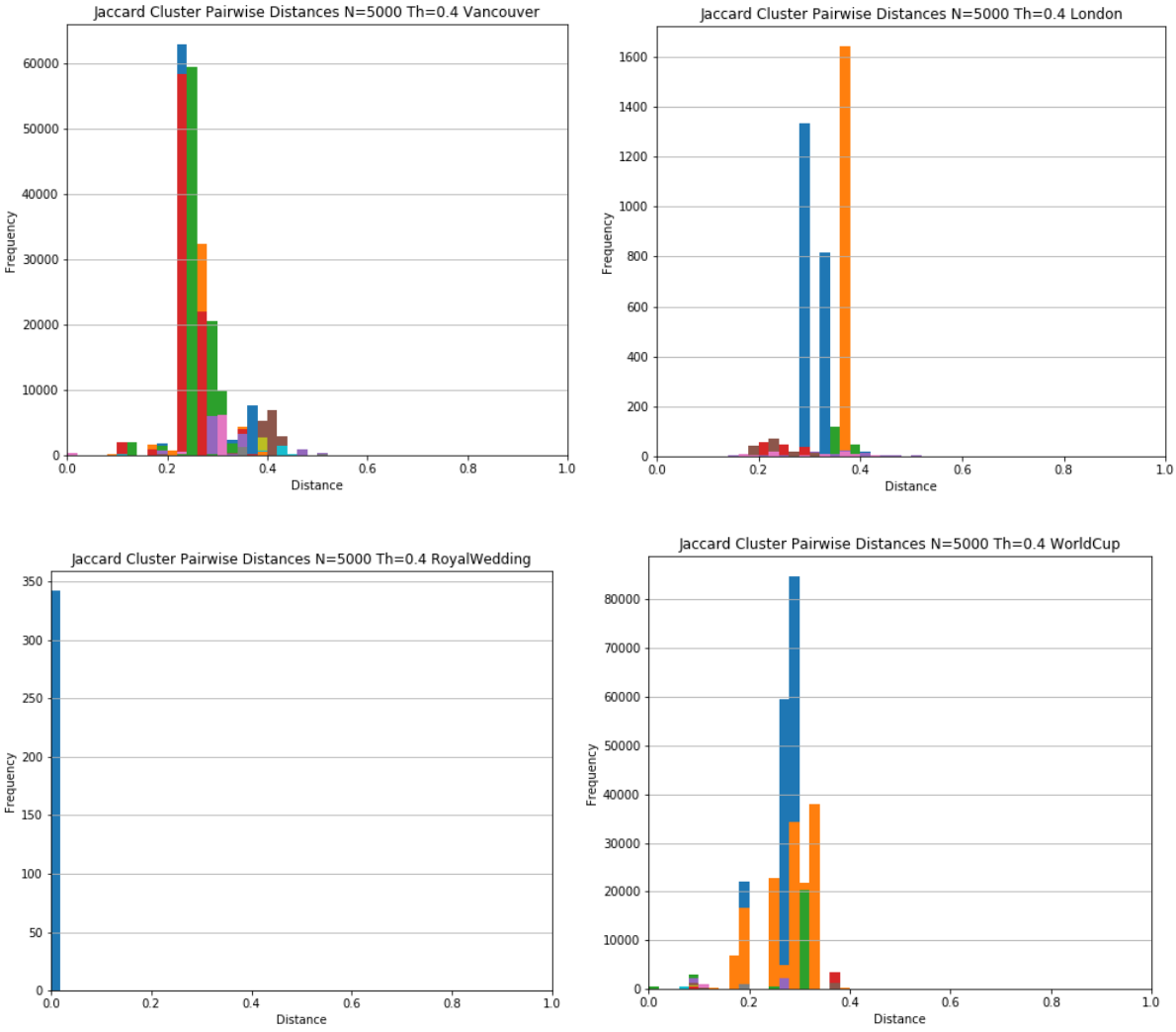

Figure 4.6: Jaccard Threshold 0.4 For For All Searches

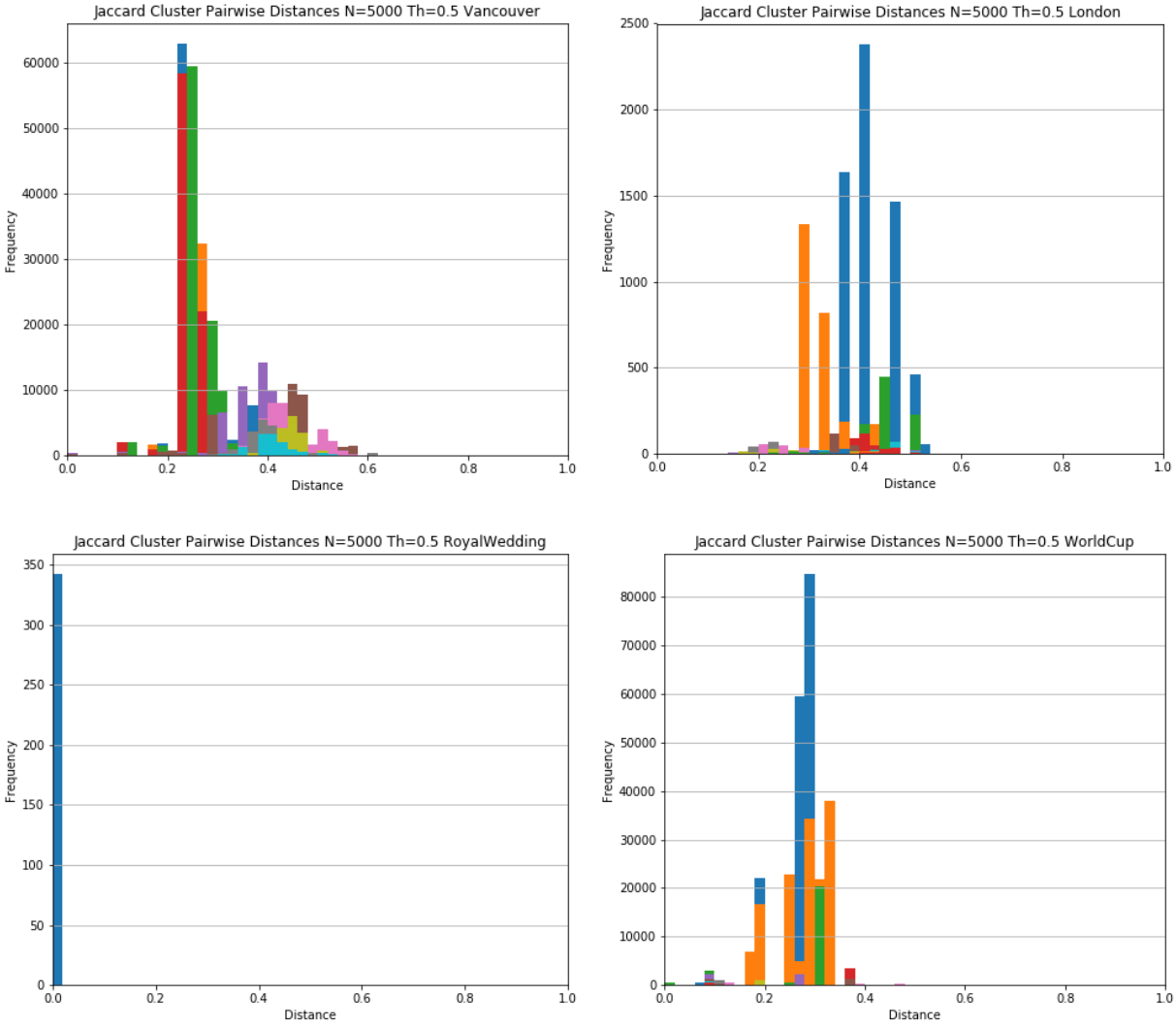

Figure 4.7: Jaccard Threshold 0.5 For For All Searches

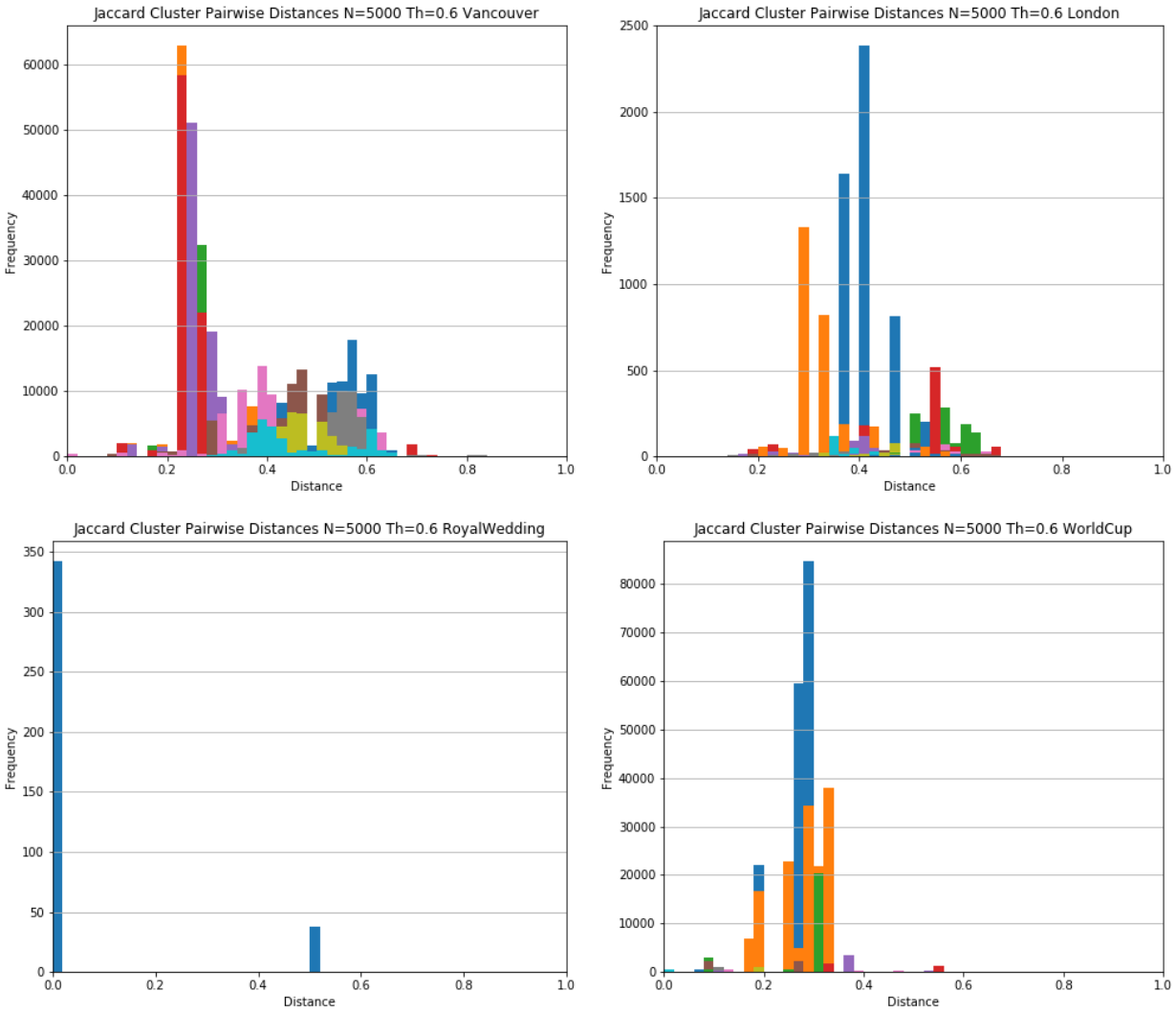

Figure 4.8: Jaccard Threshold 0.6 For For All Searches

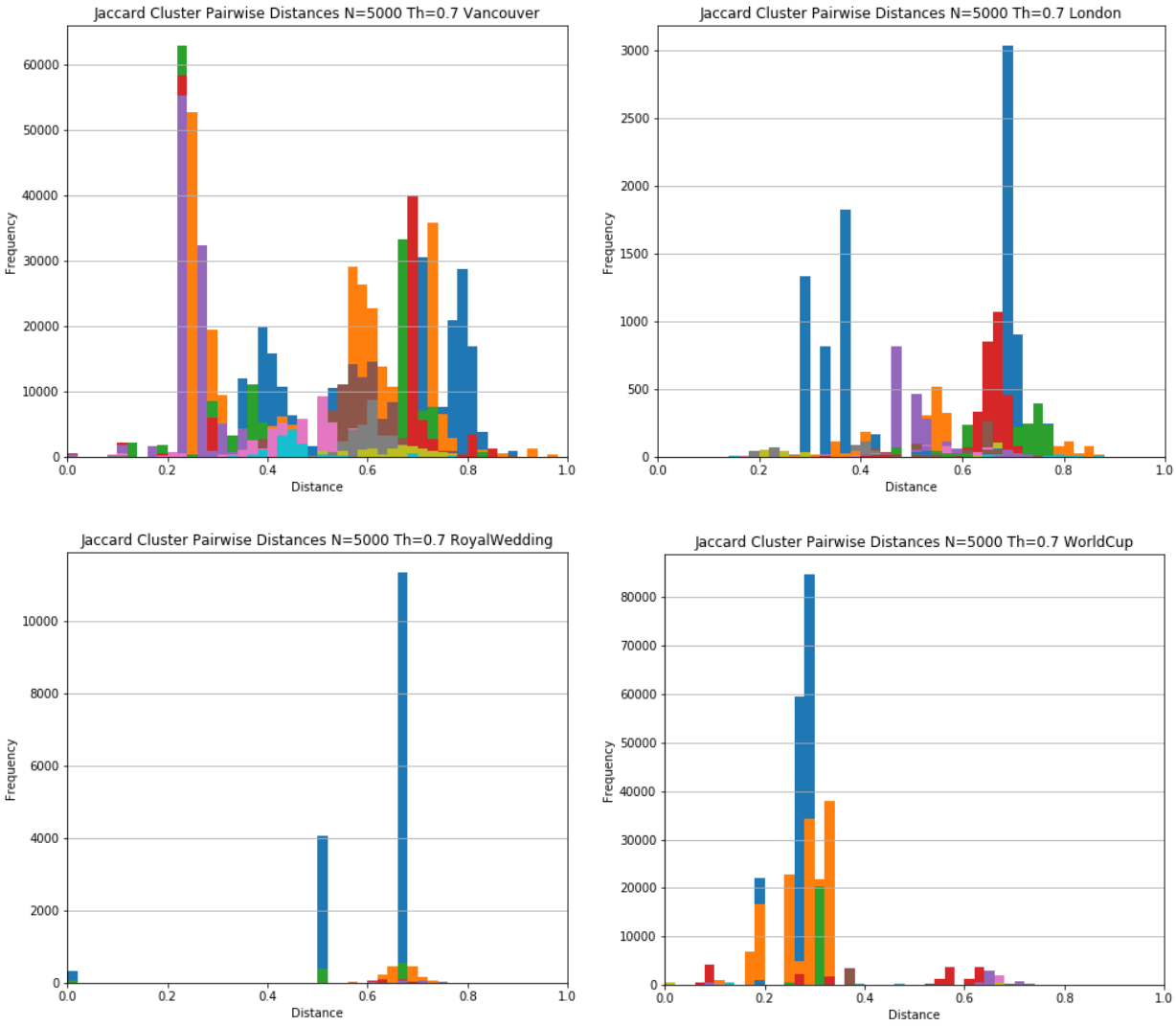

Figure 4.9: : Jaccard Threshold 0.7 For For All Searches

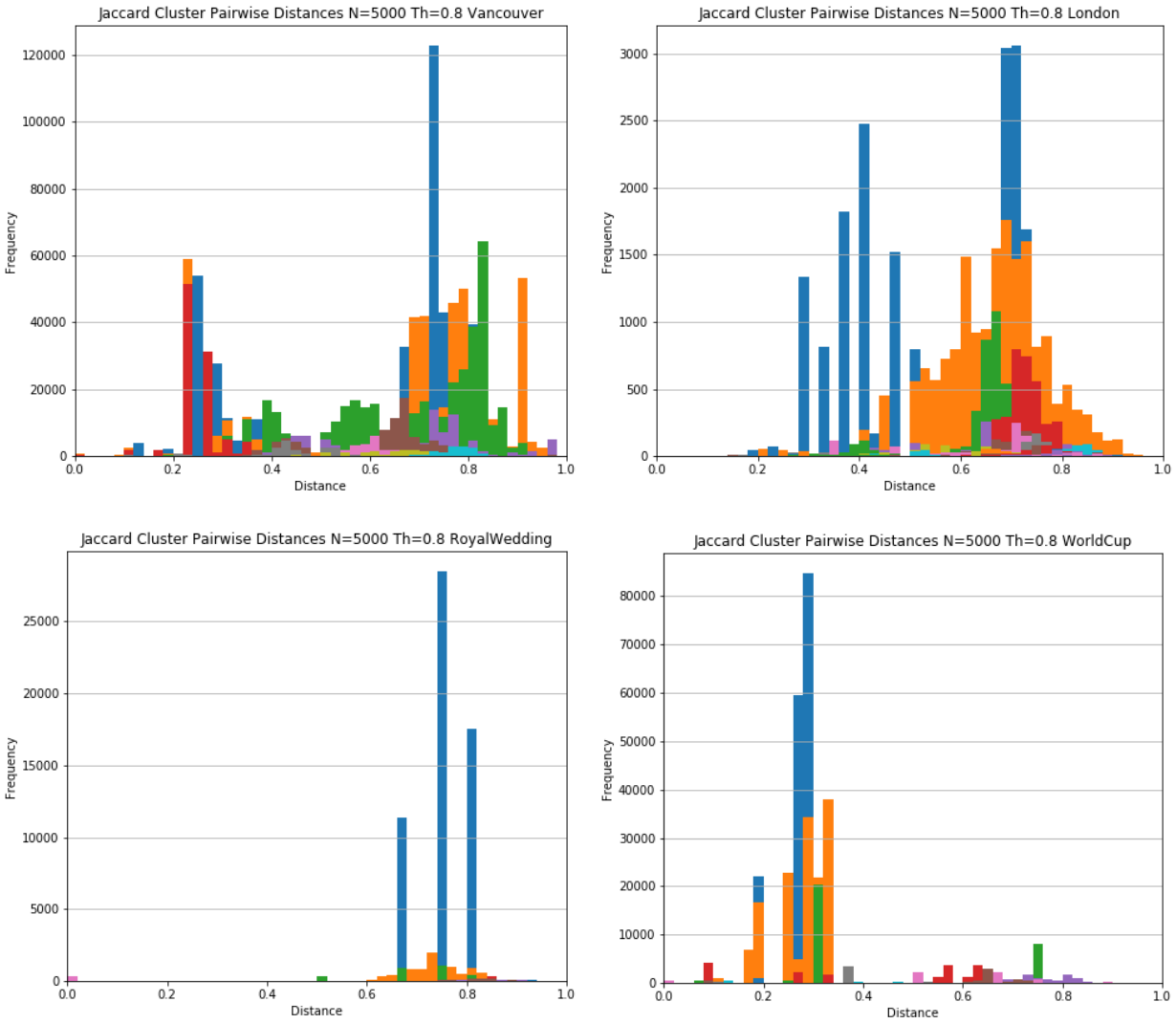

Figure 4.10: Jaccard Threshold 0.8 For For All Searches

4.1.3 Levenshtein Thresholding Performance

Figures 4.11 - 4.15 are histograms of the pairwise Tweet distances for each search as measured by the Levenshtein string token similarity distance. As similar to the Jaccard token-based measure, Levenshtein forms very narrow bands for intra-cluster pairwise distances.

As can be seen in Figure 4.11 and Figure 4.12, for the thresholds 0.4 and 0.5 many of the clusters have a smaller pair-wise distance than the original threshold distance and therefore can effectively identify similar clusters without grouping too many dissimilar tweets. As threshold size increases, clusters begin to merge and split across multiple peaks. Royal Wedding in Figure 4.13, for example, has a single cluster that has local peaks at 0, 0.5 and 1, which are highly varied and indicative that the clustering threshold is too lenient. As the threshold increases to 0.7 and 0.8, distinct clusters at any value are no longer visible or have multiple peaks.

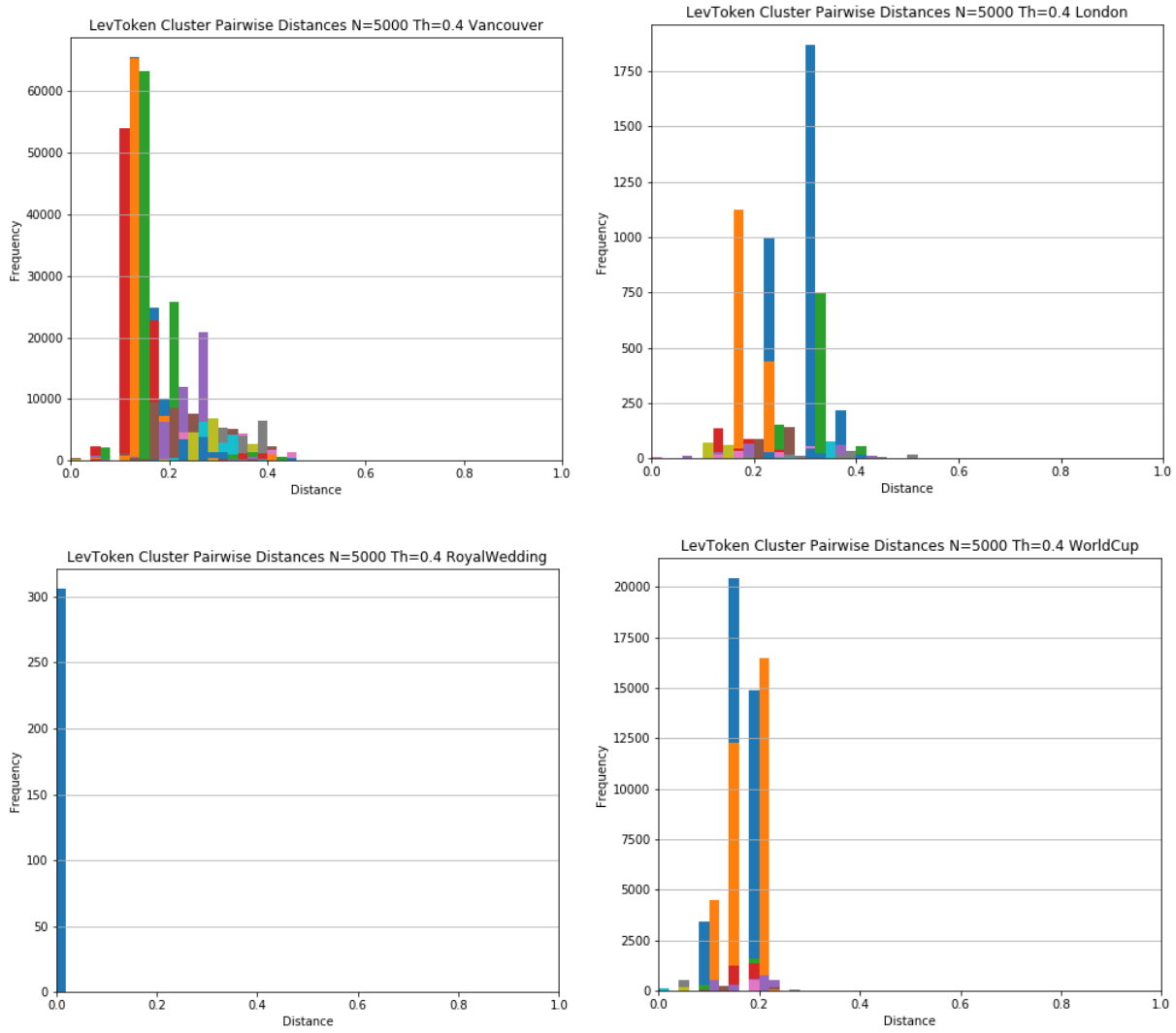

Figure 4.11: Levenshtein Threshold 0.4 for All Searches

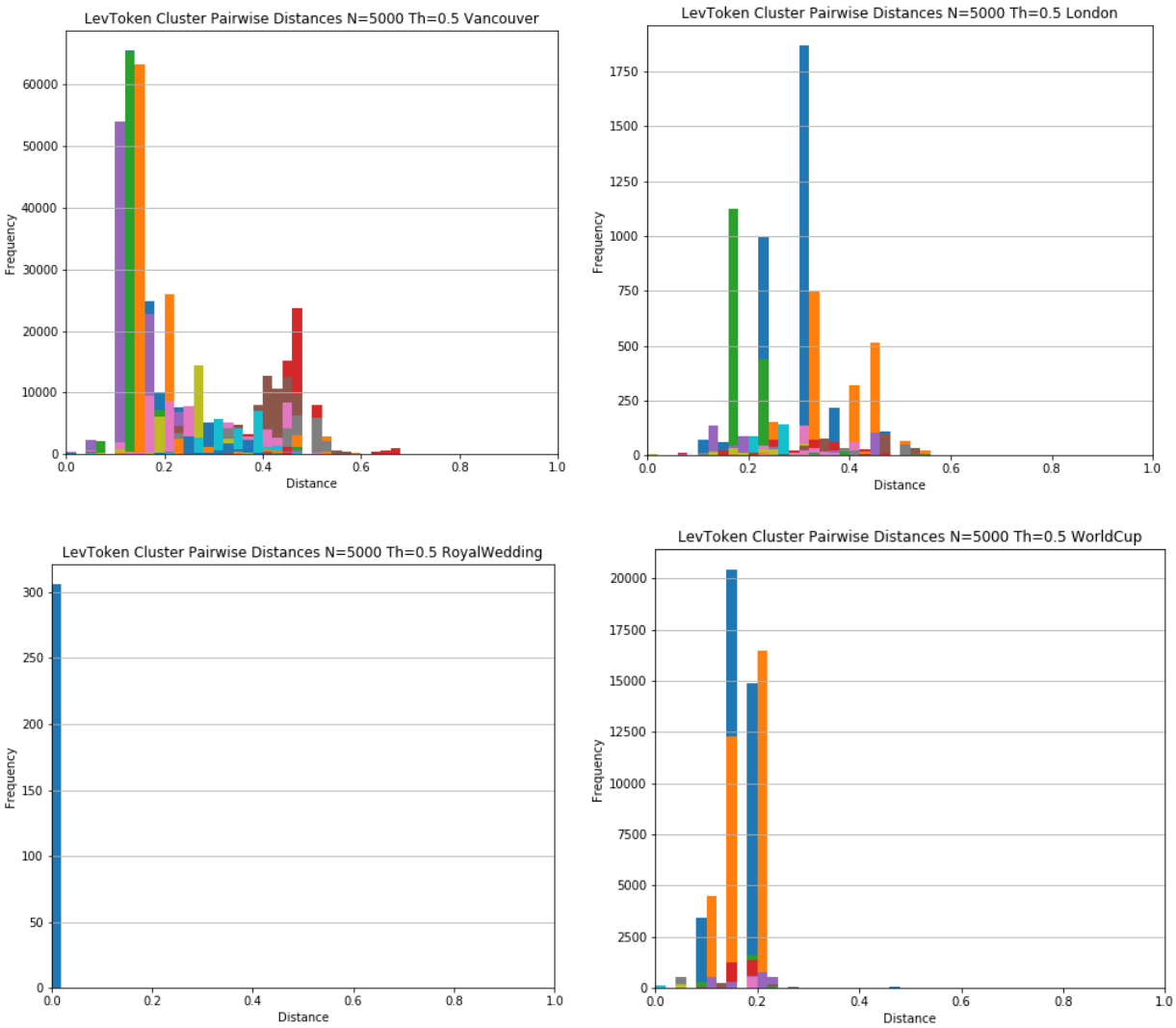

Figure 4.12: Levenshtein Threshold 0.5 for All Searches

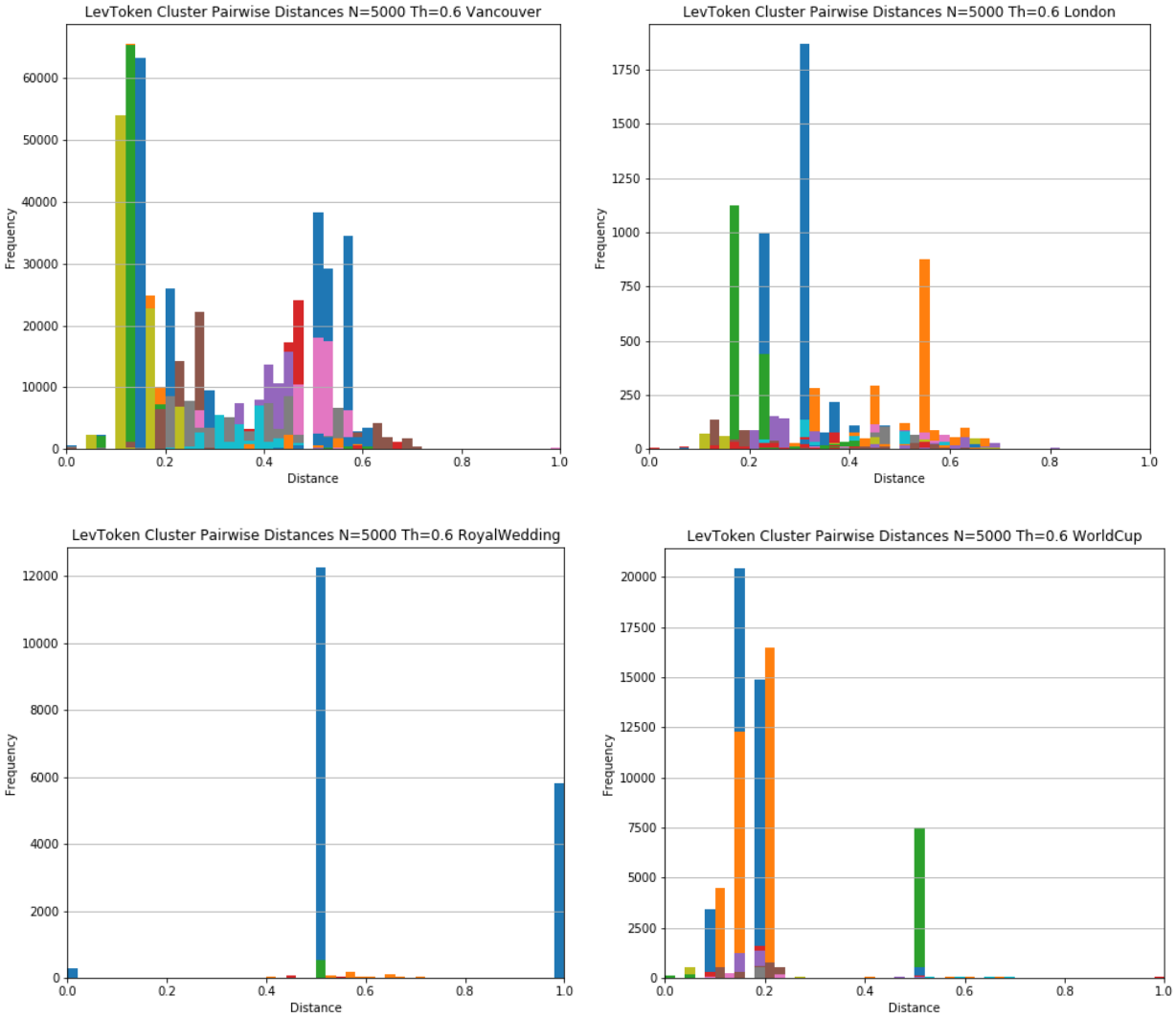

Figure 4.13: Levenshtein Threshold 0.6 for All Searches

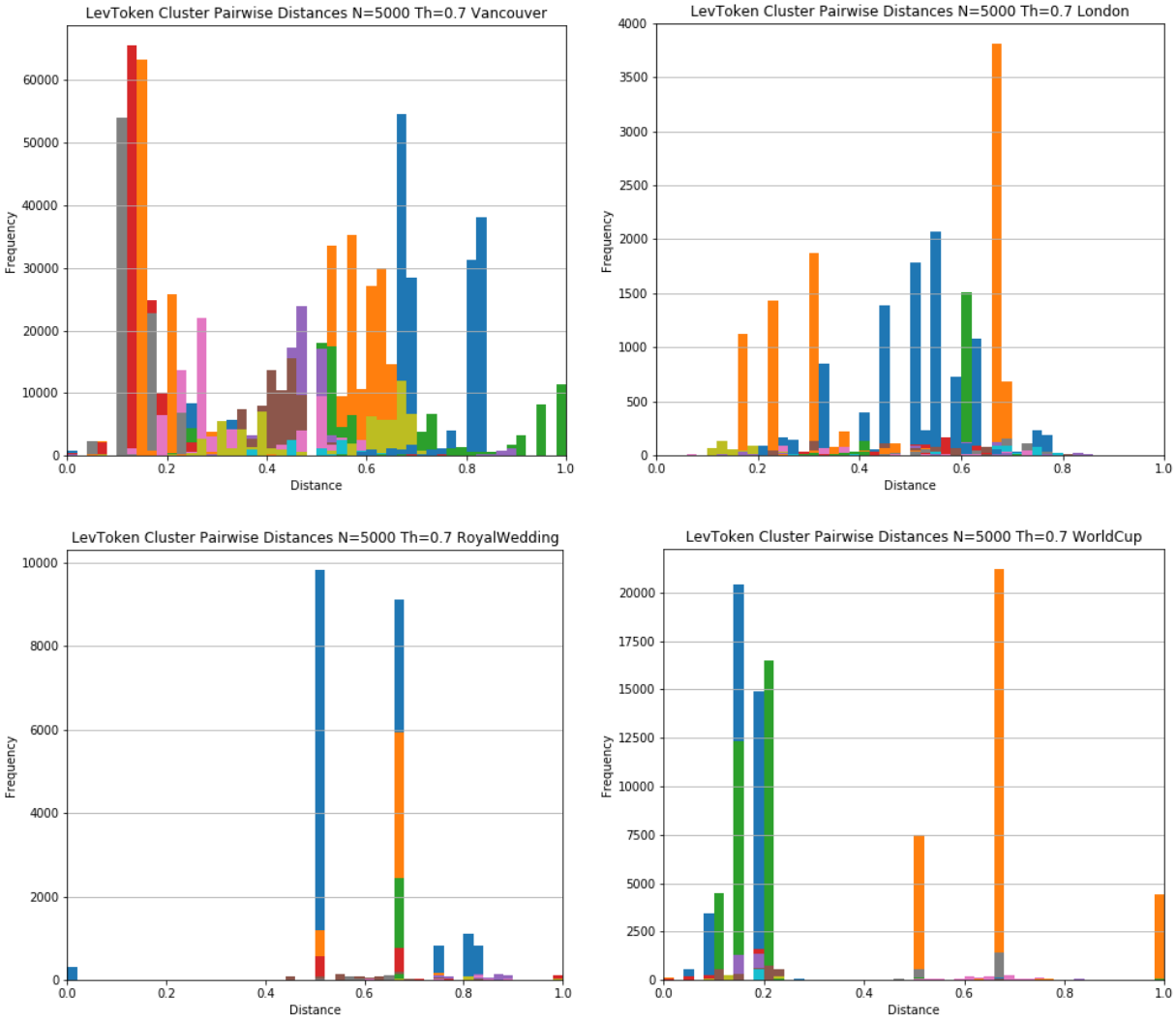

Figure 4.14: Levenshtein Threshold 0.7 for All Searches

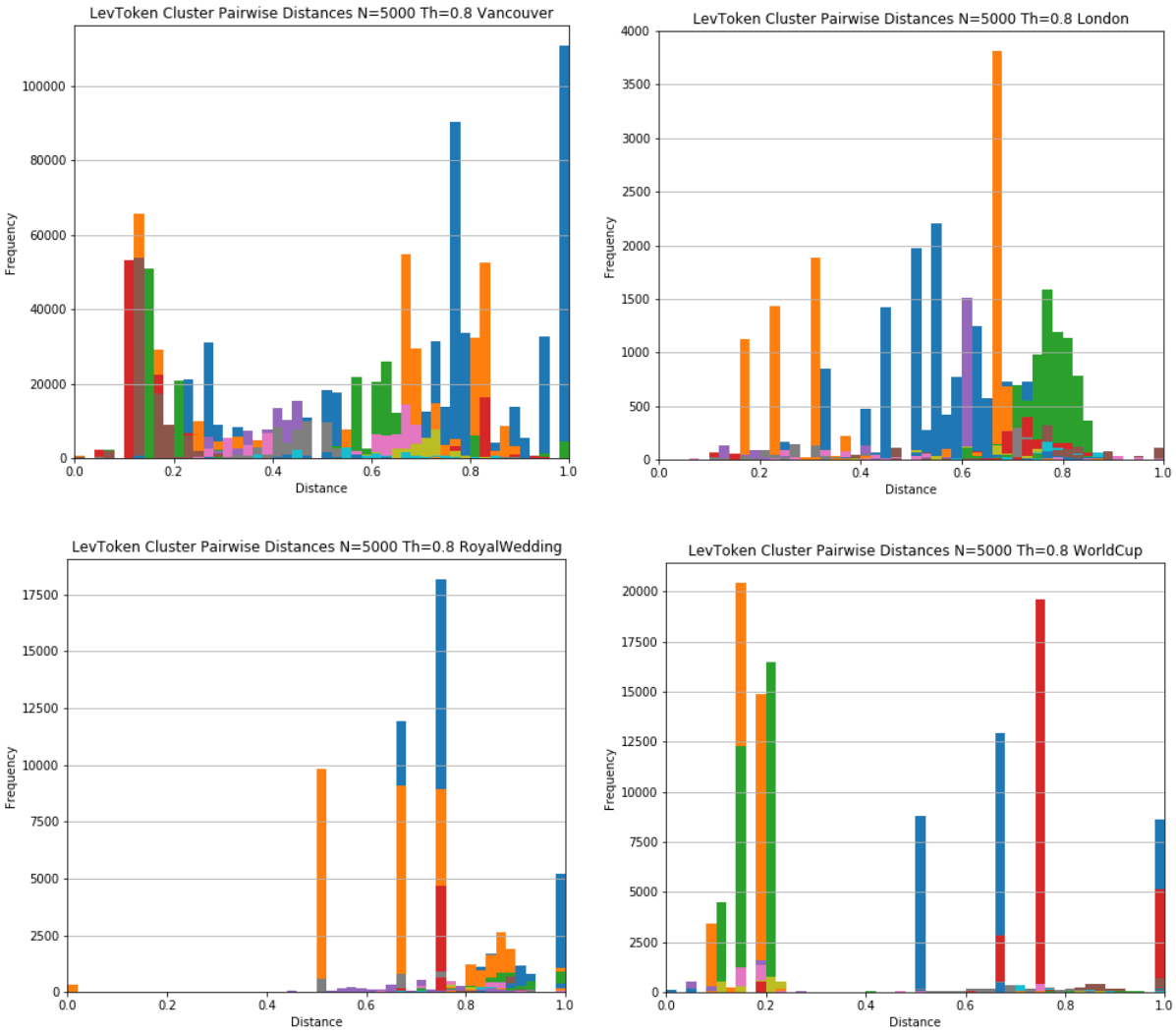

Figure 4.15: Levenshtein Threshold 0.8 for All Searches

4.1.4 Similarity Distance Thresholding Performance Comparison

While it is worth comparing clusters qualitatively, threshold parameters cannot be normalized across the similarity measures and, therefore, any given threshold value may be more or less permissive when applied to a different similarity measure. Figure 4.1 - Figure 4.15 show a few differences between each similarity measure. Firstly, T-Information results in continuous pair-wise distance clusters, whereas Jaccard and the Levenshtein token-based similarity measures result in clusters comprising narrow bands

of finite width. In a real-world implementation of the thresholding algorithm, an end user or developer would select an appropriate threshold. The continuous nature of T-Information would minimize the effects of having a sub-optimal parameter selected. A token-based similarity score carries the risk of selecting a threshold parameter that falls outside of one of the finite bands of allowable similar scores. In this case, an end user could easily cause cluster sizes to dramatically increase or decrease based on only small adjustments in the threshold. At high threshold values, as seen in Figure 4.10 and Figure 4.15, Jaccard and Levenshtein appear to establish distinct clusters, however, they are spread across multiple peaks and likely denote formatting similarity and not content similarity. At the same high thresholds, T-Information will form large clusters that include most Tweets in the set. At such high threshold values, however, the quality of clusters is significantly diminished. Ultimately, it was observed that each similarity distance functioned best at the lowest threshold 0.4.

4.2 Effects of Sample Size

The effects of sample size on cluster characteristics was explored. In Figures 4.16 - 4.31, the cluster characteristics are shown for sample size variations from 250 to 1000 samples in 250 sample increments. Each experiment was run 10 times to generate an average and standard deviation error. For each plot, the mean value is represented by a dot and with standard deviation error bars displayed across all runs.

The presented figures are grouped into cluster size, reduction, complexity, and intra-cluster distance characteristics. The cluster size characteristics include the number of clusters determined for each search and the largest cluster size for each search. The reduction characteristics present the number of unclustered posts and the reduction of the total dataset. The complexity characteristics show the amount of time and the number of calculations required to cluster each search. The complexity characteristics also include a comparison to the worse case scenario for each algorithm, which is $O(N^2)$. As time is highly dependent on computer hardware and software

implementation, absolute time and calculations per second are presented for context only. Finally, the intra-cluster distance characteristics are presented as an averaged Root Mean Square Distance (RMSD) for all clusters. As each similarity measure fundamentally denotes a distinct measure space, the RMSD for each measure is presented independently. In the case that no clusters were found for a search, the RMSD was set to the greater value between 1 or twice the minimum threshold value, which is the largest possible RMSD for a given cluster.

4.2.1 Cluster Size Characteristics by Sample Size

As can be seen in Figures 4.16, 4.20, 4.24, and 4.28, cluster size characteristics vary both on sample size and by search. As can be observed, both the number of clusters and the maximum cluster size generally increase as the sample size increases. The T-Information similarity measure consistently finds more clusters for a given search for all sample sizes. T-Information seems to dramatically increase the number of clusters for the sparse data sets WorldCup and RoyalWedding. This is likely due the character wise nature of the similarity measure. It is also likely that T-Information is able to reliably measure the similarity between short Tweets, whereas the token measures cannot. The Levenshtein and Jaccard string token measures perform similarly for both the modified and I-TWEC thresholding algorithms for all sample sizes. This is expected behaviour as I-TWEC and the modified thresholding algorithm perform similarly for a minimum cluster size of 2.

Upon inspection of the results, it is obvious that the maximum cluster size generally increases for an increased sample size. It is apparent, however, increasing maximum clusters size for increasing sample size is not always guaranteed. This phenomenon is observed for the Royal Wedding in Figure 4.24 where the maximum size of cluster decreases from sample size 500 to sample size 1000. As the decrease in maximum cluster size is only apparent for the T-Information similarity distance, this may indicate there exists two dominant cluster types in Royal Wedding that are neighbours in the T-Information distance space. It is also shown in Figure 4.24 the Royal Wedding search experiences the greatest variance in maximum cluster size. The results would

suggest each thresholding algorithm frequently finds and clusters the same dominant cluster independent of sample size and each similarity measures determines how many posts are included in the dominant cluster. Finally, in no cases did the average maximum cluster size or one standard deviation greater than the average maximum cluster size exceed one tenth of the sample size. This suggests an absolute upper bound on a minimum cluster size is one tenth of the sample size and it is likely a practical minimum cluster size is lower. Royal Wedding, for example, appears to only reliably form clusters for a minimum cluster size of 2 at 500 samples or more.

As can be seen in Figures 4.16 for the compressible search Vancouver, there is a high number of clusters and a small variance in the numbers across all similarity measures and algorithm implementations when compared to the other searches. Vancouver exhibits an interesting behaviour where both algorithms and all similarity measures find very similar maximum cluster sizes for each sample. This is likely due to the algorithms clustering the same set of posts for a given run and appears to be independent of sample size. London, WorldCup, and RoyalWedding, shown in Figures 4.20, 4.24, and 4.28, all exhibit behaviour for max cluster size, whereby each similarity measure finds a very similar maximum cluster size irrespective of the clustering algorithm for all sample sizes.

4.2.2 Reduction Characteristics by Sample Size

Reduction represents the relationship between the size of the original dataset and the number of unclustered Tweets remaining after the clustering operation. The Reduction gives an indication of how compressible the dataset is for a given algorithm and similarity measure. Figures 4.17, 4.21, 4.25, and 4.29 show the reduction and the number of unclustered tweets for each search. As seen in the plots, there is a strong linear relationship between sample size and reduction as well as the number of unclustered posts. Reduction and the number of unclustered post are mostly constant but improve with the sample size. This suggests as more possible clusters are available each algorithm will determine them and cluster more posts. The more clusters present in the data, for example Vancouver, the greater reduction available. Both thresholding

algorithms perform similarly for all sample sizes. For the compressible data set Vancouver, in Figure 4.17, the Levenshtein string-token similarity measure clusters the greatest number of posts followed by the T-Information similarity measure, for all sample sizes. For all the other searches, T-Information performs similarly to Levenshtein string-token for London, and the best for WorldCup and RoyalWedding for data reduction for all sample sizes.

4.2.3 Complexity Characteristics by Sample Size

Thresholding algorithm complexity characteristics were compared for the same sample sizes and the results can be seen in Figures 4.18, 4.22, 4.26, and 4.30. The clustering algorithms were measured for the total time elapsed, the number of calculations required, the calculations required as compared to the worst-case $O(N^2)$, and the number of calculations per second.

As expected, the time and the number of calculations required for all algorithms and all similarity measures increased non-linearly with the sample size for each search. Another result consistent across all searches is the number of calculations compared to the worst case reduces as the sample size increases. This is largely attributable to the increased number of clusters and increased number of Tweets included in each cluster reducing the number of total comparisons each algorithm must make. Further, it can be seen the number of calculations per second is consistent across all searches and independent of sample size. It can also be seen that T-Information and Jaccard similarity distances are nearly twice as fast as the Levenshtein distance in all cases. Figures 4.18, 4.22, 4.26, and 4.30 also show the search type and algorithm strongly influence the number of calculations required. As seen in Figure 4.18, Vancouver has a low number of total calculations as compared to the worst case and both thresholding algorithms perform similarly with the modified thresholding algorithm performing slightly better for all sample sizes. For the less clusterable data sets, London, RoyalWedding, and Worldcup, in Figures 4.22, 4.26, and 4.30, the ITWEC thresholding algorithm

requires far more calculations and operates closer to the worst case than the modified algorithm for all sample sizes.

4.2.4 RSMD Characteristics by Sample Size

Root Mead Squared Distance (RMSD) is a simple measure for the cluster quality that is calculated using the square distance of all intra-cluster pairwise distances and generating an average. The smaller the value for RMSD the tighter the clusters. RMSD is a function of the similarity distance, and as a result each measure is presented independently. As can been seen in Figures 4.19, 4.23, 4.27, and 4.31 RMSD generally decreases as sample size increases for all searches across algorithms and similarity measures.

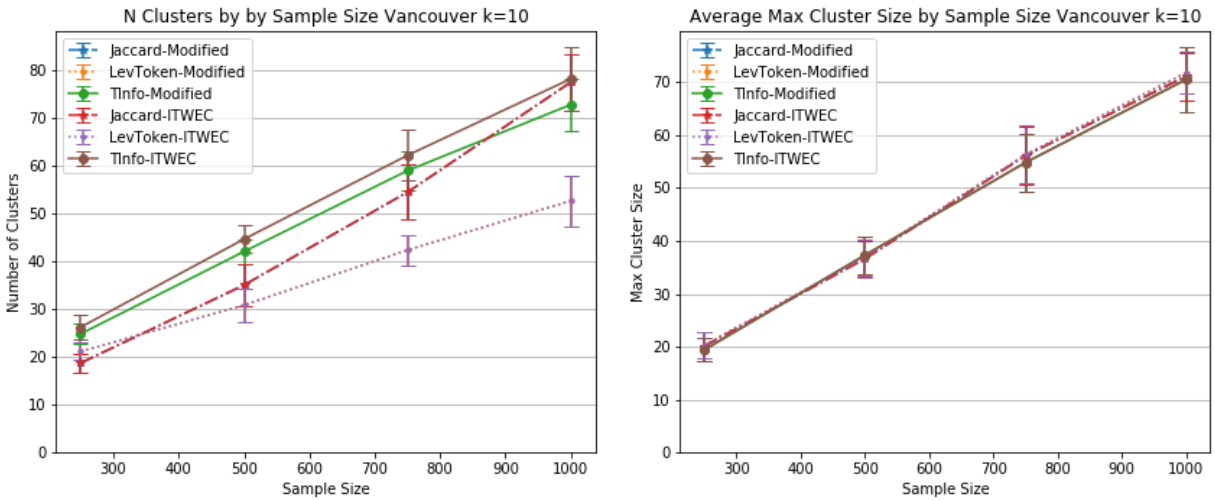

Figure 4.16: Cluster Size Characteristics by Sample Size Vancouver

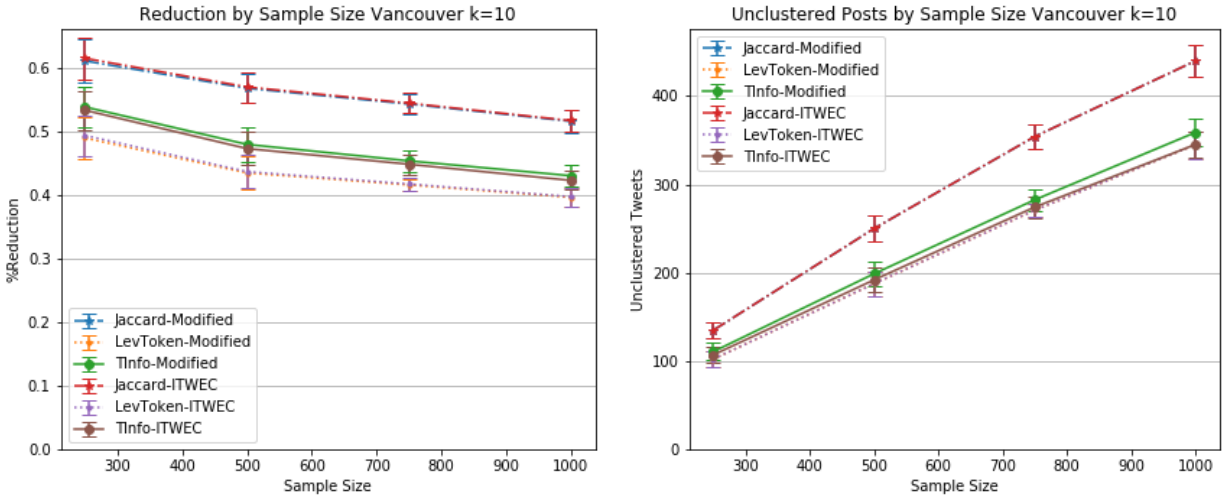

Figure 4.17: Reduction Characteristics by Sample Size Vancouver

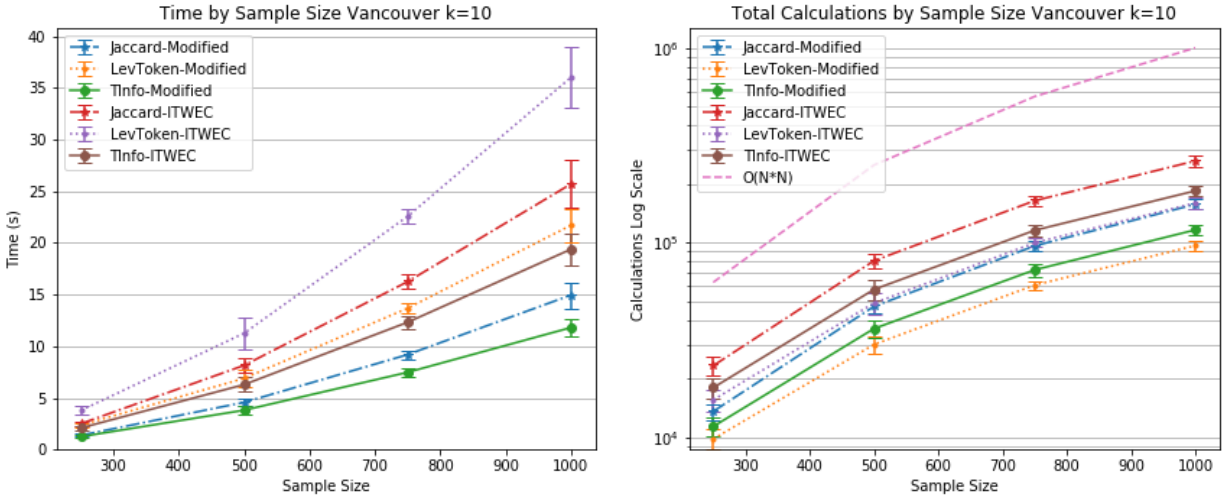

Figure 4.18: Complexity Characteristics by Sample Size Vancouver

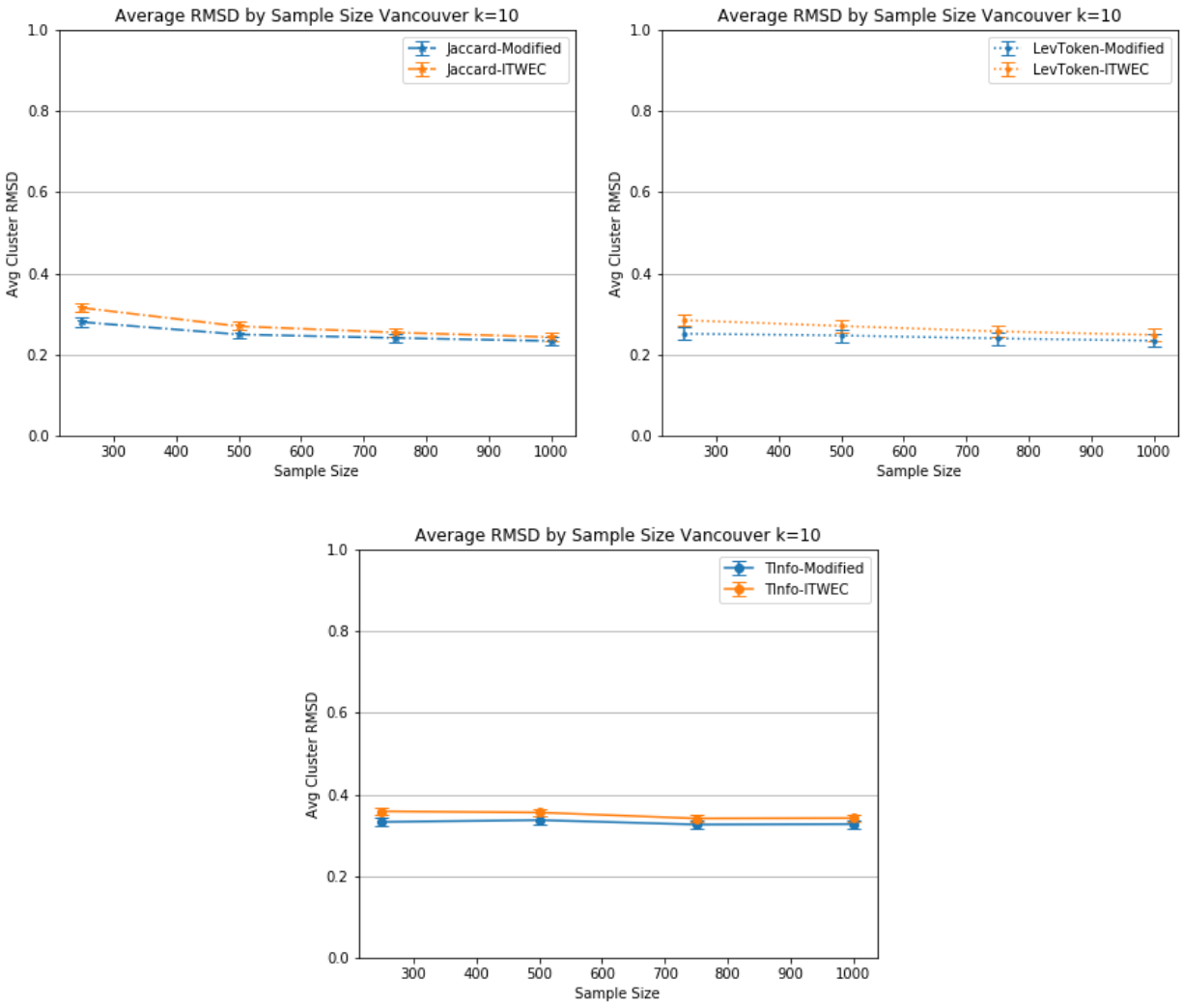

Figure 4.19: RMSD Characteristics by Sample Size Vancouver

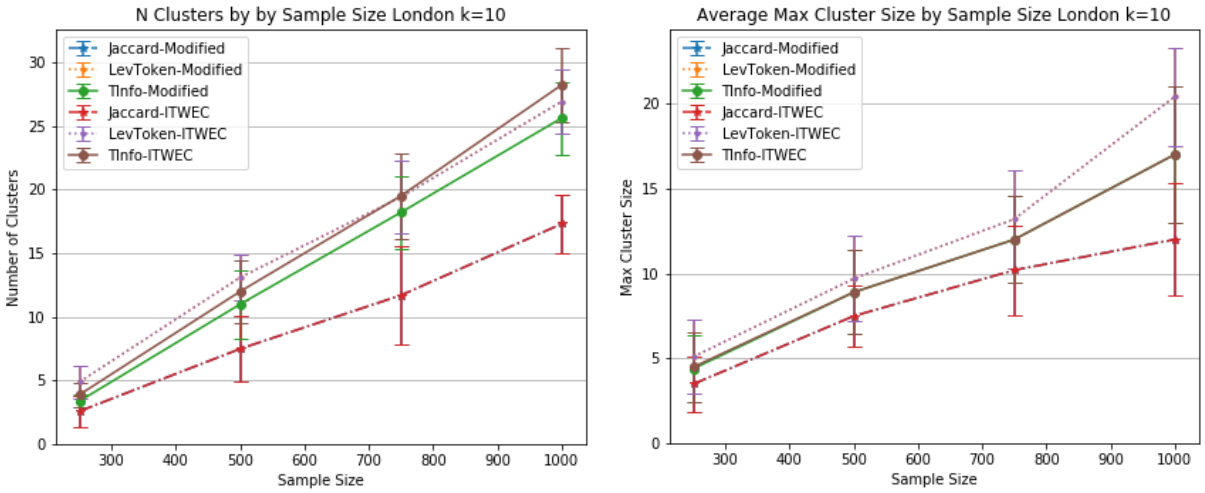

Figure 4.20: Cluster Characteristics by Sample Size London

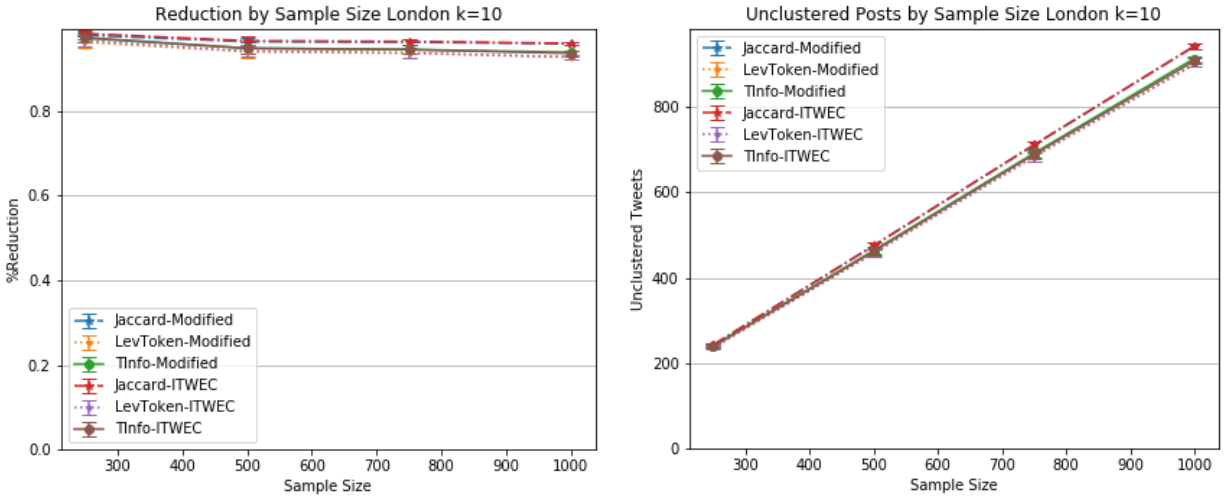

Figure 4.21: Reduction Characteristics by Sample Size London

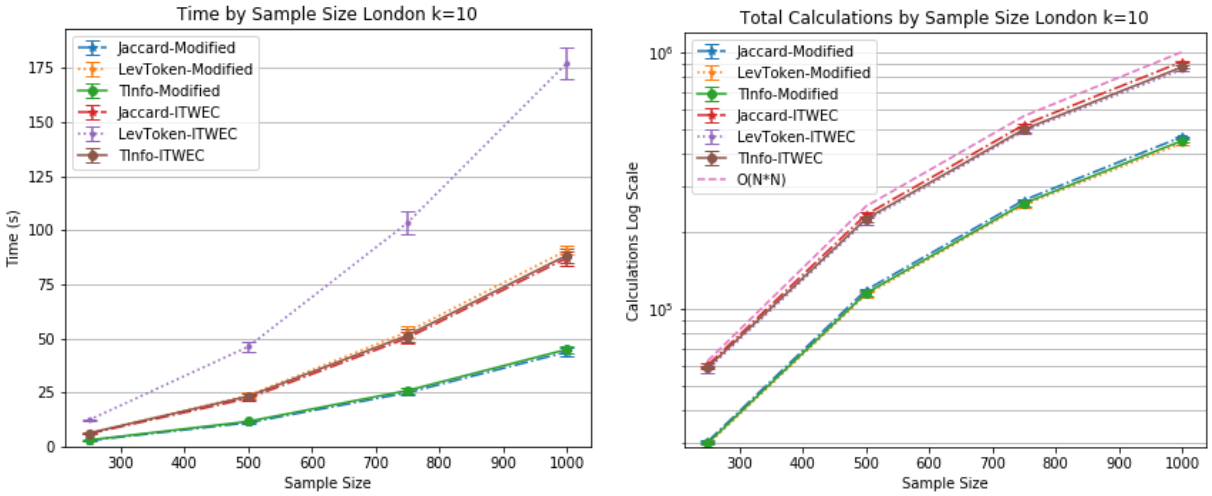

Figure 4.22: Complexity Characteristics by Sample Size London

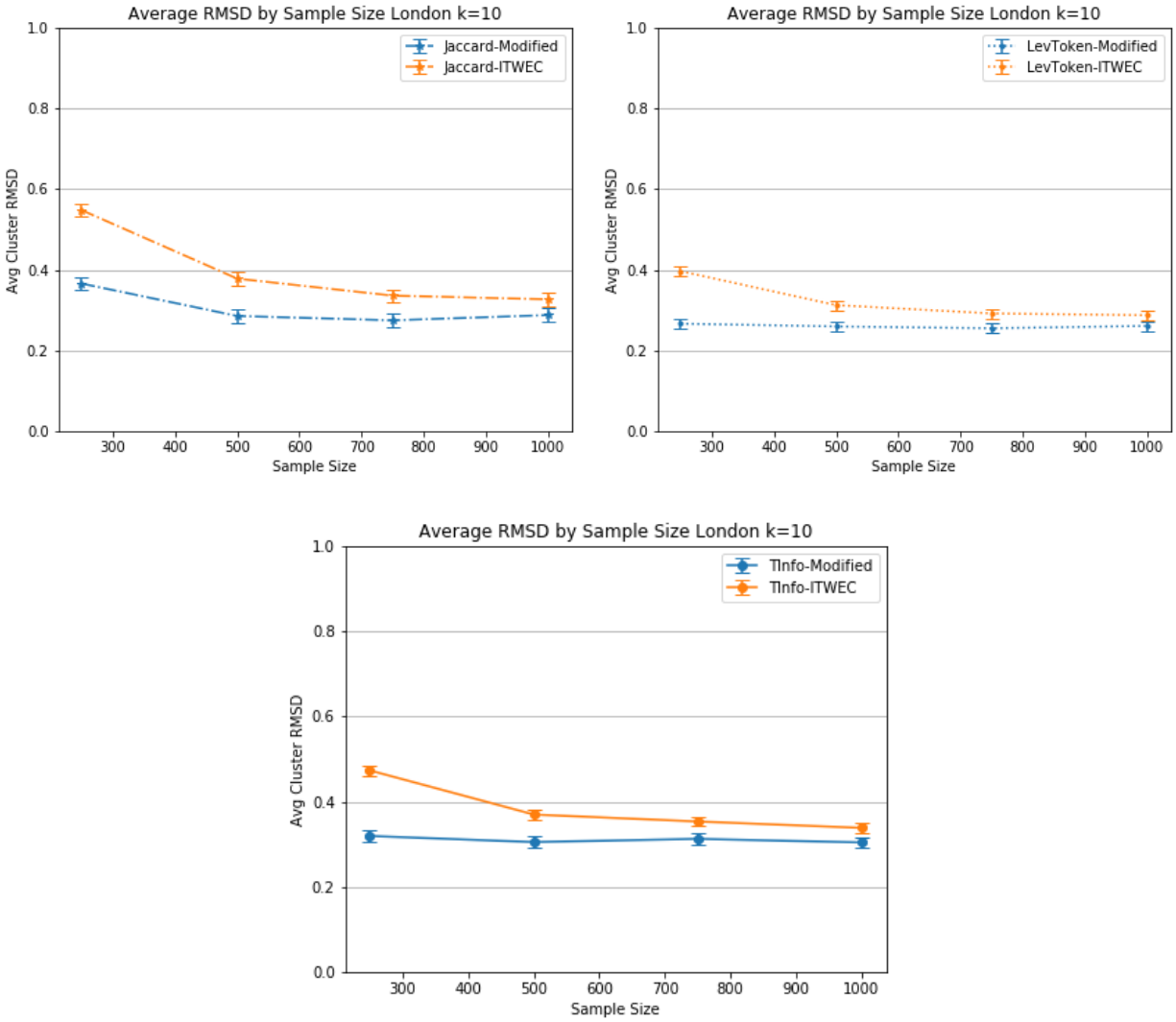

Figure 4.23: RMSD Characteristics by Sample Size London

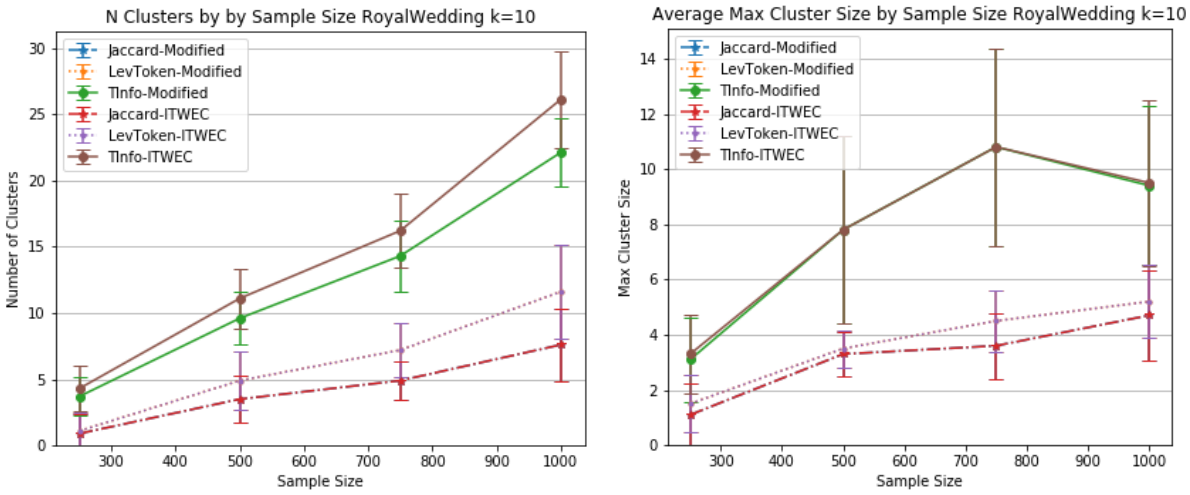

Figure 4.24: Cluster Characteristics by Sample Size Royal Wedding

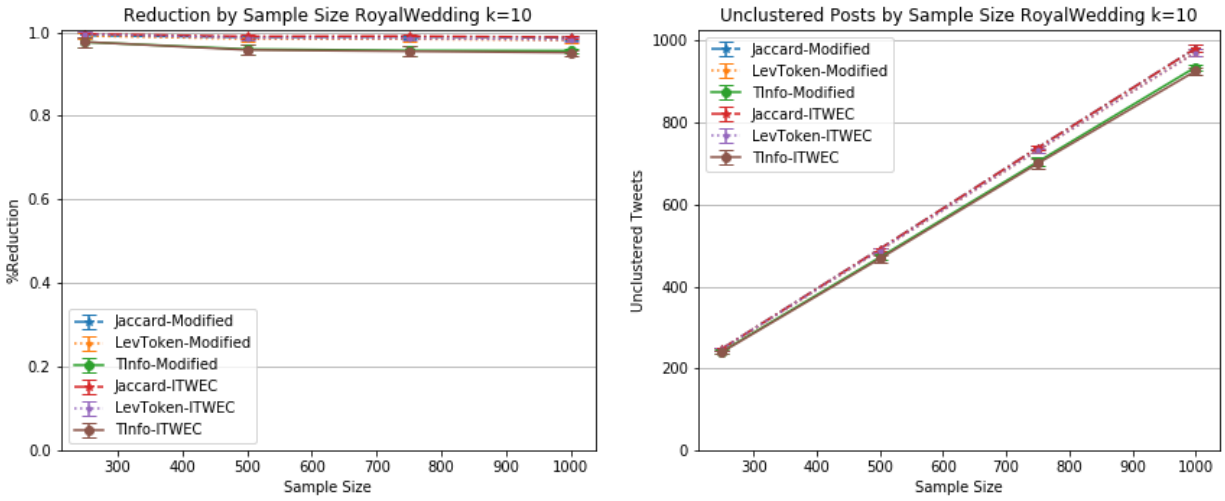

Figure 4.25: Reduction Characteristics by Sample Size Royal Wedding

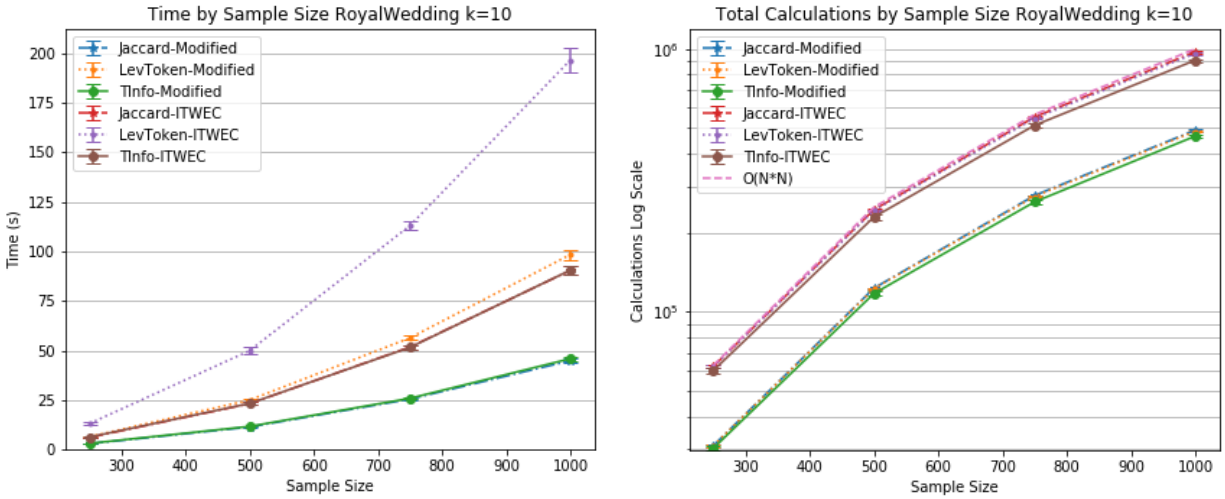

Figure 4.26: Complexity Characteristics by Sample Size Royal Wedding

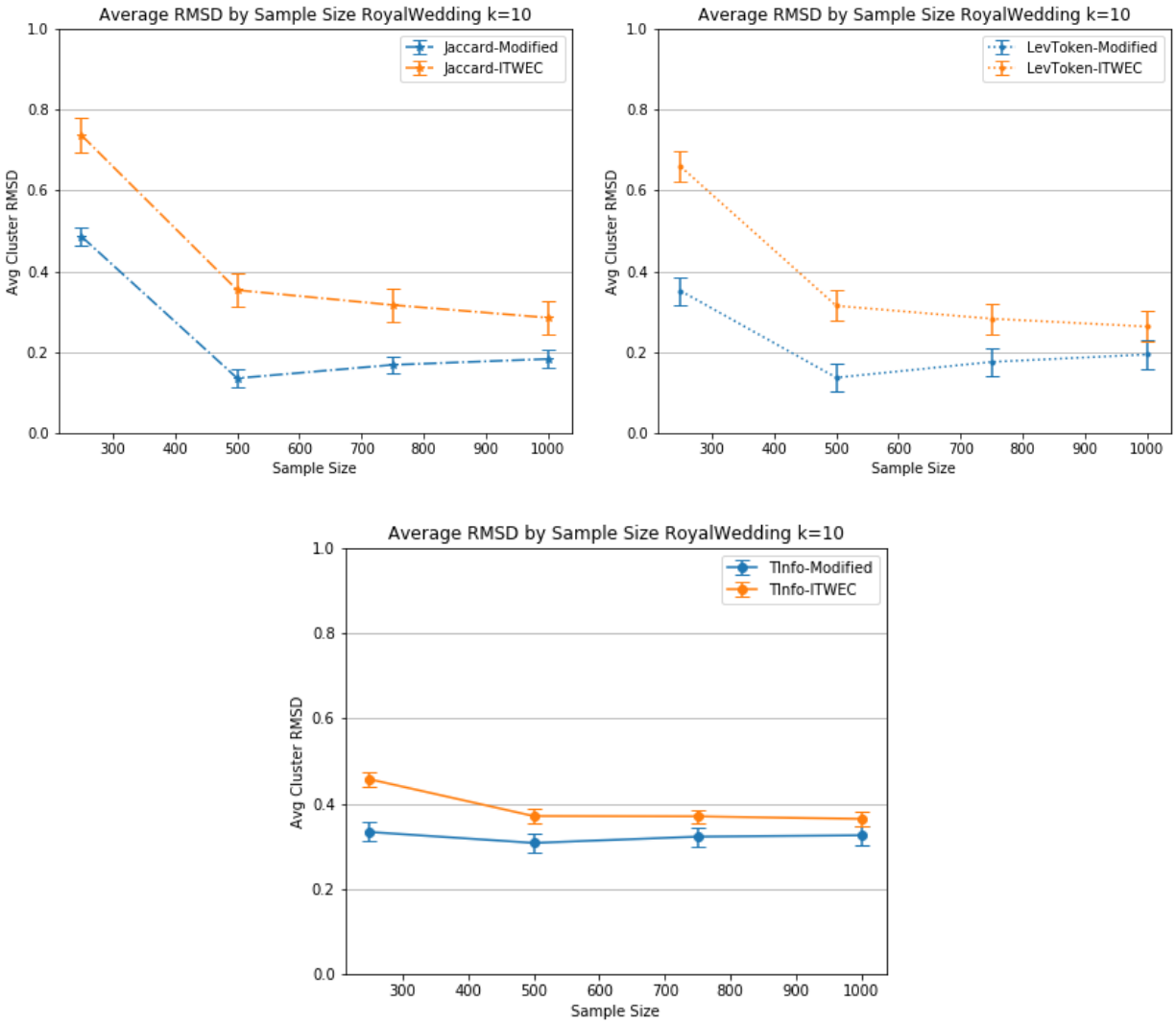

Figure 4.27: RMSD Characteristics by Sample Size Royal Wedding

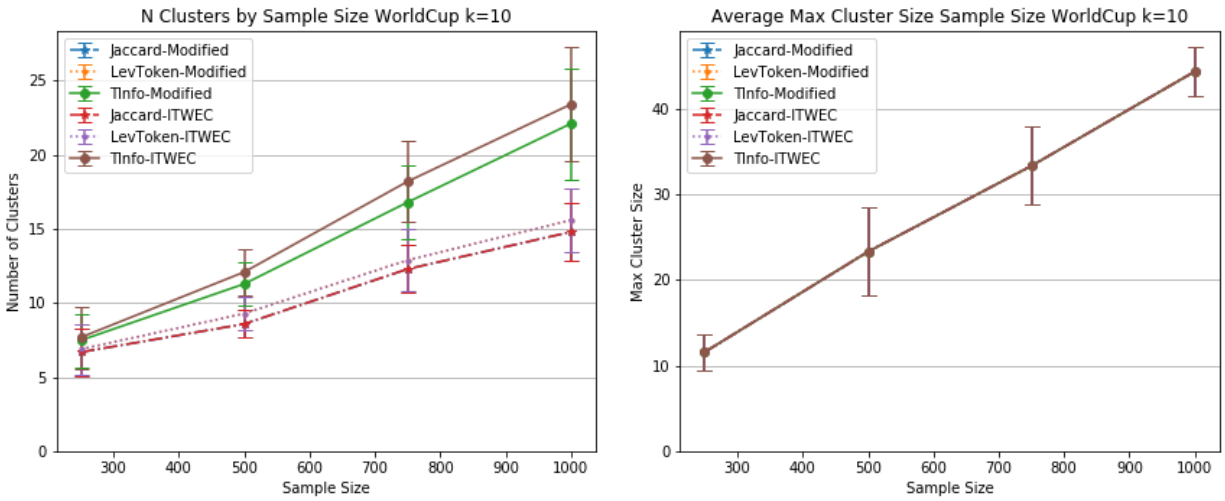

Figure 4.28: Cluster Size Characteristics by Sample Size WorldCup

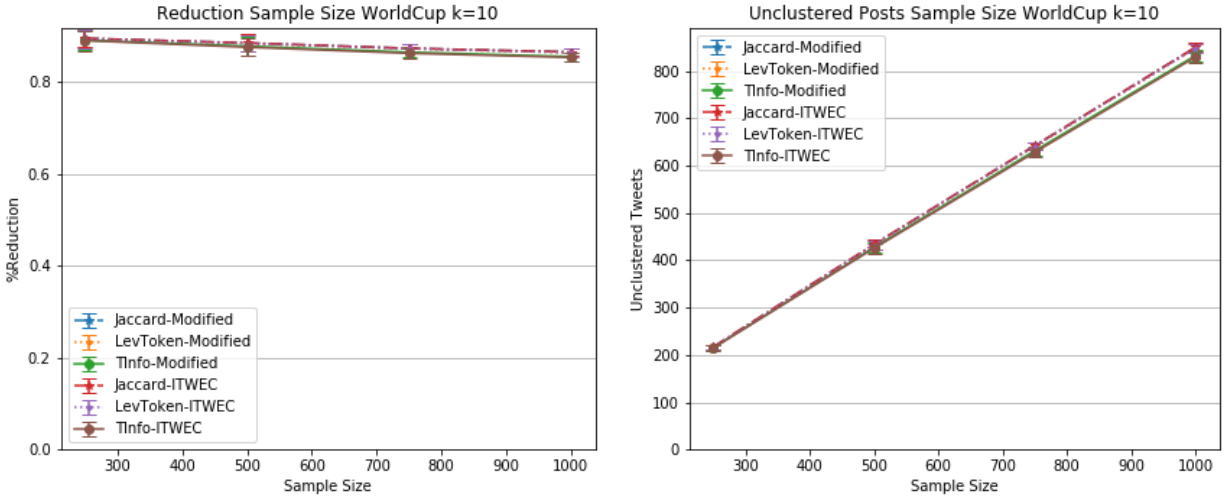

Figure 4.29: Reduction Characteristics by Sample Size WorldCup

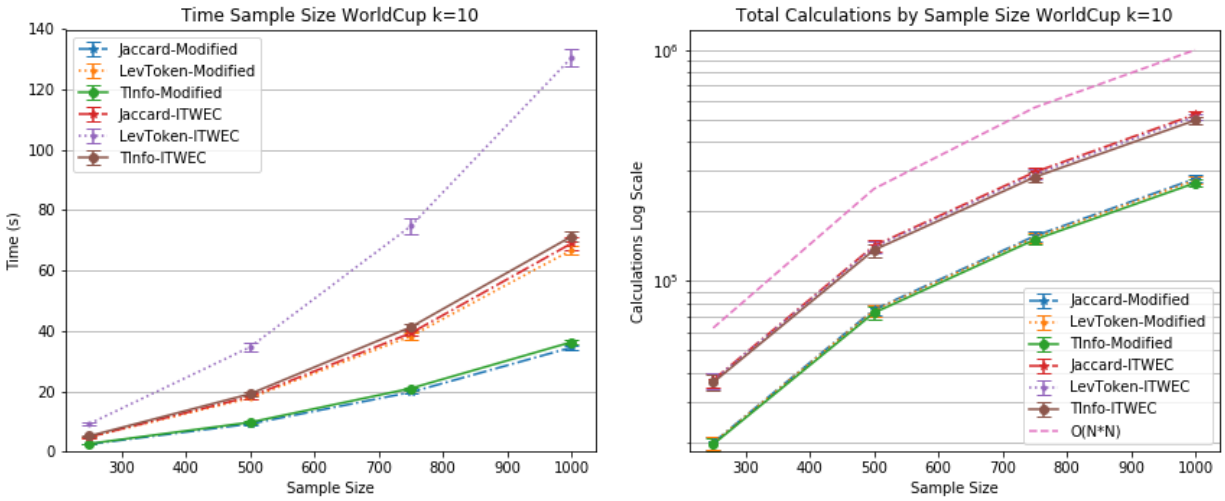

Figure 4.30: Complexity Characteristics by Sample Size WorldCup

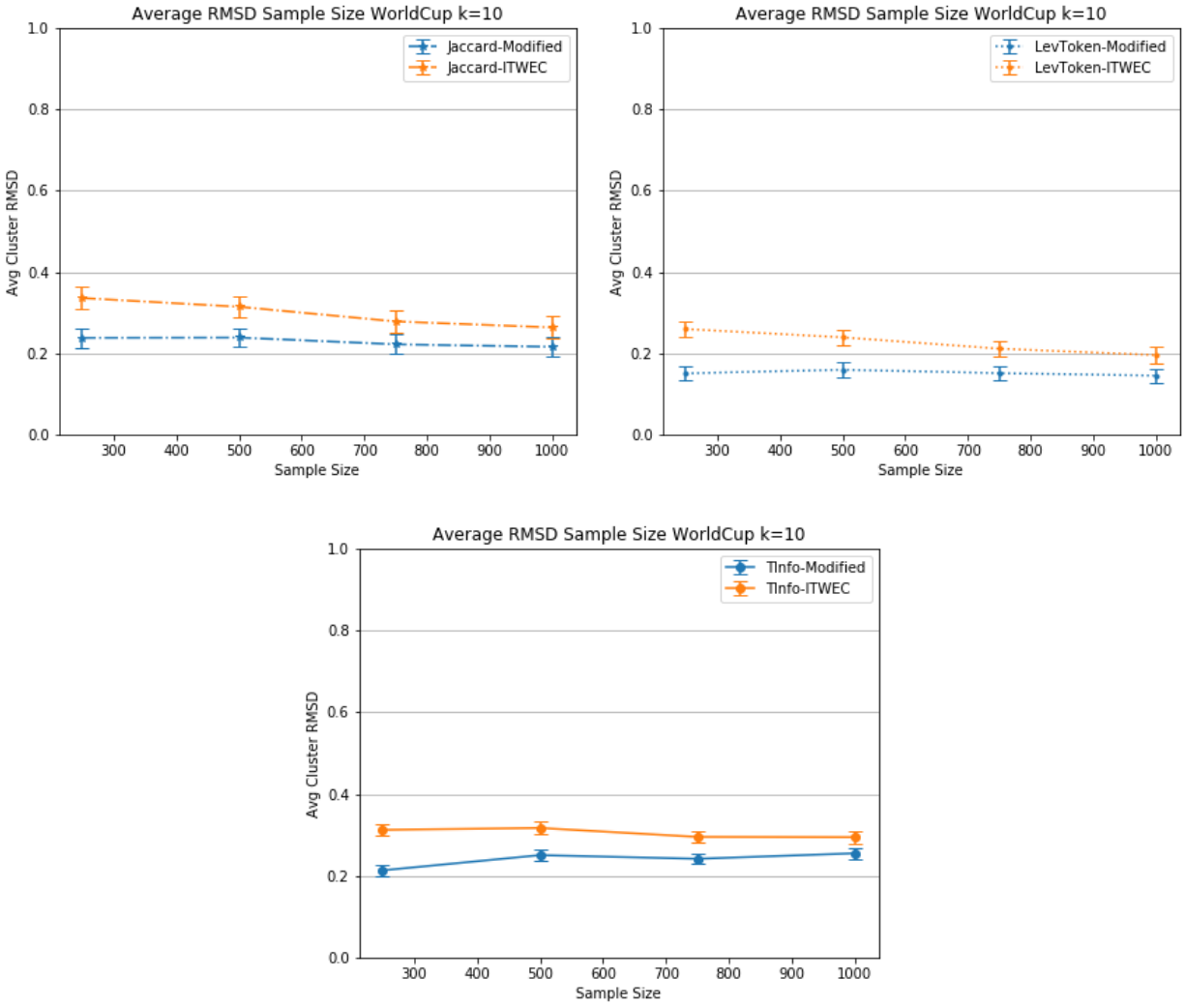

Figure 4.31: RMSD Characteristics by Sample Size WorldCup

4.3 Effects of Minimum Cluster Size

To effectively measure the effects of minimum clusters size, a 2000 Tweet sample was taken for each search and clustered for varying minimum cluster sizes. The minimum cluster values for this experiment were set at 2, 5, and 10 Tweets to represent pairwise, small, and moderately sized clusters. Figures 4.32- 4.47 represent the cluster characteristics for each search as a function of minimum cluster size.

4.3.1 Cluster Size Characteristics by Minimum Cluster Size

Figures 4.32, 4.36, 4.40, 4.44 show the cluster size characteristics as a function of minimum cluster size. As can be seen in the figures, the total number of clusters across all algorithms, similarity measures, and searches decrease as the minimum cluster size increases. Logically, this reduction in total clusters is a result of disregarding the smaller clusters in the set. As can also be seen in the figures, the average maximum cluster size appears to be independent of minimum cluster size. The one exception, RoyalWedding in Figure 4.40, frequently failed to find clusters larger than the minimum cluster size for Levenshtein and Jaccard string token similarity measures, whereas it still found clusters for T-Information. Ultimately, the minimum cluster size has the expected result of reducing the total number of clusters in a set for an increasing minimum cluster size and, in the case that the search data set is sparse, a large minimum cluster size may cause the clustering algorithms to fail to find any appropriately sized clusters.

4.3.2 Reduction Characteristics by Minimum Cluster Size

The reduction characteristic for each search as a function of minimum cluster size are shown in Figures 4.33, 4.37, 4.41, and 4.45. As can be seen in the figures, the total amount of data reduced from the original sample decreases for an increasing minimum cluster size, while the total number of unclustered posts increases. Clearly, as minimum cluster size increases, the number of unclustered Tweets increases as the algorithms focus on only the largest clusters.

4.3.3 Complexity Characteristics by Minimum Cluster Size

Figures 4.34, 4.38, 4.42, and 4.46 show the complexity characteristics for each search as a function of minimum cluster size. In this experiment, the worst-case $O(N^2)$ does not change as the sample size is fixed at 2000 Tweets. As a result, only the total time and number of calculations are presented. As can be seen in figures, for the modified thresholding algorithm, the time and total number of calculations are constant for different values of minimum cluster size. For the ITWEC thresholding algorithm, the total

time and number of calculations increase with the minimum cluster size. In all cases, T-Information performs similarly or better than the other similarity measures for the minimum cluster sizes tested.

4.3.4 RMSD Characteristics by Minimum Cluster Size

The Root Mean Square Distance (RMSD) for the varied cluster sizes are presented in Figures 4.35, 4.39, 4.43, and 4.47. As the minimum cluster size is increased, an increase in the RMSD is observed. Logically, as the pairwise and small clusters are removed from the system, the larger, less dense clusters dominate the RMSD measure causing the increase.

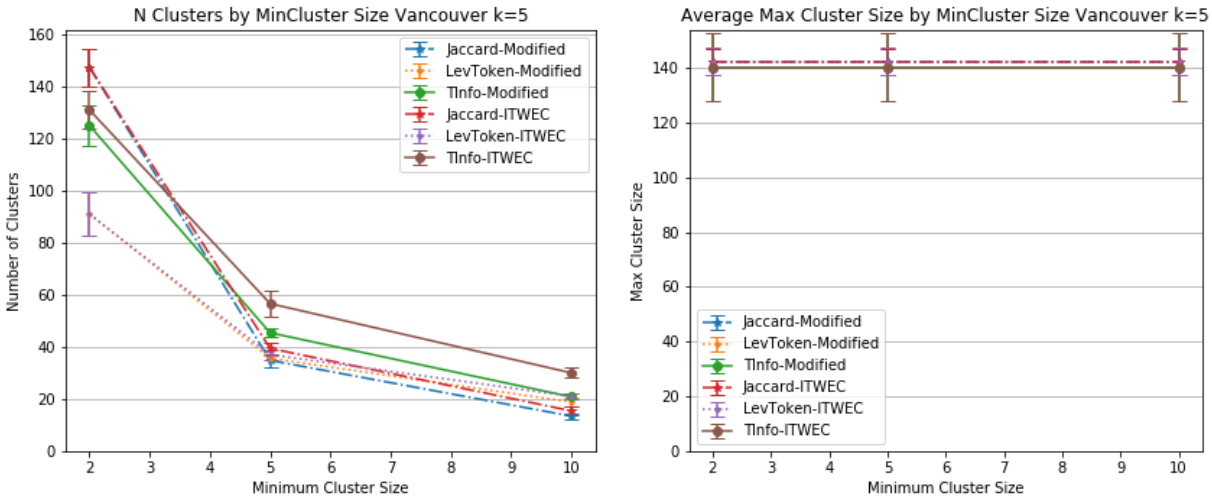

Figure 4.32: Cluster Characteristics by Minimum Cluster Size Vancouver

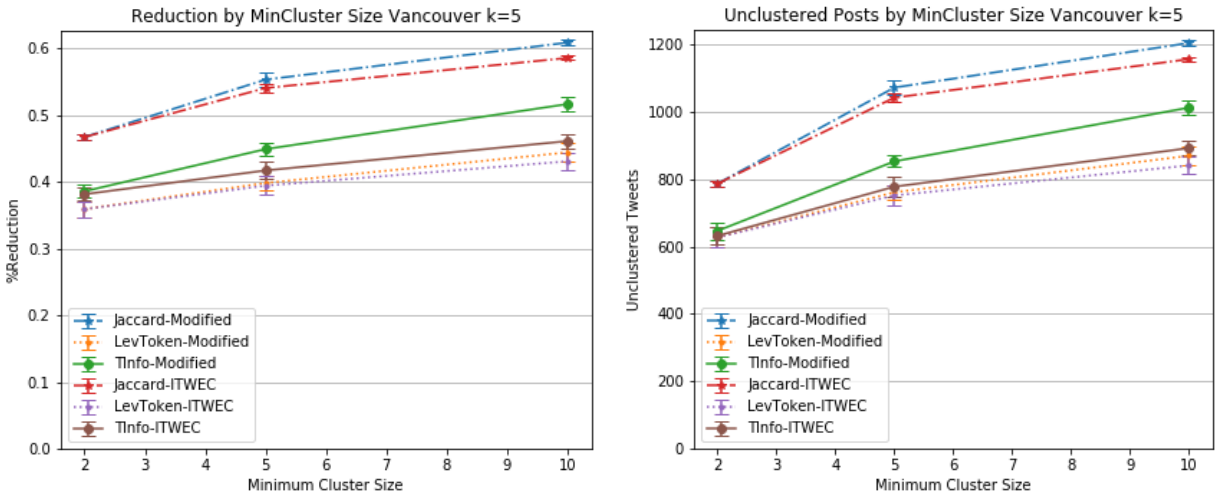

Figure 4.33: Reduction Characteristics by Minimum Cluster Size Vancouver

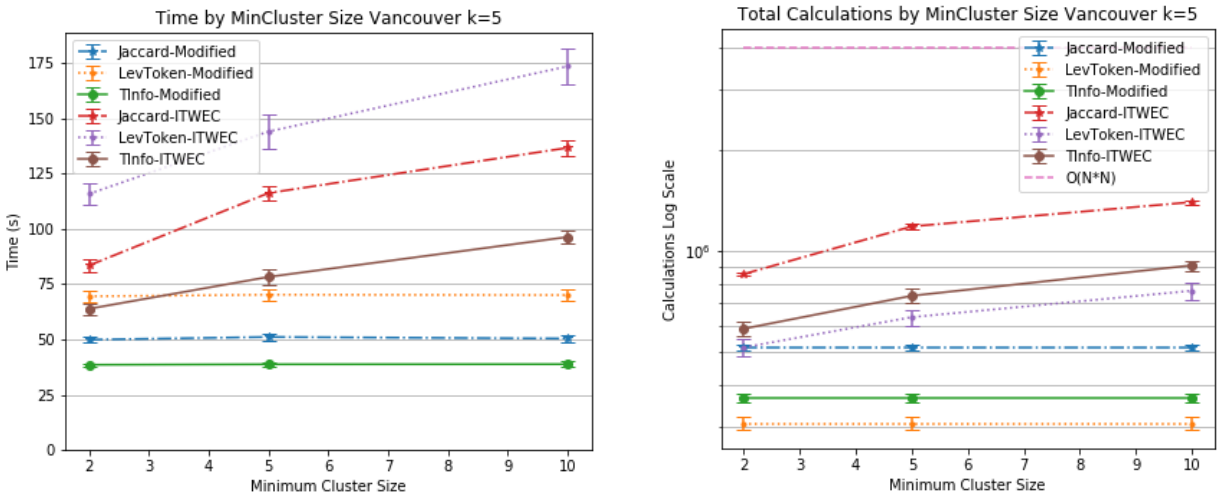

Figure 4.34: Complexity Characteristics by Minimum Cluster Size Vancouver

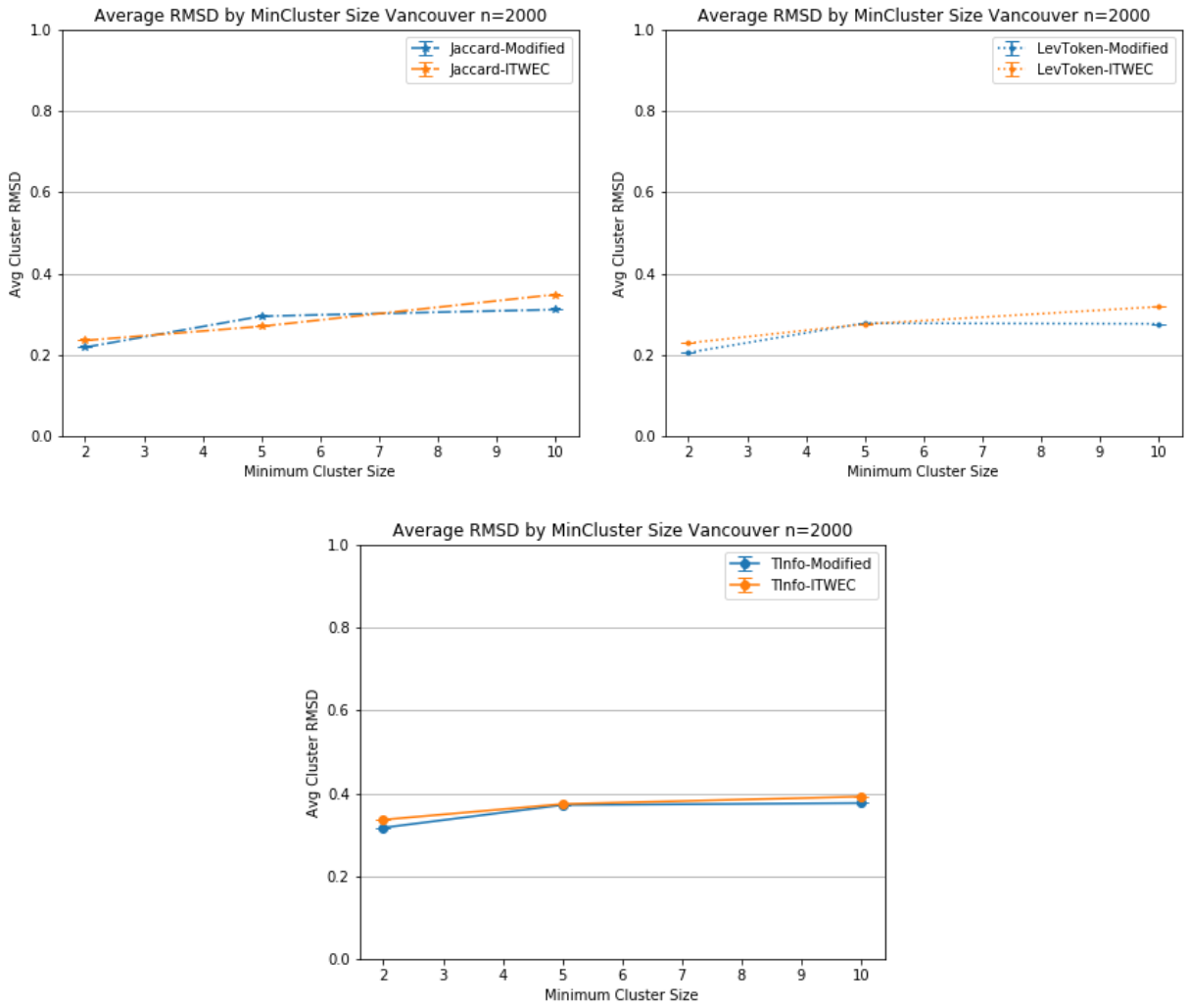

Figure 4.35: RMSD Characteristics by Minimum Cluster Size Vancouver

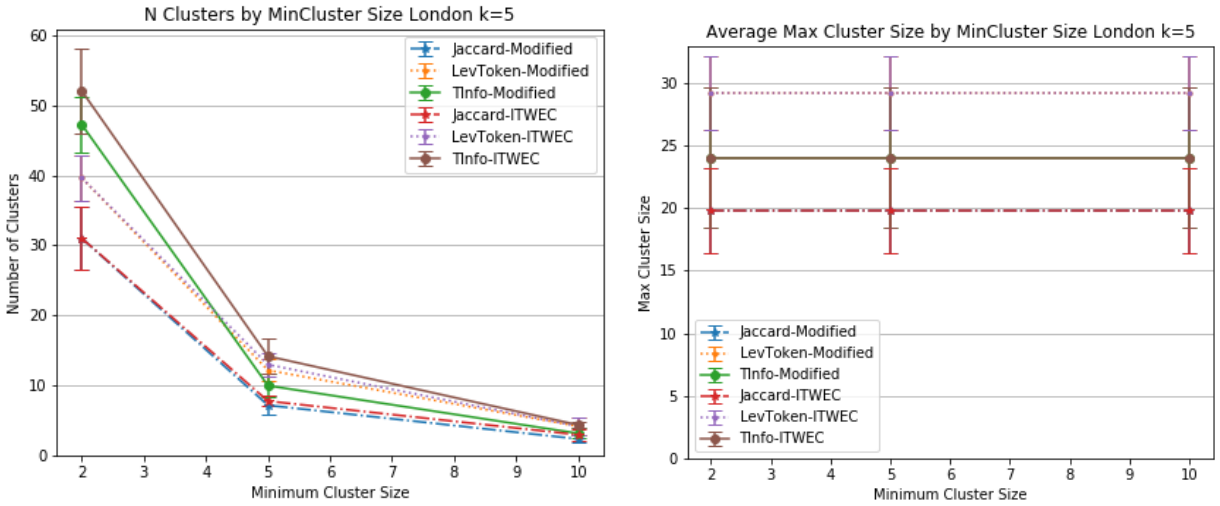

Figure 4.36: Cluster Characteristics by Minimum Cluster Size London

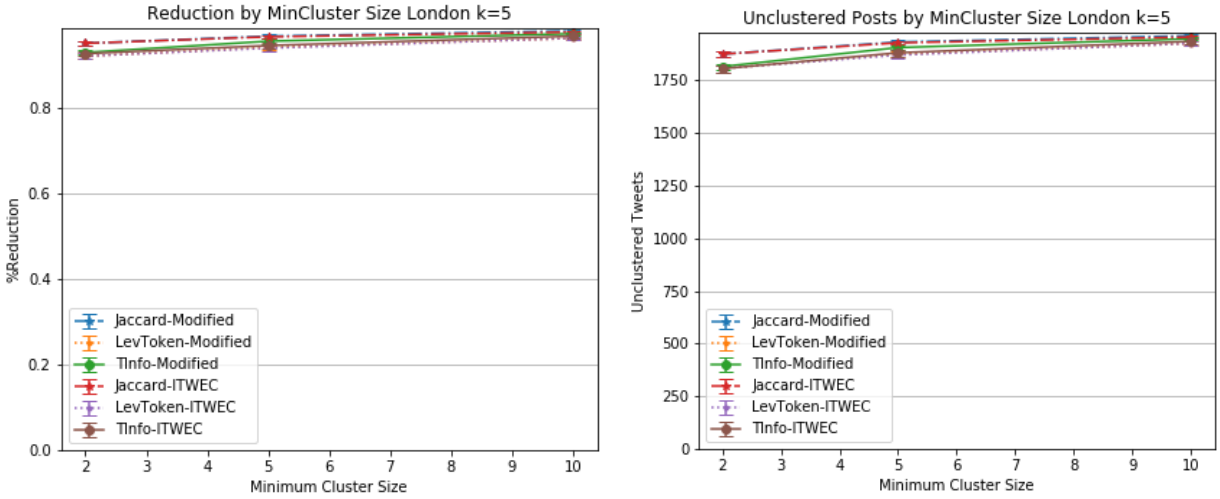

Figure 4.37: Reduction Characteristics by Minimum Cluster Size London

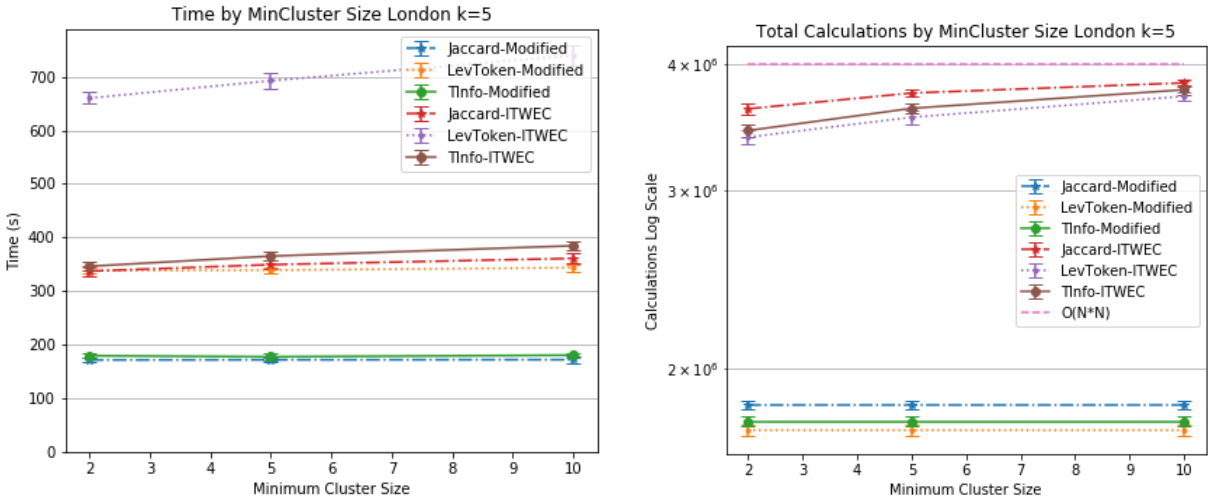

Figure 4.38: Complexity Characteristics by Minimum Cluster Size London

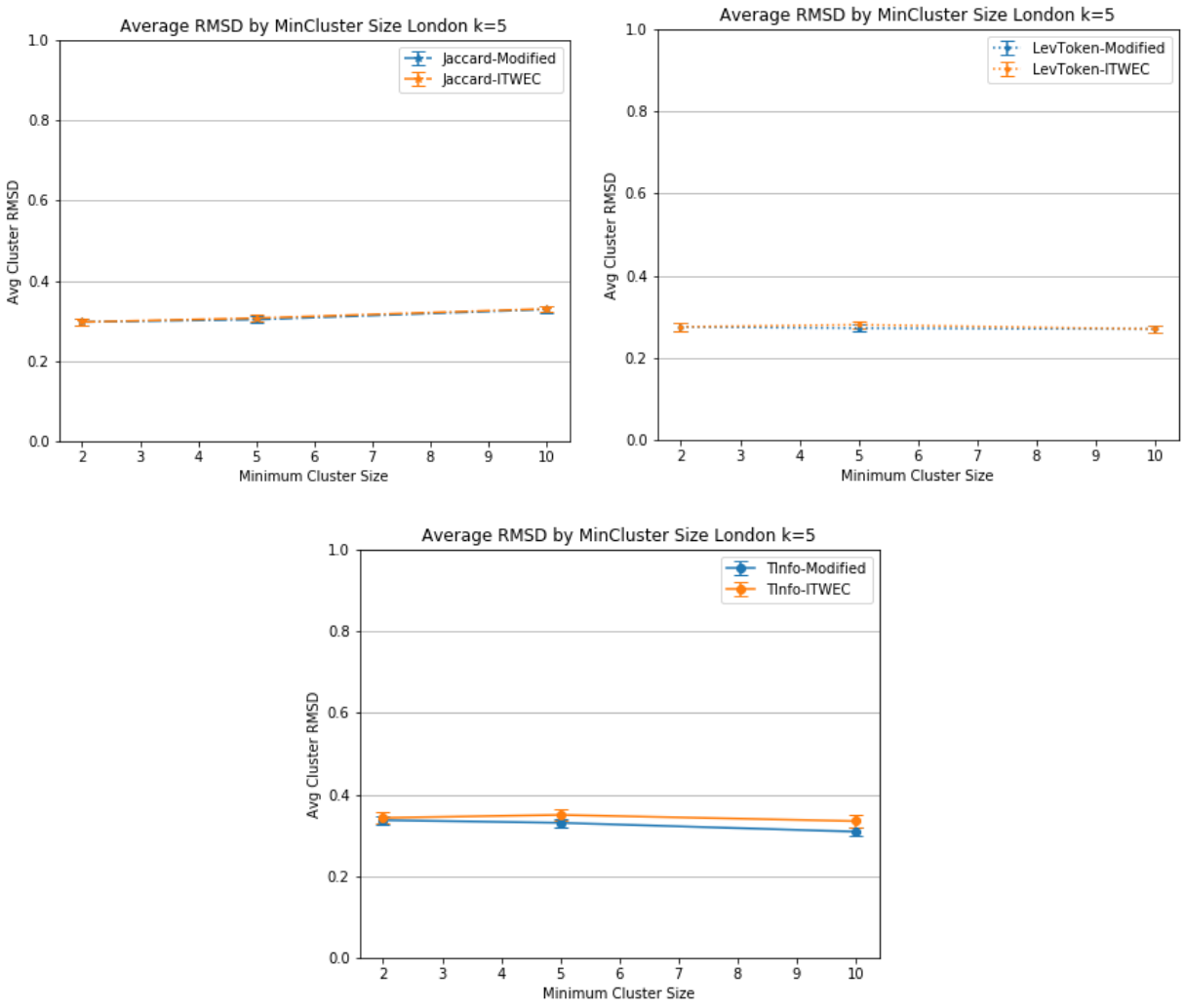

Figure 4.39: RMSD Characteristics by Minimum Cluster Size London

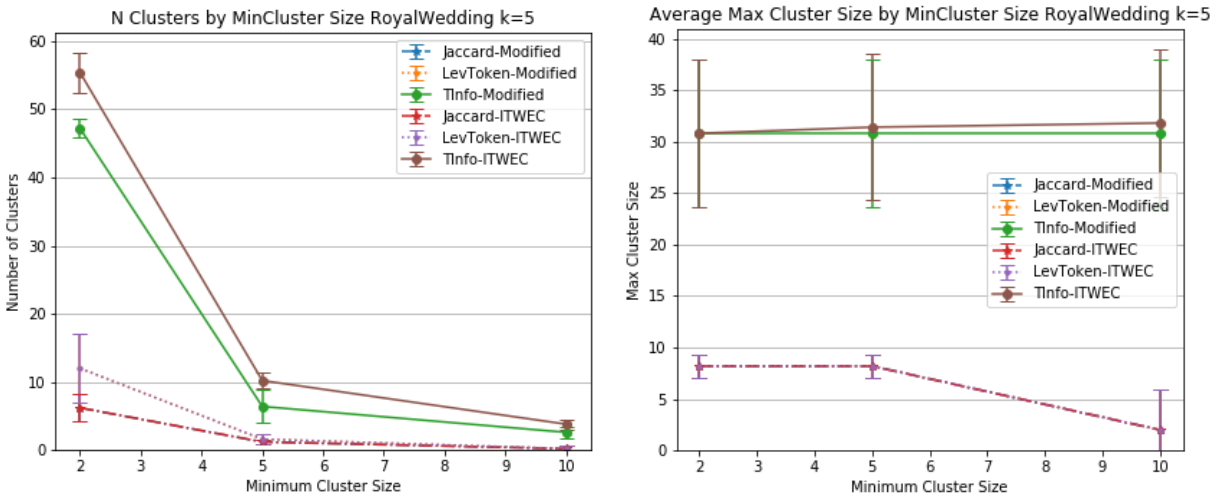

Figure 4.40: Cluster Characteristics by Minimum Cluster Size RoyalWedding

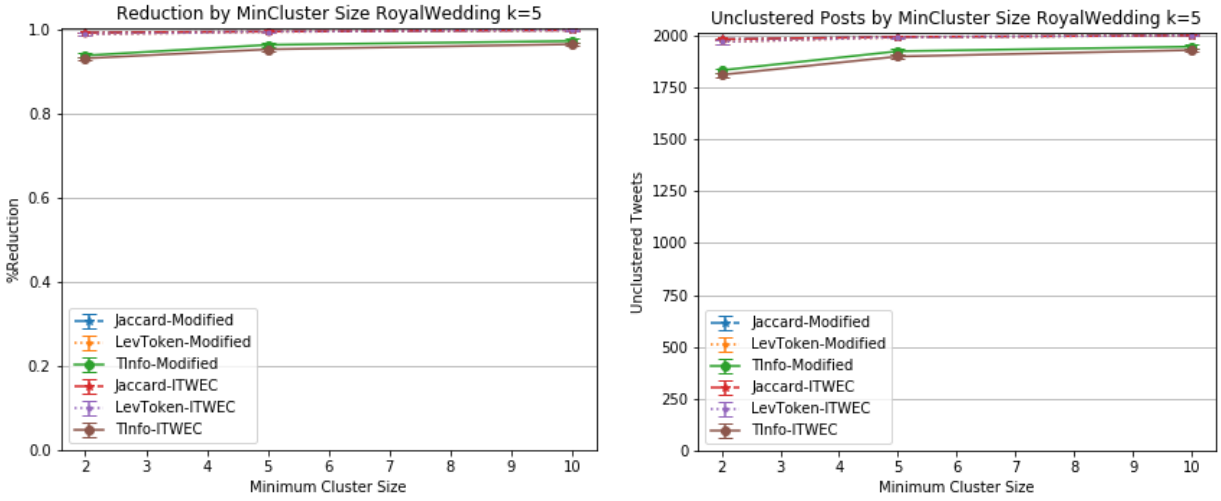

Figure 4.41: Reduction Characteristics by Minimum Cluster Size RoyalWedding

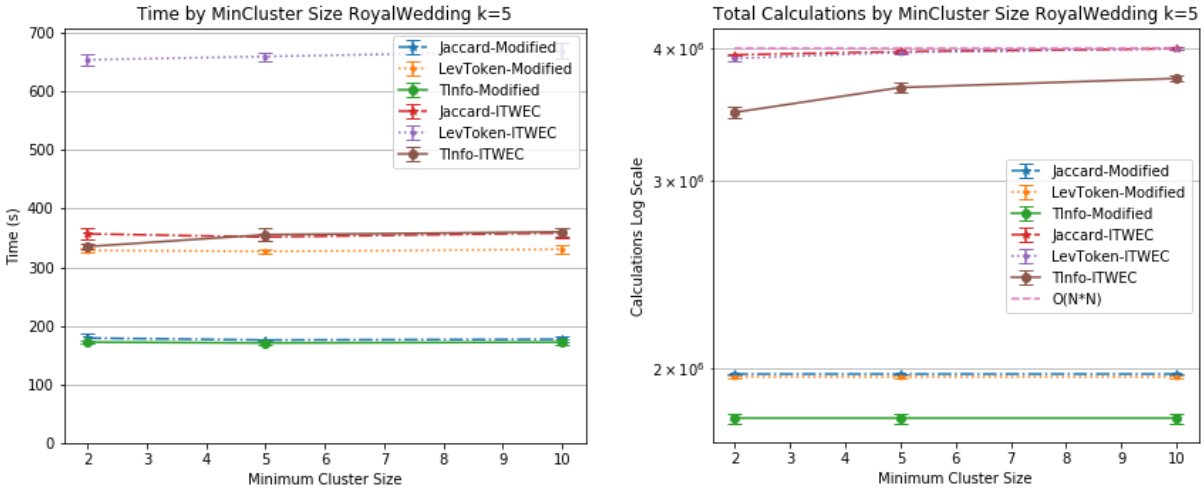

Figure 4.42: Complexity Characteristics by Minimum Cluster Size RoyalWedding

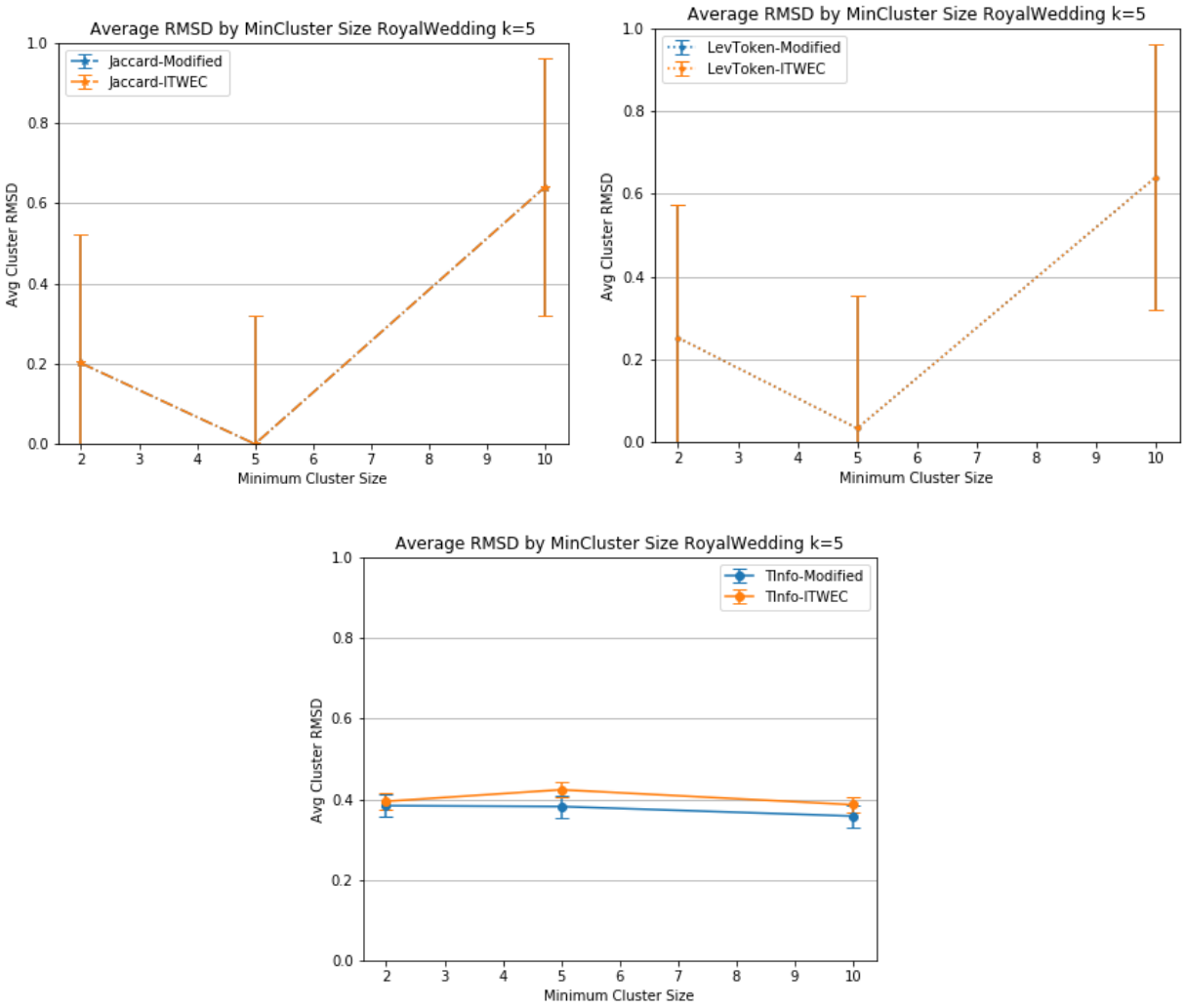

Figure 4.43: RMSD Characteristics by Minimum Cluster Size RoyalWedding

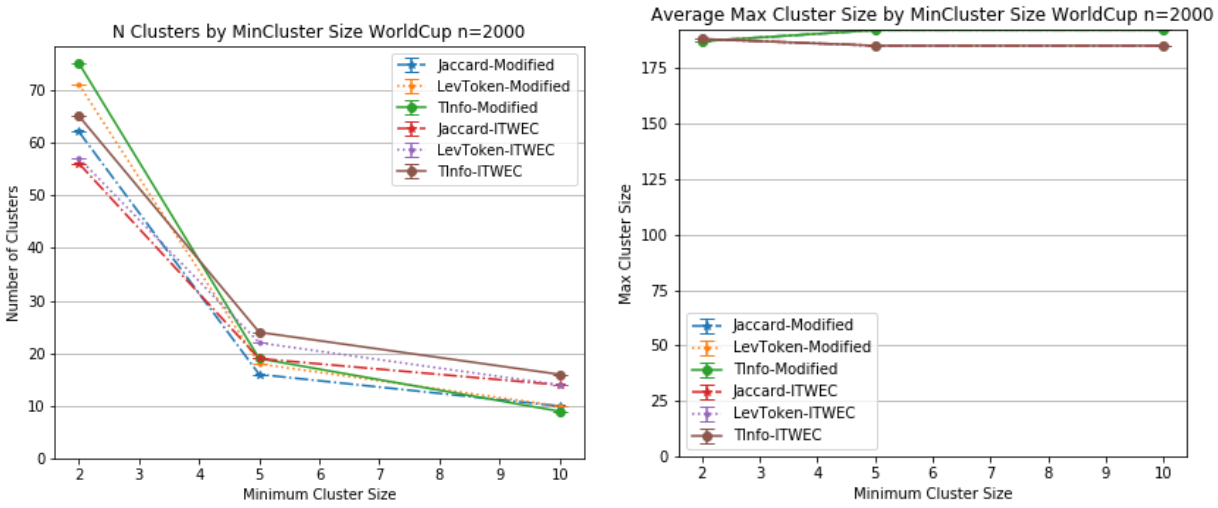

Figure 4.44: Cluster Characteristics by Minimum Cluster Size WorldCup

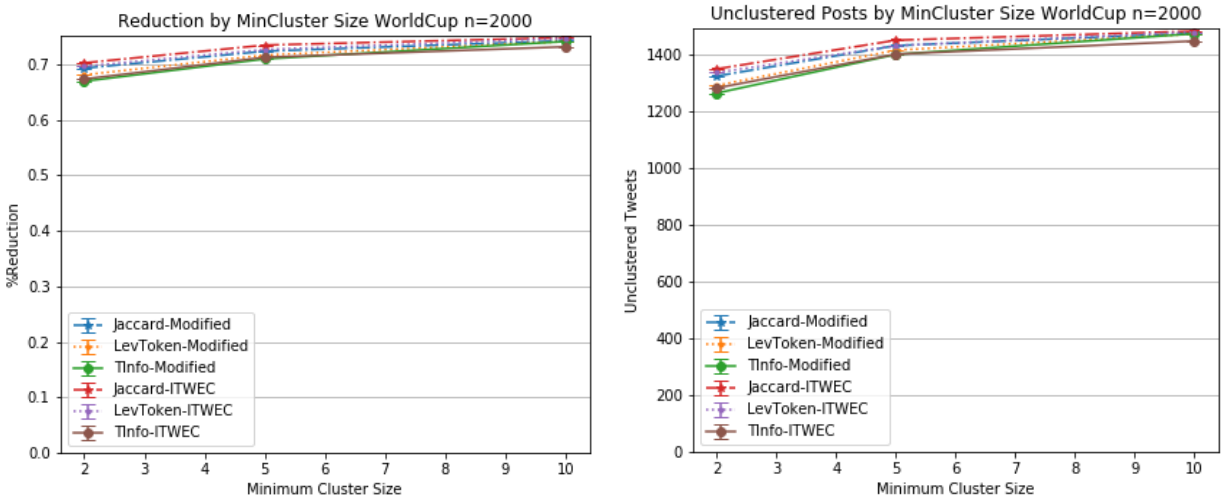

Figure 4.45: Reduction Characteristics by Minimum Cluster Size WorldCup

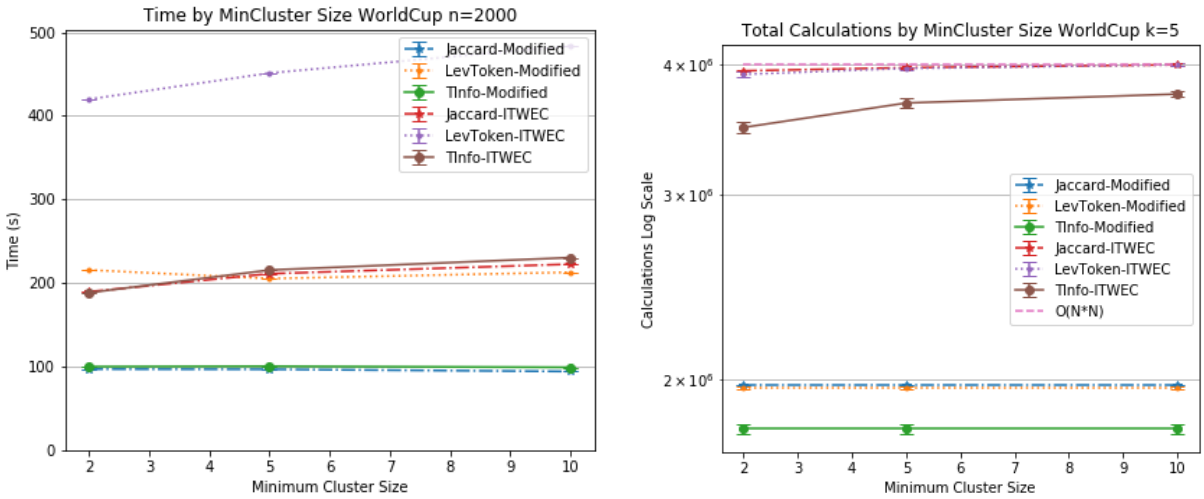

Figure 4.46: Complexity Characteristics by Minimum Cluster Size WorldCup

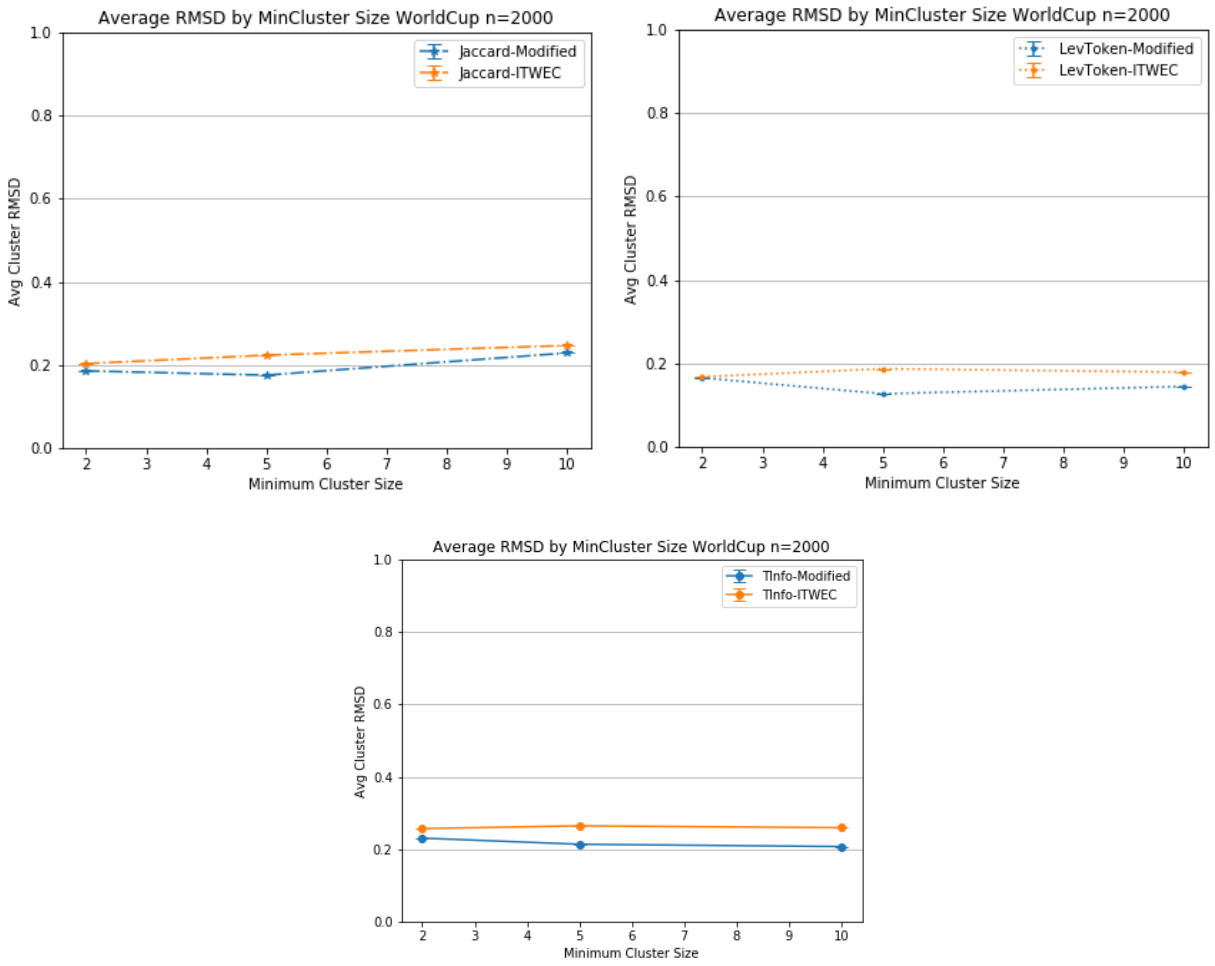

Figure 4.47: RMSD Characteristics by Minimum Cluster Size WorldCup

4.4 500-Tweet Search Clustering

An experiment was run to determine the effectiveness of each thresholding algorithm and similarity measure for an industry relevant simulated search. Specifically, 500 Tweet searches were simulated using samples from the original search data sets then clustered. For the purposes of this cluster simulation, the threshold was set to 0.4 and the minimum cluster value was set to 3 as discussed in the methodology. As it is not possible to normalize the distances across all similarity measures, each threshold represents a different value and may be more or less permissive for one measure than another. The test was run 50 times and analyzed for each of the analysis metrics including computational complexity, data reduction, cluster size measures, root mean square distance, and manual cluster validation. Table 4.2 to Table 4.37present the results of the simulation. Each table represents one characteristic belonging to one of three categories: cluster quality, reduction quality, and computational complexity.

4.4.1 Run Statistics for 500-Tweet Searches

Presented in this section are the results from running the 500 Tweet search simulation. Each table represents a run of 50 tests for the specified combination of search, similarity distance, and algorithm implementation. Included in the results are various measures across all runs and their respective statistics. Table 4.1 details what each measure represents.

Table 4.1: Analysis Metrics for 500 Tweet Simulation

Measure	Explanation
Number of Clusters	Total number of clusters found
Max Cluster Size	Largest cluster size
Number of Min Clusters	Number of clusters equal MinCluster parameter (3)
Average Cluster RMSD	Averaged Cluster RMSD for all clusters in the run
Reduction	Reduction from the original set
Unclustered Tweets	Number of Tweets not clustered
Mean Tweet Terms	Mean Tweet length per cluster by number of string tokens
Total Time (s)	Time in seconds
Total Calculations	Total calculations

4.4.1.1 Vancouver

Table 4.2 to Table 4.10 detail the results for the location-based search of Vancouver. Generally speaking, Vancouver is a redundant data set comprising a significant amount of career and job search related content. As a result, it clusters easily. As can be seen in the tables, the ITWEC clustering algorithm provides marginally better clustering for the Vancouver search. On average, ITWEC finds a few more clusters and maintains a similar average root means square distance. The cluster quality between ITWEC and the modified thresholding algorithm do not have significant performance differences. The T-Information, Jaccard, and Levenshtein similarity measures perform similarly for the Vancouver search. As can be seen by the mean Max Cluster Size in Table 4.3, each similarity measure finds similarly-sized dominant cluster for each search. The one notable difference for this experiment is that T-Information consistently finds more clusters than Levenshtein or Jaccard similarity distances. This improvement may be due

to the continuous nature of the T-Information measure and its ability to distinguish more nuanced similarities between Tweets.

Table 4.2: Number of Clusters for 500 Tweets Vancouver

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Number of Clusters	TInfo	18.0	28.0	23.8	2.72	21.0	32.0	27.4	2.86
	LevToken	11.0	25.0	17.2	2.68	12.0	25.0	18.3	2.92
	Jaccard	16.0	26.0	20.7	2.11	16.0	26.0	21.2	2.36

Table 4.3: Max Cluster Size for 500 Tweets Vancouver

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Max Cluster Size	TInfo	29	50	38.7	4.49	29	50	38.7	4.49
	LevToken	30	49	38.1	4.00	30	49	38.1	4.00
	Jaccard	31	49	38.6	3.72	31	49	38.6	3.72

Table 4.4: Number of Minimum Clusters for 500 Tweets Vancouver

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Number of Min Clusters	TInfo	3	12	6.62	2.27	3	16	8.56	2.81
	LevToken	1	13	5.62	2.78	2	14	6.4	2.84
	Jaccard	1	9	4.24	1.97	1	9	4.58	2.24

Table 4.5: Average RMSD for 500 Tweets Vancouver

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Average Cluster RMSD	TInfo	0.327	0.383	0.352	0.011	0.339	0.389	0.361	0.010
	LevToken	0.260	0.326	0.291	0.017	0.261	0.326	0.297	0.016
	Jaccard	0.239	0.295	0.271	0.012	0.245	0.295	0.274	0.012

The Vancouver results for reduction quality, presented in Table 4.6, show that the performance is also similar between the ITWEC and modified thresholding algorithms. The similarity across mean Tweet terms also suggests most clustered Tweets are of a significant length and not subject to hashtag or term bias. For similarity measures, it appears that Levenshtein string token distance clusters the most content, closely followed by T-Information. Jaccard performed the worst overall for content reduction.

Table 4.6: Reduction for 500 Tweets Vancouver

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Reduction	TInfo	0.466	0.582	0.515	0.022	0.442	0.542	0.493	0.021
	LevToken	0.554	0.644	0.603	0.022	0.542	0.640	0.596	0.021
	Jaccard	0.418	0.488	0.461	0.017	0.418	0.488	0.458	0.018

Table 4.7: Unclustered Tweets for 500 Tweets Vancouver

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Unclustered Tweets	TInfo	210	270	233.7	11.68	193	247	219.16	11.53
	LevToken	259	306	284.16	11.12	250	303	279.8	10.81
	Jaccard	186	228	209.74	9.25	186	228	207.64	9.94

Table 4.8: Mean Terms for 500 Tweets Vancouver

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Mean Tweet Terms	TInfo	15.90	17.66	16.67	0.34	15.86	17.57	16.56	0.32
	LevToken	15.97	17.20	16.70	0.27	15.92	17.20	16.68	0.26
	Jaccard	15.21	17.09	16.18	0.38	15.21	17.09	16.19	0.37

With respect to complexity, it can be seen in Table 4.9 and Table 4.10 the ITWEC algorithm exceeds the Modified thresholding algorithm for both time and calculations in all cases. As the algorithm have a different worst-case complexity, this is an expected result. T-Information performs the best in terms of overall time for all cases. Levenshtein string token similarity performs the least calculations, but at the expense of significantly longer time costs. This is likely a result of the Levenshtein similarity algorithm being more computationally complex than T-Information or Jaccard.

Table 4.9: Total Time for 500 Tweets Vancouver

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Total Time (s)	TInfo	3.53	4.78	4.22	0.28	6.78	10.02	8.20	0.70
	LevToken	4.31	5.99	5.07	0.34	8.87	12.48	10.50	0.68
	Jaccard	6.32	8.48	7.46	0.53	11.56	16.30	13.78	1.06

Table 4.10: Total Calculations for 500 Tweets Vancouver

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Total Calculations	TInfo	30078	41060	35759	2555	52854	78835	63995	5210
	LevToken	39583	51959	46210	3088	74591	102074	89637	6134
	Jaccard	24364	33730	29965	2054	45138	62496	53701	3939

4.4.1.2 London

The results for London are shown in Table 4.11 to Table 4.19. As compared to Vancouver, London is significantly less compressible and contains more unique data. As a result, each algorithm and similarity measure combination found few to no clusters in the data set. For London, ITWEC finds more clusters on average than the modified algorithm but both find the same dominant cluster. A low minimum Number of Min Clusters suggests it was common for the clustering algorithms to find a few clusters in the set that were larger than the minimum required cluster size, but no others. There appears to be no significant performance difference on cluster quality for the London based on similarity measure.

Table 4.11: Number of Clusters for 500 Tweets London

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Number of Clusters	TInfo	1.00	8	3.6	1.311	2.00	10.00	4.52	1.526
	LevToken	2.00	8.00	4.62	1.384	2.00	8.00	4.76	1.365
	Jaccard	0.00	5.00	2.66	1.124	1.00	6.00	2.82	1.108

Table 4.12: Max Cluster Size for 500 Tweets London

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Max Cluster Size	TInfo	3.00	11.00	5.26	1.83	3.00	11.00	5.28	1.81
	LevToken	3.00	12.00	6.26	1.80	3.00	12.00	6.26	1.80
	Jaccard	0.00	7.00	4.06	1.24	3	7	4.14	1.13

Table 4.13: Number of Min Clusters for 500 Tweets London

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Number of Min Clusters	TInfo	0	6	1.7	1.315	0	6	2.42	1.51
	LevToken	0	4	1.98	1.122	0	4	2.08	1.13
	Jaccard	0	4	1.56	1.003	0	4	1.72	0.98

Table 4.14: Average RMSD for 500 Tweets London

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Average Cluster RMSD	TInfo	0.242	0.376	0.316	0.027	0.242	0.397	0.333	0.027
	LevToken	0.195	0.333	0.267	0.029	0.195	0.333	0.270	0.028
	Jaccard	0.222	0.800	0.324	0.075	0.222	0.385	0.317	0.033

The London results for reduction quality, presented in Table 4.15, Table 4.16, and Table 4.17, show the performance is similar between the ITWEC and modified thresholding algorithms. The similarity across mean Tweet terms in Table 4.17 also suggests most clustered Tweets are of a significant length and not subject to hashtag or term bias. For the London search, it appears that Jaccard, Levenshtein, and T-Information distances perform similarly for the experiment's parameters as this set of Tweets is largely unclusterable.

Table 4.15: Reduction for 500 Tweets London

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Reduction	TInfo	0.958	0.992	0.979	0.008	0.950	0.992	0.974	0.009
	LevToken	0.950	0.992	0.970	0.010	0.950	0.992	0.970	0.010
	Jaccard	0.970	1.000	0.986	0.007	0.966	0.996	0.985	0.007

Table 4.16: Unclustered Tweets for 500 Tweets London

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Unclustered Tweets	TInfo	471	495	485.7	5.12	465	494	482.7	5.95
	LevToken	468	494	480.5	5.97	467	494	480.0	5.97
	Jaccard	480	500	490.4	4.33	477	497	489.9	4.35

Table 4.17: Mean Terms for 500 Tweets London

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Mean TweetTerms	TInfo	7.68	16.88	12.28	1.78	7.83	15.50	11.96	1.55
	LevToken	8.89	16.13	12.52	1.45	8.89	16.13	12.57	1.44
	Jaccard	0.00	17.21	12.59	2.56	9.00	17.33	13.10	1.89

For complexity, represented in Table 4.18 and Table 4.19, as the dataset was largely unclusterable, each algorithm performed close to its worst-case for sample size 500, which are 250,000 calculations for ITWEC and 124,750 calculations for the modified thresholding algorithm. For absolute time performance, T-Information and Jaccard performed similarity, while Levenshtein took significantly longer.

Table 4.18: Total Time for 500 Tweets London

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Total Time (s)	TInfo	12.83	14.40	13.43	0.33	26.60	29.74	27.92	0.59
	LevToken	23.02	26.45	24.47	0.78	47.20	54.09	50.16	1.61
	Jaccard	12.41	14.17	12.84	0.29	25.48	28.54	26.69	0.48

Table 4.19: Total Calculations for 500 Tweets London

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Total Calculations	TInfo	113859	121443	118397	1950	224695	245570	238019	4300
	LevToken	112546	121697	117075	2229	224435	245570	234753	4961
	Jaccard	117670	124044	120903	1384	232736	248202	242696	3409

4.4.1.3 Royal Wedding

RoyalWedding represents the least clusterable dataset at small samples and neither algorithm regularly found clusters for the experimental parameters. As can be seen in Tables 4.20 - 4.23 there are no significant differences in cluster characteristics between the ITWEC thresholding and the modified thresholding algorithm. There are, however, significant differences between the T-Information similarity measure and both Jaccard and Levenshtein measures. T-Information is the only similarity measure to consistently find clusters within the RoyalWedding dataset, as observed by the relatively high mean number of clusters. This is either because of T-Information's continuous nature or due to Tweet term length bias. Jaccard and Levenshtein string token distances cannot effectively compare two strings comprising a small number of tokens, whereas T-Information can. As can be seen in Table 4.22, the low number of mean Tweet terms suggests that T-Information is finding and clustering small Tweets where the other algorithms are failing. Upon investigation, it was found that the RoyalWedding has a number of small Tweets comprising a hashtag '#RoyalWedding' and a link similar to "t.co/<string>" or a few other characters. T-Information was able to find and cluster these Tweets while the other algorithms could not.

Table 4.20: Number of Clusters for 500 Tweets Royal Wedding

		Modified				**ITWEC**			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Number of Clusters	TInfo	1.0	7.0	2.74	1.412	1.0	9.0	3.58	1.638
	LevToken	0.0	1.0	0.26	0.439	0.0	1.0	0.26	0.439
	Jaccard	0.0	1.0	0.26	0.439	0.0	1.0	0.28	0.449

Table 4.21: Max Cluster Size for 500 Tweets Royal Wedding

		Modified				**ITWEC**			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Max Cluster Size	TInfo	3.0	17.0	9.3	3.318	3.0	17.0	9.52	3.195
	LevToken	0.0	5.0	0.86	1.483	0.0	5.0	0.86	1.483
	Jaccard	0.0	5.0	0.86	1.483	0.0	5.0	0.92	1.508

Table 4.22: Mean Terms for 500 Tweets Royal Wedding

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Mean Tweet Terms	TInfo	1.00	4.49	2.57	0.694	1.58	4.27	2.75	0.598
	LevToken	0.00	13.00	0.50	1.836	0.00	13.00	0.50	1.836
	Jaccard	0.00	13.00	0.50	1.836	0.00	13.00	0.55	1.859

Table 4.23: Average RMSD for 500 Tweets Royal Wedding

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Average Cluster RMSD	TInfo	0.000	0.449	0.365	0.067	0.267	0.469	0.385	0.041
	LevToken	0.000	0.800	0.594	0.348	0.000	0.800	0.594	0.348
	Jaccard	0.000	0.800	0.595	0.347	0.000	0.800	0.587	0.346

There are no significant differences between algorithms for the reduction in quality for RoyalWedding, as seen in Table 4.24 and Table 4.25. As previously discussed, T-Information out performs the other similarity measures due to its better accuracy for small Tweets.

Table 4.24: Reduction for 500 Tweets Royal Wedding

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Reduction	TInfo	0.96	1.00	0.97	0.010	0.95	0.99	0.97	0.010
	LevToken	0.99	1.00	1.00	0.002	0.99	1.00	1.00	0.002
	Jaccard	0.99	1.00	1.00	0.002	0.99	1.00	1.00	0.002

Table 4.25: Unclustered Tweets for 500 Tweets Royal Wedding

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Unclustered Tweets	TInfo	473	497	484.0	6.086	467	494	481.0	6.456
	LevToken	495	500	499.1	1.483	495	500	499.1	1.483
	Jaccard	495	500	499.1	1.483	495	500	499.1	1.508

Table 4.26: Mean Terms for 500 Tweets Royal Wedding

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Mean Tweet Terms	TInfo	1.00	4.49	2.57	0.694	1.58	4.27	2.75	0.598
	LevToken	0.00	13.00	0.50	1.836	0.00	13.00	0.50	1.836
	Jaccard	0.00	13.00	0.50	1.836	0.00	13.00	0.55	1.859

For time and computational complexity, as the dataset was largely unclusterable, each algorithm performed close to or at the worst case for sample size 500, which are 250,000 calculations for ITWEC and 124,750 calculations for the modified thresholding algorithm. For absolute time performance, T-Information and Jaccard performed similarity, while Levenshtein took significantly longer.

Table 4.27: Total Time for 500 Tweets Royal Wedding

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Total Time (s)	TInfo	11.91	14.76	12.66	0.460	24.54	28.90	26.43	0.987
	LevToken	20.55	24.04	22.78	0.706	43.13	49.74	46.75	1.335
	Jaccard	11.90	13.78	12.55	0.359	24.84	27.29	26.17	0.662

Table 4.28: Total Calculations for 500 Tweets Royal Wedding

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Total Calculations	TInfo	113777	122488	117511	2029	226710	245052	235473	4745
	LevToken	122515	124750	124171	540	245995	250000	249385	1100
	Jaccard	122515	124750	124302	500	245995	250000	249384	1100

4.4.1.4 WorldCup

Finally, the WorldCup search experiment was run and the results are tabulated in Table 4.29 to Table 4.37. As observed in Table 4.36 and Table 4.37, similar to the previous searches, T-Information and Jaccard similarity out perform the Levenshtein distance measure in time and the Modified thresholding algorithm outperforms the ITWEC clustering algorithm for both time and calculations.

As can be seen, the WorldCup cluster quality did not vary significantly across ITWEC or the modified algorithm. Further, there was little or no variance across the similarity measure for cluster quality. This is likely due to the redundant nature the content related to the WorldCup. Tweets are either unique and unclusterable or highly repetitive and very clusterable. In the case that they are clusterable, each algorithm finds them without trouble.

Table 4.29: Number of Clusters for 500 Tweets WorldCup

		Modified				**ITWEC**			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Number of Clusters	TInfo	5.0	13.0	9.46	1.69	5.0	13.0	9.66	1.75
	LevToken	5.0	13.0	9.38	1.65	5.0	13.0	9.38	1.65
	Jaccard	5.0	13.0	9.32	1.65	5.0	13.0	9.32	1.65

Table 4.30: Max Cluster Size for 500 Tweets WorldCup

		Modified				**ITWEC**			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Max Cluster Size	TInfo	6.0	13.0	8.44	1.85	6.0	13.0	8.44	1.85
	LevToken	6.0	13.0	8.44	1.85	6.0	13.0	8.44	1.85
	Jaccard	6.0	13.0	8.44	1.85	6.0	13.0	8.44	1.85

Table 4.31: Number of Min Clusters for 500 Tweets WorldCup

		Modified				**ITWEC**			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Number of Min Clusters	TInfo	0.0	7.0	2.62	1.35	0.0	8.0	2.78	1.45
	LevToken	0.0	7.0	2.62	1.41	0.0	7.0	2.62	1.41
	Jaccard	0.0	7.0	2.54	1.40	0.0	7.0	2.54	1.40

Table 4.32: Average RMSD for 500 Tweets WorldCup

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Average Cluster RMSD	TInfo	0.198	0.253	0.228	0.011	0.198	0.265	0.232	0.013
	LevToken	0.135	0.197	0.160	0.012	0.135	0.197	0.160	0.012
	Jaccard	0.226	0.285	0.260	0.014	0.226	0.285	0.260	0.014

As can be observed in Table 4.33, Table 4.34, and Table 4.35, the reduction quality characteristics for the WorldCup search vary insignificantly for all search algorithms and similarity distances. This supports the concept that there are a few very strong clusters within the WorldCup set that each is clustering.

Table 4.33: Reduction for 500 Tweets WorldCup

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Reduction	TInfo	0.882	0.964	0.925	0.017	0.882	0.964	0.924	0.017
	LevToken	0.888	0.964	0.926	0.016	0.888	0.964	0.926	0.016
	Jaccard	0.888	0.964	0.926	0.016	0.888	0.964	0.926	0.016

Table 4.34: Unclustered Tweets for 500 Tweets WorldCup

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Unclustered Tweets	TInfo	428	477	453.1	9.6	428	477	452.5	10.0
	LevToken	432	477	453.5	9.3	432	477	453.5	9.3
	Jaccard	432	477	453.6	9.2	432	477	453.6	9.2

Table 4.35: Mean Terms for 500 Tweets WorldCup

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Mean Tweet Terms	TInfo	18.05	21.34	20.34	0.82	17.32	21.34	20.07	1.09
	LevToken	18.20	21.34	20.54	0.62	18.20	21.34	20.54	0.62
	Jaccard	18.94	21.34	20.62	0.44	18.94	21.34	20.62	0.44

As observed in Table 4.36 and Table 4.37, similarly to the previous searches, T-Information and Jaccard similarity out perform the Levenshtein distance measure in time and the Modified thresholding algorithm outperforms the ITWEC clustering algorithm for both time and calculations.

Table 4.36: Total Time for 500 Tweets WorldCup

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Total Time (s)	TInfo	11.09	13.46	12.36	0.50	22.78	27.80	25.41	1.14
	LevToken	20.32	24.37	22.14	0.91	40.07	48.17	44.27	1.82
	Jaccard	10.60	13.39	11.58	0.50	21.69	25.90	23.87	1.02

Table 4.37: Total Calculations for 500 Tweets WorldCup

		Modified				ITWEC			
	Measure	Min	Max	Mean	STD	Min	Max	Mean	STD
Total Calculations	TInfo	101794	114995	108123	3384.67	194671	231707	213481	7946
	LevToken	102214	115118	108437	3296.10	196523	231707	213881	759
	Jaccard	102214	115118	108615	3289.14	196523	231707	213978	7576

4.4.2 Cluster Validation for 500 Tweet Searches

The following section presents a quantitative and qualitative cluster validation for a single 500 Tweet clustering run to verify the results of the previous section. The purpose for the validation is to assess if the clustering algorithms were responding according to expectations and to determine if the clusters were in fact redundant data. The results for each search used a 500 Tweet sample with threshold 0.4 and minimum cluster size of 3.

The quantitative validation includes three components, the distance measure of exemplar Tweets for the top five clusters found from the run, the distance between five randomly selected unclustered posts, and the distance between the aggregate content of each cluster. Exemplar Tweets were defined as the Tweet with the minimum root mean square distance to all other Tweets in the cluster. The validation also includes a word cloud of the largest cluster and a world could of the unclustered content.

For most searches, the five largest clusters were used for the qualitative comparison. However, in some cases, specifically London and RoyalWedding, five

clusters were not defined. In these cases, all clusters were used. In addition, the RoyalWedding search did not reliably return clusters for a 500 Tweet search, as such a 1000 Tweet search was used.

4.4.2.1 Vancouver Cluster Validation

The following section presents the quantitative cluster validation for the Vancouver dataset for T-Information, Jaccard, and Levenshtein similarity measures. Each section presents data for both ITWEC and the modified thresholding algorithms.

4.4.2.1.1 Vancouver T-Information Validation

Table 4.38 - Table 4.45 present the results of the validation tests for T-Information as applied in the Vancouver search. Table 4.38 and Table 4.39show the modified thresholding algorithms results for exemplar distances for the five largest clusters, respectively. As can be seen, the exemplar Tweet distances are relatively close to each other, with the largest being 0.718. Upon inspection of the Tweets, it is reaffirmed that the Vancouver data set is significantly careers focused and highly redundant.

Table 4.38: Modified T-Information Exemplar Distances for Vancouver

Modified			**Distance to:**				
Cluster No.	**Cluster Size**	**Exemplar RMSD**	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
0	43	0.277	0.009	0.536	0.564	0.667	0.461
1	40	0.239	0.536	0.000	0.646	0.600	0.587
2	31	0.257	0.551	0.633	-0.009	0.614	0.493
3	29	0.305	0.718	0.630	0.614	0.000	0.684
4	26	0.228	0.536	0.552	0.537	0.642	0.012

Table 4.39: Modified T-Information Exemplar Tweets for Vancouver

Cluster No.	Exemplar Tweet
0	Can you recommend anyone for this #job in #Vancouver, BC? https://t.co/jChGIfFynO #Retail #Hiring #CareerArc
1	Interested in a #job in #Vancouver, BC? This could be a great fit: https://t.co/m2KpqYyJlW #IT #Hiring #CareerArc
2	Want to work in #Vancouver, BC? View our latest opening: https://t.co/gFBLCxoxHc #QA #Job #Jobs #Hiring #CareerArc
3	Join the Mastercard team! See our latest #job opening here: https://t.co/IibbEk86pK #IT #Vancouver, BC #Hiring #CareerArc
4	If you're looking for work in #Vancouver, BC, check out this #job: https://t.co/NsZZjiSoJs #QA #Hiring #CareerArc

Similarly, Table 4.40 and Table 4.42 show the results for the exemplar distances for clusters formed using the ITWEC algorithm. In fact, the identified exemplar tweets are the same. The slight differences in values determined may be a result of T-Information's sensitivity to ordering.

Table 4.40: ITWEC T-Information Exemplar Distances for Vancouver

ITWEC			Distance to:				
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3	Ex4
0	43	0.277	0.009	0.536	0.564	0.667	0.461
1	40	0.239	0.536	0.000	0.646	0.600	0.587
2	31	0.257	0.551	0.633	-0.009	0.614	0.493
3	29	0.305	0.718	0.630	0.614	0.000	0.684
4	26	0.228	0.536	0.552	0.537	0.642	0.012

Table 4.41: ITWEC T-Information Exemplar Distances for Vancouver

Cluster No.	Exemplar Tweet
0	Can you recommend anyone for this #job in #Vancouver, BC? https://t.co/jChGIfFynO #Retail #Hiring #CareerArc
1	Interested in a #job in #Vancouver, BC? This could be a great fit: https://t.co/m2KpqYyJlW #IT #Hiring #CareerArc
2	Want to work in #Vancouver, BC? View our latest opening: https://t.co/gFBLCxoxHc #QA #Job #Jobs #Hiring #CareerArc
3	Join the Mastercard team! See our latest #job opening here: https://t.co/IibbEk86pK #IT #Vancouver, BC #Hiring #CareerArc
4	If you're looking for work in #Vancouver, BC, check out this #job: https://t.co/NsZZjiSoJs #QA #Hiring #CareerArc

Table 4.42 and Table 4.43 show results of comparing five randomly sampled unclustered Tweets to the exemplar Tweets. As can be seen, most Tweets are distant from the exemplar Tweets. However, values are not as high as might be seen from a less redundant dataset.

Table 4.42: Modified Unclustered Exemplar T-Information Distances Vancouver

Modified	Distance to:				
Unclusterd Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
Decided to do #happyHour at @JoeFortesVan with @iwa_realestate after our #TaiwanFest walk #amexWelcomed https://t.co/8UmoGFgxUk	0.779	0.837	0.820	0.781	0.869
ç¶šã„ã¦ãƒãƒ§ã‚³ãƒ¬ãƒ¼ãƒˆãƒ‡ã‚£ãƒƒãƒ—ã•«æƒ¹ã‹ã‚Œã¦Cacao70ã¸ã€‚ç¶šã—ã„Earl Grayã®ãƒ‡ã‚£ãƒ—ã‚’ãƒãƒ§ã‚¤ã‚¹ã€‚ãƒ›ãƒ¯ã‚¤ãƒˆãƒãƒ§ã‚³ãƒ™ãƒ¼ã‚¹ã§ç¾Žå‘³ã—ã‹ã£ãŸã€œï¼å‰ã®ãŠã˜ã•ã‚“ãŒãƒ¡ãƒ¼ãƒ—ƒ«ã•¨ãƒ›ãƒ¯ã‚¤ãƒˆã®WæŽ›ã‘é ¼ã‚“ã§ãŸã‹ã‚‰ã€ä»Šåº¦é ¼ã‚“ã§ã¿ãŸã„ðŸ˜Œ #RuSKæ—… https://t.co/clo0sS8p6F	1.06	1.07	1.03	1.04	1.06
Can you recommend anyone for this #job? Carpenters and Carpenter Apprentices - Vancouver, BC -â€¦ https://t.co/ZBap5rQxm5	0.502	0.818	0.773	0.832	0.696
This #job might be a great fit for you: Financial Services Representative Trainee (Chinese/Vietnamese Asset) -â€¦ https://t.co/BNEjpVGE1V	0.838	0.836	0.914	0.910	0.847
Consistency is the key to success at the gym. #fitfam #gym #motivationmonday #swoll #armsday #musclegay (@ Steve Naâ€¦ https://t.co/aImp0aTQFo	0.925	0.922	0.925	0.949	0.894

Table 4.43: ITWEC Unclustered Exemplar T-Information Distances Vancouver

ITWEC	**Distance to:**				
Unclusterd Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
UMMM CUTE!!!! What more do I have to say? Other than itâ€™s crazy cheap of course!! Did I mention today is the best dâ€¦ https://t.co/pxjqytbPHJ	0.892	0.890	0.945	0.888	0.936
We're #hiring! Click to apply: Brand Ambassadors for Leading Financial Institution in Vancouver -â€¦ https://t.co/ykf53nUS9L	0.818	0.855	0.818	0.841	0.847
UTC -7 | AUNZ â€” Dealing with dementia: Alzheimer's New Zealand provides support and education https://t.co/m5YAkRrY2H	0.976	0.912	0.935	0.897	0.944
Great to be back at BC Place! #HereWeCome #VWFC https://t.co/qXuCMUkx7u	0.884	0.840	0.857	0.831	0.853
This #job might be a great fit for you: Financial Services Representative Trainee (Chinese/Vietnamese Asset) - â€¦ https://t.co/BNEjpVGE1V	0.838	0.836	0.914	0.910	0.847

Table 4.44 and Table 4.45 list the aggregate cluster T-Information distances for the Vancouver validation search. As can be seen, the aggregate distances are greater than the exemplar distances for the same clusters, which is the expected results considering each cluster comprises additional information.

Table 4.44: Modified Aggregate Cluster T-Information Distances Vancouver

Modified	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.78	0.79	0.86	0.85
C1	0.78	0.00	0.83	0.89	0.83
C2	0.80	0.79	-0.01	0.87	0.82
C3	0.87	0.87	0.89	-0.01	0.88
C4	0.90	0.81	0.82	0.83	-0.01

Table 4.45: ITWEC Aggregate Cluster T-Information Distances Vancouver

ITWEC	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.78	0.79	0.86	0.85
C1	0.78	0.00	0.83	0.89	0.83
C2	0.80	0.79	-0.01	0.87	0.82
C3	0.87	0.87	0.89	-0.01	0.88
C4	0.90	0.81	0.82	0.83	-0.01

4.4.2.1.2 Vancouver Jaccard Validation

Table 4.46 to Table 4.53 present the results of the Jaccard validation tests for the Vancouver search. Table 4.46 to Table 4.49 show the modified and ITWEC thresholding algorithms results for exemplar distances and their respective exemplar Tweets. Similar to the T-Information results, the distances are relatively separate, but not very far apart. Further, the Modified and ITWEC exemplar tweets are almost the same, differing only in the fifth largest cluster.

Table 4.46: Modified Exemplar Tweet Jaccard Distances for Vancouver

Modified			**Distance to:**				
Cluster No.	**Cluster Size**	**Exemplar RMSD**	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
0	40	0.199	0.000	0.727	0.800	0.846	0.846
1	39	0.269	0.727	0.000	0.696	0.846	0.750
2	35	0.243	0.800	0.696	0.000	0.815	0.815
3	27	0.335	0.846	0.846	0.815	0.000	0.897
4	25	0.223	0.846	0.750	0.815	0.897	0.000

Table 4.47: Modified Jaccard Exemplar Tweets for Vancouver

Cluster No.	Exemplar Tweet
0	Interested in a #job in #Vancouver, BC? This could be a great fit: https://t.co/VmKVc4xRAL #Hiring #CareerArc
1	Can you recommend anyone for this #job in #Vancouver, BC? https://t.co/rJrtbePq1z #Hospitality #Hiring #CareerArc
2	Want to work in #Vancouver, BC? View our latest opening: https://t.co/rJrtbePq1z #Hospitality #Job #Jobs #Hiring #CareerArc
3	Join the Mastercard team! See our latest #job opening here: https://t.co/vXYvt4v9wF #Marketing #Vancouver, BC #Hiring #CareerArc
4	If you're looking for work in #Vancouver, BC, check out this #job: https://t.co/SvXYA6PaSP #Clerical #Hiring #CareerArc

Table 4.48: ITWEC Jaccard Exemplar Tweet Distances for Vancouver

ITWEC			Distance to:				
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3	Ex4
0	40	0.199	0.000	0.727	0.800	0.846	0.846
1	39	0.269	0.727	0.000	0.696	0.846	0.750
2	35	0.243	0.800	0.696	0.000	0.815	0.815
3	27	0.335	0.846	0.846	0.815	0.000	0.897
4	25	0.223	0.846	0.750	0.815	0.897	0.000

Table 4.49: ITWEC Jaccard Exemplar Tweets for Vancouver

Cluster No.	Exemplar Tweet
0	Interested in a #job in #Vancouver, BC? This could be a great fit: https://t.co/VmKVc4xRAL #Hiring #CareerArc
1	Can you recommend anyone for this #job in #Vancouver, BC? https://t.co/rJrtbePq1z #Hospitality #Hiring #CareerArc
2	Want to work in #Vancouver, BC? View our latest opening: https://t.co/rJrtbePq1z #Hospitality #Job #Jobs #Hiring #CareerArc
3	Join the Mastercard team! See our latest #job opening here: https://t.co/vXYvt4v9wF #Marketing #Vancouver, BC #Hiring #CareerArc
4	If you're looking for work in #Vancouver, BC, check out this #job: https://t.co/SvXYA6PaSP #Clerical #Hiring #CareerArc

Listed in Table 4.50 and Table 4.51 are the distances between the exemplar Tweets and unclustered posts using the Jaccard similarity measure for Vancouver. The majority

of the Tweets are visibly and quantitatively distance from the exemplars. The few Tweets that are closer resemble the careers style posts we expect from Vancouver.

Table 4.50: Modified Unclustered Exemplar Jaccard Distances Vancouver

Modified	**Distance to:**				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
I'm at @LionsPub in Vancouver, BC https://t.co/XHInHfEWkU https://t.co/vYan6wdcKb	0.95	0.95	0.96	0.96	0.96
More cute surprise #succulent plants for my desk! ðŸŒµðŸŒº I have a mini #garden in my office now.. ðŸŒ±â˜ºï¸ðŸŽðŸ©ðŸ»â€ðŸ’»âœ¨â€¦ https://t.co/xBM7pQ97KR	0.94	0.94	0.97	1.00	0.94
Drinking a London Calling Special Ale by Yaletown Brewing Company at @yaletownbrewing â€” https://t.co/eo9IuGGbCk	0.96	1.00	1.00	1.00	1.00
Important presentation by 2018 @CFDRTO Morgan Medal recipient @LisaBlundellRD at #dcconf18 on â€˜Exploring Coping Stâ€¦ https://t.co/qM3MdW1GiG	1.00	1.00	1.00	1.00	1.00
We're #hiring! Read about our latest #job opening here: Pharmaceutical Sales Representative - Vancouver / Calgary /â€¦ https://t.co/H2lntorSvR	0.97	0.97	0.94	0.83	1.00

Table 4.51: ITWEC Unclustered Exemplar Jaccard Distances Vancouver

ITWEC	**Distance to:**				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
Rally and March for #TinaFontaine #Vancouver March at 2:00pm starts at CBC Building #MMIW #Indigenous #Womenâ€¦ https://t.co/3zGLvQRs4o	1.00	0.96	1.00	1.00	0.97
We're #hiring! Read about our latest #job opening here: Carpenters and Carpenter Apprentices -â€¦ https://t.co/2hUKBjj31P	0.96	0.96	0.93	0.81	1.00
This #job might be a great fit for you: Regional Medical Advisor - Western Canada - https://t.co/frQmc3I05wâ€¦ https://t.co/psR0VvUBaS	0.81	0.93	1.00	0.97	0.97
UTC -7 | AUNZ â€” Dealing with dementia: Alzheimer's New Zealand provides support and education https://t.co/m5YAkRrY2H	1.00	1.00	1.00	1.00	1.00
Can you recommend anyone for this #job? DevOps Engineer - NuData Security - https://t.co/6uXr43k0Nr #OpenSourceâ€¦ https://t.co/6RXRePxkI7	1.00	0.74	1.00	1.00	0.93

In Table 4.52 and

Table 4.53 are the aggregate cluster distances for both the modified algorithm and the ITWEC algorithm, respectively. Similar to the previous results, the cluster distances are greater than those of the exemplar Tweets, but less than the unique Tweets.

Table 4.52: Modified Aggregate Cluster Jaccard Distances Vancouver

Modified	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.77	0.82	0.88	0.83
C1	0.77	0.00	0.78	0.88	0.83
C2	0.82	0.78	0.00	0.87	0.83
C3	0.88	0.88	0.87	0.00	0.86
C4	0.83	0.83	0.83	0.86	0.00

Table 4.53: ITWEC Aggregate Cluster Jaccard Distances Vancouver

ITWEC	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.77	0.82	0.88	0.83
C1	0.77	0.00	0.78	0.88	0.83
C2	0.82	0.78	0.00	0.87	0.83
C3	0.88	0.88	0.87	0.00	0.86
C4	0.83	0.83	0.83	0.86	0.00

4.4.2.1.3 Vancouver Levenshtein Validation

Presented in Table 4.54 to Table 4.61 are the results for the Levenshtein distance validation of the Vancouver search. From Table 4.54 to Table 4.57, it can be seen that similar to the previous two similarity measures, Levenshtein identifies the same exemplar tweets for both algorithm and each is relatively distance from the others.

Table 4.54: Modified Levenshtein Exemplar Tweet Distances for Vancouver

Modified			**Distance to:**				
Cluster No.	clusterSize	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3	Ex4
0	40	0.126	0.000	0.765	0.706	0.882	1.000
1	39	0.169	0.765	0.000	0.813	0.813	0.867
2	36	0.164	0.706	0.813	0.000	0.875	0.813
3	30	0.239	0.882	0.813	0.875	0.000	1.000
4	25	0.308	1.000	0.867	0.813	1.000	0.000

Table 4.55: Modified Levenshtein Exemplar Tweets for Vancouver

Cluster No.	**Content**
0	Interested in a #job in #Vancouver, BC? This could be a great fit: https://t.co/AAeBWezPks #Hospitality #Hiring #CareerArc
1	Can you recommend anyone for this #job in #Vancouver, BC? https://t.co/bvJQa509R7 #Hospitality #Hiring #CareerArc
2	Want to work in #Vancouver, BC? View our latest opening: https://t.co/rJrtbePq1z #Hospitality #Job #Jobs #Hiring #CareerArc
3	Join the Mastercard team! See our latest #job opening here: https://t.co/vXYvt4v9wF #Marketing #Vancouver, BC #Hiring #CareerArc
4	Want to work at Starbucks? We're #hiring in #Vancouver, BC! Click for details: https://t.co/flsoiAEEgzâ€¦ https://t.co/KL3uPA0FBW

Table 4.56: ITWEC Levenshtein Exemplar Tweet Distances for Vancouver

ITWEC			**Distance to:**				
Cluster No.	clusterSize	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3	Ex4
40	0.126	0.000	0.765	0.706	0.882	1.000	40
39	0.169	0.765	0.000	0.813	0.813	0.867	39
36	0.164	0.706	0.813	0.000	0.875	0.813	36
30	0.239	0.882	0.813	0.875	0.000	1.000	30
25	0.308	1.000	0.867	0.813	1.000	0.000	25

Table 4.57: ITWEC Levenshtein Exemplar Tweets for Vancouver

Cluster No.	Content
0	Interested in a #job in #Vancouver, BC? This could be a great fit: https://t.co/AAeBWezPks #Hospitality #Hiring #CareerArc
1	Can you recommend anyone for this #job in #Vancouver, BC? https://t.co/bvJQa509R7 #Hospitality #Hiring #CareerArc
2	Want to work in #Vancouver, BC? View our latest opening: https://t.co/rJrtbePq1z #Hospitality #Job #Jobs #Hiring #CareerArc
3	Join the Mastercard team! See our latest #job opening here: https://t.co/vXYvt4v9wF #Marketing #Vancouver, BC #Hiring #CareerArc
4	Want to work at Starbucks? We're #hiring in #Vancouver, BC! Click for details: https://t.co/flsoiAEEgzâ€¦ https://t.co/KL3uPA0FBW

Table 4.58 and Table 4.59 show Vancouver's unclustered Tweet distances to the exemplar Tweets for Levenshtein similarity. It was observed that some of the unclustered Tweets were maximally far from the exemplar Tweet as denoted by a distance of 1. As these do share some content and appear to be quite similar to some of the exemplar Tweets, it indicates a problem with the Levenshtein distance. As the Levenshtein edit distance is order dependent it appears that it cannot detect small similarities in a pair of Tweets that are constructed in a different format.

Table 4.58: Modified Unclustered Exemplar Levenshtein Distances Vancouver

Modified	Distance to:				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
UTC -7 | AUNZ â€" Meet the breakout Kiwi stars of 2018 https://t.co/oBqEmDGA3x	1.00	1.00	1.00	1.00	1.00
Attention college students, we've got #entrylevel #Recruiting #jobs! Apply today: https://t.co/ZBnMHCMSRsâ€¦ https://t.co/IJhRzCR3jo	1.00	1.00	1.00	1.00	1.00
I really like the â€œunconferenceâ€• format. You vote on the talks during breakfast using stickers you get with your baâ€¦ https://t.co/lO78FYWB2H	1.00	1.00	1.00	0.95	1.00
@ a concert and a girl tried to be OG using her lighter instead of her phone light & lit someoneâ€™s hair on fire	0.96	1.00	0.96	1.00	0.96
@Joshbal4 You are the embodiment of sin	1.00	1.00	1.00	1.00	1.00

Table 4.59: ITWEC Unclustered Exemplar Levenshtein Distances Vancouver

ITWEC	**Distance to:**				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
.@nico_mezquida curls in a beauty! 2-0 @WhitecapsFC!! #VWFC #VANvSJ #MLS https://t.co/ZLHPYeMEOx	0.94	1.00	1.00	0.94	0.94
Join the Robert Half Finance & Accounting team! See our latest #job opening here: https://t.co/I1Z0xuPRgsâ€¦ https://t.co/eIhb1JDoiO	1.00	1.00	0.65	1.00	1.00
Breakfast done right ðŸ˜• https://t.co/ZZxjsA3bUg	1.00	1.00	1.00	1.00	1.00
@Sonicray Are you fucking serious	1.00	1.00	1.00	1.00	1.00
Mike Bevilacqua explaining #homedialysis â€œentryâ€• and â€œexitâ€• points #CANASummit #CSN18 https://t.co/IjOrBkPxwR	1.00	1.00	1.00	1.00	1.00

Table 4.60 and Table 4.61 list the Levenshtein distances for the aggregate five largest clusters. As can be see, there are no differences between the two results sets.

Table 4.60: Modified Aggregate Cluster Levenshtein Distances Vancouver

Modified	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.82	0.87	0.78	0.82
C1	0.82	0.00	0.87	0.82	0.87
C2	0.87	0.87	0.00	0.92	0.90
C3	0.78	0.82	0.92	0.00	0.83
C4	0.82	0.87	0.90	0.83	0.00

Table 4.61: ITWEC Aggregate Cluster Levenshtein Distances Vancouver

I-TWEC	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.78	0.75	0.90	0.91
C1	0.78	0.00	0.87	0.93	0.91
C2	0.75	0.87	0.00	0.86	0.78
C3	0.90	0.93	0.86	0.00	0.95
C4	0.91	0.91	0.78	0.95	0.00

4.4.2.2 London Cluster Validation

Table 4.62 to Table 4.85 present the results for the London search validation for T-Information, Jaccard, and Levenshtein similarity distances. The modified and ITWEC clustering algorithms frequently did not define more than five clusters, as a result, all clusters are used for comparison.

4.4.2.2.1 London T-Information Validation

Table 4.62 - Table 4.65 show the results for the T-Information validation of the London search. Noted in Table 4.62 - Table 4.65 are the results for the exemplar Tweet distances and the exemplar Tweets. It can be seen that the exemplar Tweets and distances are the same for both ITWEC and the modified algorithm. The exemplar root mean square distances are different as a result of the Modified algorithm finding larger clusters and thereby containing more information.

Table 4.62: Modified T-Information Exemplar Distances for London

Modified			**Distance to:**				
Cluster No.	**Cluster Size**	**Exemplar RMSD**	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
0	9	0.229	0.000	0.753	0.431	0.815	1.020
1	5	0.356	0.784	0.000	0.771	0.931	1.060
2	4	0.204	0.467	0.819	0.000	0.859	0.931
3	4	0.347	0.831	0.911	0.900	0.000	0.993
4	3	0.369	1.032	1.004	1.076	1.047	0.011

Table 4.63: Modified T-Information Exemplar Tweets for London

Cluster No.	**Exemplar Tweets**
0	Trend Alert: 'Lopes'. More trends at https://t.co/do7Hdxwcnc #trndnl https://t.co/Upxuileo85
1	The tweet with the most impact of the #HUDMUN Trend, was published by @ManUtd: https://t.co/qcFxG5LskT (5794 RTs) #trndnl
2	'Morris' just started trending with 12688 tweets. More trends at https://t.co/do7Hdxwcnc #trndnl
3	I'm at @Selfridges & Co in London, Greater London, Greater London https://t.co/RUImJk4qlY

4	#Top3Apps for #myweekasamuslim Twitter for iPhone 57% Twitter for Android 26% Twitter Web Client 11%

Table 4.64: Modified T-Information Exemplar Distances for London

ITWEC			Distance to:				
Cluster No.	**Cluster Sizes**	**Exemplar RMSD**	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
0	9	0.229	0.000	0.753	0.431	0.815	1.020
1	5	0.356	0.784	0.000	0.771	0.931	1.060
2	4	0.204	0.467	0.819	0.000	0.859	0.931
3	4	0.347	0.831	0.911	0.900	0.000	0.993
4	3	0.369	1.032	1.004	1.076	1.047	0.011

Table 4.65: Modified T-Information Exemplar Tweets for London

Cluster No.	**Exemplar Tweets**
0	Trend Alert: 'Lopes'. More trends at https://t.co/do7Hdxwcnc #trndnl https://t.co/Upxuileo85
1	The tweet with the most impact of the #HUDMUN Trend, was published by @ManUtd: https://t.co/qcFxG5LskT (5794 RTs) #trndnl
2	'Morris' just started trending with 12688 tweets. More trends at https://t.co/do7Hdxwcnc #trndnl
3	I'm at @Selfridges & Co in London, Greater London, Greater London https://t.co/RUImJk4qlY
4	#Top3Apps for #myweekasamuslim Twitter for iPhone 57% Twitter for Android 26% Twitter Web Client 11%

Table 4.66 and Table 4.67 show results of comparing five randomly sampled unclustered Tweets to the exemplar Tweets for London. As expected, the unclustered Tweets are distant from the exemplars Tweets for both the modified algorithm and the ITWEC algorithm.

Table 4.66: Modified Unclustered Exemplar T-Information Distances London

Modified	**Distance to:**				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
PSA: Bigger smartphone apertures donâ€™t count if the sensors get smaller https://t.co/M5BbCAZdle https://t.co/ONort743x6	0.851	0.866	0.894	0.816	1.004
@dyakomard @gatusuchi @piotr408 @DilrubaLees @saravastiares @ViktorMochalin @hakim3220 @RcCamera @JenaC2â€¦ https://t.co/oD9B1naIaX	0.963	0.909	0.988	0.876	1.051
@TheGoonerette @ViewsWeekly @TCanton94 And Sadia your point is... are you affiliated per chance we are all entitledâ€¦ https://t.co/KPqKiUmuPZ	0.919	0.876	1.000	0.898	1.028
Drinking an 05|23 - India Pale Ale - Columbus Simcoe Summit by @BrewByNumbers at @brewbynumbers â€” https://t.co/7AuPap8Ri7	0.914	0.837	0.879	0.839	0.972
So who managed to bag Flight of Conchords tickets before they got to the touts?	1.112	0.967	1.006	0.925	1.065

Table 4.67: ITWEC Unclustered Exemplar T-Information Distances London

ITWEC	**Distance to:**				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
Question for #AllVoicesCount crowd - should 178 grants be seen as measure of success? Starting again would you do same, or fewer larger...?	1.03	1.02	1.08	1.03	1.06
Louis Theroux stays undefeated https://t.co/9xgRxuPE0u	0.870	0.871	0.934	0.843	1.11
These VERY long K-wires were removed from 3 of my toes todayðŸ˜·ðŸ¤¢ #4weeks post-op. Yes I admit I had to have gas and aâ€¦ https://t.co/AprA1vUgdI	0.929	0.916	0.953	0.933	1.08
Is your MP standing up for democracy and transparency by supporting the release of the #BrexitReports? Find out now: https://t.co/J2ROr98MG1	0.887	0.848	0.874	0.953	1.06
We are proud to have contributed to this work https://t.co/TCHQDQjziT	0.914	0.925	0.950	0.952	1.10

Table 4.68 and Table 4.69 show the aggregate cluster distance between the two clusters found in the London validation sample. As can be seen, the results are similar, but likely differ due to the modified clusters being larger.

Table 4.68: Modified Aggregate Cluster T-Information Distances

Modified	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.91	0.83	0.90	1.02
C1	0.90	0.01	0.89	0.89	1.06
C2	0.84	0.96	0.00	0.95	1.14
C3	0.88	1.02	0.91	0.01	1.09
C4	1.01	1.07	1.07	0.95	0.02

Table 4.69: ITWEC Aggregate Cluster Distances T-Information London

ITWEC	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.91	0.83	0.90	1.02
C1	0.90	0.01	0.89	0.89	1.06
C2	0.84	0.96	0.00	0.95	1.14
C3	0.88	1.02	0.91	0.01	1.09
C4	1.01	1.07	1.07	0.95	0.02

4.4.2.2.2 London Jaccard Validation

Tables 4.70 - 4.73 show London's Jaccard distance validation exemplar Tweet distance and exemplar Tweets for both the modified algorithm and the ITWEC algorithm. Similar to the T-Information validation result, the exemplar distances are effectively the same between the modified and ITWEC algorithm. The algorithms found different exemplar

Tweets. However, it is obvious the exemplar Tweets follow the same structure and would evaluate similarly in a string token comparison.

Table 4.70: Modified Jaccard Exemplar Distances for London

Modified			Distance to:			
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3
0	7	0.364	0.000	0.688	1.000	0.950
1	4	0.286	0.688	0.000	1.000	0.957
2	3	0.233	1.000	1.000	0.000	0.963
3	3	0.297	0.950	0.957	0.963	0.000

Table 4.71: Modified Jaccard Exemplar Tweets London

Cluster No.	Exemplar Tweets
0	Trend Alert: 'Lopes'. More trends at https://t.co/do7Hdxwcnc #trndnl https://t.co/Upxuileo85
1	'ATTACK' just started trending with 155435 tweets. More trends at https://t.co/do7Hdxwcnc #trndnl
2	If you're looking for work in #London, England, check out this #job: https://t.co/TL4PaTTXf1 #Marketing #Hiring #CareerArc
3	I'm at @Selfridges & Co in London, Greater London, Greater London w/ @raghadmohammed https://t.co/s0ofQXSD0K

Table 4.72: ITWEC Jaccard Exemplar Distances for London

ITWEC			Distance to:			
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3
0	7	0.364	0.000	0.688	1.000	0.950
1	4	0.286	0.688	0.000	1.000	0.957
2	3	0.233	1.000	1.000	0.000	0.963
3	3	0.297	0.950	0.957	0.963	0.000

Table 4.73: ITWEC Jaccard Exemplar Distances for London

Cluster No.	Exemplar Tweets
0	Trend Alert: 'Lopes'. More trends at https://t.co/do7Hdxwcnc #trndnl https://t.co/Upxuileo85
1	'ATTACK' just started trending with 155435 tweets. More trends at https://t.co/do7Hdxwcnc #trndnl
2	If you're looking for work in #London, England, check out this #job: https://t.co/TL4PaTTXf1 #Marketing #Hiring #CareerArc
3	I'm at @Selfridges & Co in London, Greater London, Greater London w/ @raghadmohammed https://t.co/s0ofQXSD0K

Table 4.74 and Table 4.75 list the unclustered Tweets and their respective Jaccard distances to the exemplar Tweets. Most of the tweets are visibly and quantitatively different, with the exception of the fourth modified algorithm example which is similar to the exemplar Tweet but greater than the evaluation threshold 0.4.

Table 4.74: Modified Unclustered Exemplar Jaccard Distances London

Modified	Distance to:			
Unclustered Tweets	**Ex0**	**Ex1**		
Got a virus which has knocked me for six. Managed to catch up on some admin, although probably full of errors? https://t.co/nutpCVTikP	1.00	1.00	0.97	1.00
I'm at @OxfordStreet in London, Greater London https://t.co/wE1HOa5Hfr	0.94	0.95	0.96	0.57
I'm at Dark Sphere in London, Greater London https://t.co/aU5RHDtjOa	0.94	0.95	0.96	0.60
Ù,Ø¯Ø±Øª Ù‡ÛŒÚ† Ù^Ù,Øª Ù¾Ø§ Ù†Ù…ÛŒ Ú©Ø´Ù‡ØŒ Ø¨Ø§ Ø§Ø§Ø®Ø§Ø° Ø´ÛŒÙ^Ù‡ Ù‡Ø§ÛŒ Ø§Ø¹Ù…Ø§„ Ù†Ø±Ù… Ø®Ù^Ø¯Ø´Ù^ Ø¯Ø± Ø¨Ø±Ø§Ø¨Ø± Ù‡Ø± Ø¬Ù^Ø± Ø§Ø¹Ø±Ø§Ø¶ Ø¨Ù†Ûŵ Ù^Ø§Ù‹Ù„„Ø§Ø¨ÛŒ Ù^Ø§Ú©Ø³ÛŒÙ†Ù‡ Ù…ÛŒ Ú©Ù†Ù‡	1.00	1.00	1.00	1.00
Drinking a Courage SE1 Green Hop Amber Ale by @southwarkbeer at @thebreelouise â€” https://t.co/Svx3KXdDbP	0.95	0.96	1.00	0.96

Table 4.75: ITWEC Unclustered Exemplar Jaccard Distances London

ITWEC	**Distance to:**			
Unclustered Tweets	Ex0	Ex1	Ex2	Ex3
@DavidAKrupp 2. most important- does it answer questions businesses need to answer easily and beautifully? you gotta bring both	1.00	1.00	1.00	1.00
Stigmabase | UK â€" UN Condemns Recent Spate Of Anti-Gay Mass Arrests https://t.co/TzPYtM8wle	1.00	1.00	1.00	1.00
Rich, chocolatey, and lingering... perfect complement to ... (Heavy Water with Hazelnut and Cocoa) https://t.co/a4FAoNDmcw	1.00	0.96	1.00	1.00
@GettyImagesNews @jackhipgrave	1.00	1.00	1.00	1.00
Join the CEB team! See our latest #job opening here: https://t.co/upqV0IdGiO #InsideSales #SalesLife #BusinessMgmt #London #Hiring	1.00	1.00	0.97	1.00

The two tables, Table 4.76 and Table 4.77, show the aggregate cluster distances for the London validation sample. As can be seen, the aggregate distances in the modified algorithm and the ITWEC algorithm case are the same as both clusters represent very similarly structured Tweets.

Table 4.76: Modified Aggregate Cluster Jaccard Distances London

Modified	**Distance to:**			
Aggregate Cluster	C0	C1	C2	C3
C0	0.00	0.90		
C1	0.90	0.00		
C2				
C3				

Table 4.77: ITWEC Aggregate Cluster Jaccard Distances London

ITWEC	**Distance to:**			
Aggregate Cluster	C0	C1	C2	C3
C0	0.00	0.90		
C1	0.90	0.00		
C2				
C3				

4.4.2.2.3 London Levenshtein

Table 4.78 - Table 4.85 display the results from the Levenshtein distance validation sample of London. As can be seen in the Table 4.78 and Table 4.80, which represent the exemplar distances for the modified algorithm and the ITWEC algorithm.

Table 4.78: Modified Levenshtein Exemplar Tweet Distances for London

Modified			**Distance to:**				
Cluster No.	**Cluster Size**	**Exemplar RMSD**	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
0	13	0.258	0.000	0.667	1.000	1.000	1.000
1	8	0.224	0.667	0.000	1.000	1.000	1.000
2	5	0.307	1.000	1.000	0.000	0.941	0.923
3	3	0.279	1.000	1.000	0.941	0.000	0.882
4	3	0.218	1.000	1.000	0.923	0.882	0.000

Table 4.79: Modified Levenshtein Exemplar Tweet for London

Cluster No.	**Exemplar Tweets**
0	Trend Alert: #USWNT. More trends at https://t.co/do7Hdxwcnc #trndnl https://t.co/BMgTXfOzYA
1	'Valencia' just started trending with 75136 tweets. More trends at https://t.co/do7Hdxwcnc #trndnl
2	I'm at @TheBookClubEc2 in London, Greater London https://t.co/4HXvtKX8vH
3	Interested in a #job in #London, England? This could be a great fit: https://t.co/cMsWzrUHQo #WebDesign #Veterans #Hiring
4	Can you recommend anyone for this #job in #London, England? https://t.co/jgL6qKbJI1 #Marketing #Hiring

Table 4.80: ITWEC Levenshtein Exemplar Tweet Distances for London

ITWEC			Distance to:		
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2
0	12	0.261	0.000	0.667	1.000
1	7	0.232	0.667	0.000	1.000
2	4	0.324	1.000	1.000	0.000

Table 4.81: ITWEC Levenshtein Exemplar Tweet Distances for London

Cluster No.	Exemplar Tweets
0	Trend Alert: 'Smalling'. More trends at https://t.co/do7Hdxwcnc #trndnl https://t.co/GMtPkNqXiG
1	'Carson' just started trending with 12814 tweets. More trends at https://t.co/do7Hdxwcnc #trndnl
2	I'm at @FutureGov in London, Greater London https://t.co/DyWcOX1VgP

Table 4.82: Modified Unclustered Exemplar Levenshtein Distances London

Modified	Distance to:				
Unclustered Tweets	Ex0	Ex1	Ex2	Ex3	Ex4
@funesdamemorius That's amazingly beautiful. What's the context?	1.00	1.00	1.00	1.00	1.00
â’• Toro Rosso â’Ž #payinterns â’• Ulysses â’• #SustainableSoilsNow â’‘ Labour MP 2017/10/23 17:53 BST #trndnl https://t.co/do7Hdxwcnc	0.94	0.94	1.00	1.00	1.00
@NormieGardener @Math_oma @SorosAccountant @AtheistApeMan @YouTube Rubbish. Pretending they are personification ofâ€¦ https://t.co/9hKqyluRRk	1.00	1.00	1.00	1.00	1.00
Ah Drunk Richard, ever the centre of attention https://t.co/BUOvq5w51F	1.00	1.00	1.00	1.00	1.00
@sarahbowen74 @albsar1970 One of the most shocking things in social media is how many people don't know the differeâ€¦ https://t.co/sFeVemOHIc	1.00	1.00	0.95	1.00	0.95

Table 4.83: Modified Unclustered Exemplar Levenshtein Distances London

ITWEC	**Distance to:**		
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**
wait what where you goin https://t.co/dylWyNwvNk	1.00	1.00	1.00
#WindowsInsiders: this has been the most fun release for us, no? Ready to share with your families/friends? https://t.co/FBLn6AitCY	1.00	1.00	1.00
Good routine. ðŸ›€ https://t.co/eDHZLibaMQ	1.00	1.00	1.00
Recap | UK â€" California enacts protections for LGBTI seniors in long-term care facilities -Â Trump administrationâ€¦ https://t.co/l45R5g6SGB	1.00	1.00	0.94
We are pleased to announce #Braintree's Christmas events over the festive season! Check out details here >â€¦ https://t.co/JEle8Aqkky	1.00	1.00	1.00

Table 4.84: Modified Aggregate Cluster Levenshtein Distances London

Modified	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.73	0.96	1.00	1.00
C1	0.73	0.00	0.95	1.00	1.00
C2	0.96	0.95	0.00	0.94	0.98
C3	1.00	1.00	0.94	0.00	0.86
C4	1.00	1.00	0.98	0.86	0.00

Table 4.85: ITWEC Aggregate Cluster Levenshtein Distances London

ITWEC	**Distance to:**		
Aggregate Cluster	C0	C1	C2
C0	0.00	0.72	0.96
C1	0.72	0.00	0.96
C2	0.96	0.96	0.00

4.4.2.3 Royal Wedding Cluster Validation

The RoyalWedding cluster validation could not be reliably run on 500 Tweet samples as the thresholding algorithms would regularly find no data. As a result, the RoyalWedding validation was run on a sample of 1000 posts to guarantee a meaningful result.

4.4.2.3.1 RoyalWedding T-Information

Table 4.86 - Table 4.93 show the data for the T-Information Distance validation of the RoyalWedding search. As can bee seen in Table 4.86 and Table 4.88, T-Information found similar exemplar Tweets for both algorithms. Perhaps most interestingly, T-Information found the one largest cluster to be the format <#RoyalWedding *link*> which was suspected to be the reason it performed better at clustering than Jaccard and Levenshtein for the RoyalWedding search. Jaccard and Levenshtein string token measures are not capable of clustering format of Tweet for a threshold of 0.4. Also worthy of note is the low RMSD of exemplar Tweet for cluster 2.This indicates that the cluster is very similar tweets likely with only the link changing.

Table 4.86: Modified T-Information Exemplar Distances for RoyalWedding

Modified			**Distance to:**			
Cluster No.	clusterSize	tweetsRMSD	Ex0	Ex1	Ex2	Ex3
0	12	0.275	0.000	0.759	0.813	0.613
1	12	0.274	0.733	0.000	0.935	0.693
2	5	0.123	0.755	0.871	0.017	0.856
3	3	0.370	0.777	0.693	0.956	0.000

Table 4.87: Modified T-Information Exemplar Tweets for RoyalWedding

Cluster No.	**Content**
0	#RoyalWedding https://t.co/PCtG3RzgCM
1	#RoyalWedding
2	#WelshCPC18 #RoyalWedding #PprinceCharles #Taylorswift #JamieMurphy #Manchester #Liverpool #London #Bradfordâ€¦ https://t.co/wTzb7cVeYs
3	honestly so excited to watch the #RoyalWedding

Table 4.88: ITWEC T-Information Exemplar Distances for RoyalWedding

ITWEC			**Distance to:**			
Cluster No.	clusterSize	tweetsRMSD	Ex0	Ex1	Ex2	Ex3
0	11	0.276	0.000	0.759	0.739	0.445
1	11	0.260	0.733	0.000	0.900	0.844
2	4	0.129	0.791	0.909	0.000	0.800
3	3	0.309	0.425	0.808	0.811	0.018

Table 4.89: ITWEC T-Information Exemplar Tweets for RoyalWedding

Cluster No.	**Content**
0	#RoyalWedding https://t.co/PCtG3RzgCM
1	#RoyalWedding
2	#WelshCPC18 #RoyalWedding #PrinceCharles #Taylorswift #JamieMurphy #Manchester #Liverpool #London #Bradfordâ€¦ https://t.co/i7G8S9Fro0
3	#royalwedding https://t.co/7lcEN17IVV

Table 4.90 and Table 4.91 show the T-Information distances from a selection of unclustered posts. As can be seen, all the unclustered posts are reasonably distant from the exemplars.

Table 4.90: Modified Unclustered Exemplar T-Information Distances RoyalWedding

Modified	**Distance to:**			
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**
The latest The Daily Morning Walk Fresh! https://t.co/iZPAlZj0sn #royalwedding #riverdale	0.787	0.970	0.902	0.953
AI will be used to identify celebs at the Royal Wedding. https://t.co/uVEdywIcpf #RoyalWedding #AI #Software #Tech https://t.co/6K0u4PPYtC	0.783	0.928	0.853	0.831
Up at 3:00am to watch THE wedding!!! Who's with me??? #RoyalWedding	0.784	0.829	0.977	0.679
There are two types of people: Those getting up at the crack of dawn tomorrow to watch the #RoyalWedding and thoseâ€¦ https://t.co/iFmJUNsL33	0.836	0.884	0.864	0.790
The latest The CCsexyshop Daily! https://t.co/hVJxjZcm98 #royalwedding #asmsg	0.720	0.879	0.882	0.809

Table 4.91: ITWEC Unclustered Exemplar T-Information Distances RoyalWedding

ITWEC	**Distance to:**			
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**
what must it feel like ðŸ–¤ðŸ˜• #RoyalWedding	0.820	0.755	0.921	0.889
Iâ€™m really looking forward to the #royalwedding. Being over.	1.007	0.959	0.912	0.751
Eeeeeee! Itâ€™s tomorrow!! #RoyalWedding I keep thinking about the first Royal Wedding I ever watched, which was Soâ€¦ https://t.co/JQqGe6ePzO	0.808	0.987	0.891	0.862
This is what you need to know the day before the #RoyalWedding ðŸ’• https://t.co/jF5wXPm1kF https://t.co/7nNFPMwcY6	0.728	0.906	0.823	0.760
@radioleary surely everyone whoâ€™s been invited to the #RoyalWedding is notable and quotable to Harry and Meghan? Not just the famousâ€™??	0.866	0.838	0.937	0.957

Table 4.92 and Table 4.93 so the aggregate cluster T-Information distance between each cluster found for RoyalWedding. Each cluster appears to be significantly distance from the others in the set. Considering the similarity of C0 and C1 exemplar Tweets, this indicates that T-Information strongly considers the variable data in the short links as unique and therefor the aggregate is distant from the simple hashtag cluster.

Table 4.92: Modified Aggregate Cluster T-Information Distances Royal Wedding

Modified	**Distance to:**			
Aggregate Cluster	C0	C1	C2	C3
C0	0.02	1.0	0.89	1.0
C1	1.0	0.05	0.95	0.84
C2	0.87	1.0	0.01	1.03
C3	1.0	0.82	0.93	0.00

Table 4.93: ITWEC Aggregate Cluster T-Information Distances RoyalWedding

ITWEC	Distance to:			
Aggregate Cluster	C0	C1	C2	C3
C0	0.03	1.00	0.84	0.87
C1	1.00	0.06	0.92	1.00
C2	0.86	1.00	0.01	0.87
C3	0.88	1.00	0.89	0.01

4.4.2.3.2 RoyalWedding Jaccard

Table 4.94 to Table 4.101 show the validation results for the Jaccard distance measure on the RoyalWedding data set. As compared to T-Information, Jaccard finds fewer clusters for the same dataset. As seen in Tables 4.94 - 4.97 both ITWEC and the modified algorithm find similar exemplar Tweets and distances. Notably, the exemplar RMSD for cluster 0 is 0.0. This indicates that the cluster is, in fact, a unique redundant post. The second clusters exemplar RMSD is also small, indicating that the format of the Tweets in the cluster is the same and only the link changes.

Table 4.94: Modified Jaccard Exemplar Distances for Royal Wedding

Modified			Distance to:	
Cluster No.	**Cluster Size**	**Tweet RMSD**	**Ex0**	**Ex1**
0	8	0.000	0.000	0.900
1	5	0.182	0.900	0.000

Table 4.95: Modified Jaccard Exemplar Tweets for RoyalWedding

Cluster No.	Exemplar Tweets
0	#RoyalWedding
1	#WelshCPC18 #RoyalWedding #PrinceCharles #Taylorswift #JamieMurphy #Manchester #Liverpool #London #Bradfordâ€¦ https://t.co/1t9dB0PhYg

Table 4.96: Modified Jaccard Exemplar Distances for RoyalWedding

ITWEC			Distance to:	
Cluster No.	Cluster Size	Tweet RMSD	Ex0	Ex1
0	8	0.000	0.000	0.900
1	5	0.182	0.900	0.000

Table 4.97: ITWEC Jaccard Exemplar Tweets for RoyalWedding

Cluster No.	Exemplar Tweets
0	#RoyalWedding
1	#WelshCPC18 #RoyalWedding #PrinceCharles #Taylorswift #JamieMurphy #Manchester #Liverpool #London #Bradfordâ€¦ https://t.co/i7G8S9Fro0

Table 4.98 and Table 4.99 show the Jaccard distances of a sample of unclustered posts to the exemplar Tweets. It can be seen each unclustered post is distant to the exemplars and well above the decision threshold 0.4.

Table 4.98: Modified Unclustered Exemplar Jaccard Distances London

Modified	Distance to:	
Unclustered Tweets	Ex0	Ex1
It's #RoyalWedding time! https://t.co/vcQlyyGsVv	0.750	0.923
Before you watch the #RoyalWedding look into SERCO	0.875	0.941
Former royal butler says Princess Diana will be at #RoyalWedding in spirit https://t.co/W751D6QaRh	0.923	0.955
18 de Mayo del 2018: 1) #RoyalWedding 2) Tiroteo en Texas Curioso que los tiroteos coincidan con eventos de gran importancia.	0.950	0.966
@xkom_pl #porannazmiana chyba wymiÄ™ka wobec #RoyalWedding ðŸ¤”	0.857	0.938

Table 4.99: ITWEC Unclustered Exemplar Jaccard Distances RoyalWedding

ITWEC	**Distance to:**	
Unclustered Tweets	**Ex0**	**Ex1**
Si el sÃ¡bado pasado fue EurovisiÃ³n hoy le toca a twitter petarla con #RoyalWedding Me gustarÃa saber todos aquelloâ€¦ https://t.co/s85nJLLwXe	0.950	0.966
Ada janji khusus di #RoyalWedding Pangeran Harry dan Meghan Markle. #Liputan6SCTV https://t.co/xy2R6dG5PE	0.917	0.952
Wow a hashtag for you ate @aini_almond #RoyalWedding this is not a coincidence ðŸ˜ŽðŸ'«	0.923	0.955
Looking forward to the #RoyalWedding tomorrow! ðŸŽ©ðŸ'°ðŸ•» My girls have got their tiaras ready! ðŸ˜Š #bbcone https://t.co/V53y4wvB45	0.941	0.962
8 totally incredible celebrity wedding looks our Fashion team love: https://t.co/IVmilcsz5b #royalwedding https://t.co/8OayEgoFXR	1.000	1.000

Finally, Table 4.100 and Table 4.101 show the aggerate Jaccard cluster distances for the RoyalWedding search. It can be seen they are significantly far apart for this sample.

Table 4.100: Modified Aggregate Cluster Jaccard Distances RoyalWedding

Modified	**Distance to:**	
Aggregate Cluster	C0	C1
C0	0.00	0.93
C1	0.93	0.00

Table 4.101: ITWEC Aggregate Cluster Jaccard Distances RoyalWedding

ITWEC	**Distance to:**	
Aggregate Cluster	C0	C1
C0	0.00	0.92
C1	0.92	0.00

4.4.2.3.3 RoyalWedding Levenshtein

Table 4.102 to Table 4.109 present the results for the Levenshtein distance validation of the RoyalWedding search. As can be seen from the exemplar Tweets and their distances, in Table 4.102 - Table 4.105, the results are almost exactly similar to the Jaccard results in the previous section. The exemplar Tweet RMSD is either very small or zero, indicating the Tweets in the cluster are exactly redundant or of the same format.

Table 4.102: Modified Levenshtein Exemplar Distances for RoyalWedding

Modified			**Distance to:**	
Cluster No.	**Cluster Size**	**Exemplar RMSD**	**Ex0**	**Ex1**
0	8	0	0	0.9
1	5	0.1	0.9	0

Table 4.103: Modified Levenshtein Exemplar Tweets for RoyalWedding

Cluster No.	**Exemplar Tweets**
0	#RoyalWedding
1	#WelshCPC18 #RoyalWedding #PrinceCharles #Taylorswift #JamieMurphy #Manchester #Liverpool #London #Bradfordâ€¦ https://t.co/1t9dB0PhYg

Table 4.104: ITWEC Levenshtein Exemplar Distances for RoyalWedding

ITWEC			**Distance to:**	
Cluster No.	**Cluster Size**	**Exemplar RMSD**	**Ex0**	**Ex1**
0	7	0	0	0.9
1	4	0.1	0.9	0

Table 4.105: ITWEC Levenshtein Exemplar Tweets for RoyalWedding

Cluster No.	Exemplar Tweets
0	#RoyalWedding
1	#WelshCPC18 #RoyalWedding #PrinceCharles #Taylorswift #JamieMurphy #Manchester #Liverpool #London #Bradfordâ€¦ https://t.co/i7G8S9Fro0

Table 4.106 and Table 4.107 show the Levenshtein distances of a sample of unclustered posts to the exemplar Tweets. Interestingly, some of the unique Tweets are maximally distant from the exemplar Tweets. As some of these Tweets do share some small similarities, it shows that Levenshtein distances is not an intelligent edit distance and may not detect small amounts of similarity.

Table 4.106: Modified Unclustered Exemplar Levenshtein Distances RoyalWedding

Modified	Distance to:	
Content	Ex0	Ex1
Idris Elba ðŸ˜•â•¤ #RoyalWedding	0.750	1.000
It's the #RoyalWedding tomorrow and we know that the royalists amongst you are going to want to keep up to date witâ€¦ https://t.co/iixwEbdnQ5	0.957	0.957
@Niederegger_UK My partner Paul but I'm not sure he's even heard of Twitter! #win #RoyalWedding	0.933	1.000
Allison Hamilton from Anglesey has bagged a prime spot ahead of #RoyalWedding @BBCRadioWales @BBCWalesNews https://t.co/JFetpbfp2s	0.933	1.000
If you donâ€™t want to watch racist POS @megynkelly ruin the #royalwedding turn on @MSNBC with @JoyAnnReid anchoringâ€¦ https://t.co/UGPR8zwW1U	1.000	1.000

Table 4.107: ITWEC Unclustered Exemplar Levenshtein Distances RoyalWedding

ITWEC	Distance to:	
Content	Ex0	Ex1
Is it wrong that when I saw #Slayer trending, I thought they'd been booked for the #RoyalWedding?	1.000	1.000
You can watch it too #RoyalWedding https://t.co/xFk9VCyyNL	0.857	1.000
Itâ€™s the Great British Face-Off. Find out what role facial recognition technology has to play in the #royalweddingâ€¦ https://t.co/z2sunHABpE	1.000	1.000
There are two types of people: Those getting up at the crack of dawn tomorrow to watch the #RoyalWedding and thoseâ€¦ https://t.co/iFmJUNsL33	0.955	1.000
I get the cynicism around the #RoyalWedding I do. But in a world full of so much shite at times can't we just wallâ€¦ https://t.co/7112oXtVSd	0.960	0.960

Table 4.108 and Table 4.109 show the aggregate cluster distances for both the modified and ITWEC algorithm clusters. As expected, they are similar, and each cluster is distant from the other.

Table 4.108: Modified Unclustered Exemplar Levenshtein Distances RoyalWedding

Modified	**Distance to:**	
Aggregate Cluster	C0	C1
C0	0	0.9
C1	0.9	0

Table 4.109: ITWEC Unclustered Exemplar Levenshtein Distances RoyalWedding

ITWEC	**Distance to:**	
Aggregate Cluster	C0	C1
C0	0	0.9
C1	0.9	0

4.4.2.4 WorldCup Cluster Validation

The following sections to present the results for the WorldCup search validation for T-Information, Jaccard, and Levenshtein similarity distances.

4.4.2.4.1 World Cup T-Information

Table 4.110 - Table 4.117 show the results for the T-Information validation of the WorldCup search. Table 4.110 to Table 4.113 are the exemplar Tweet distances and the exemplar Tweets. The exemplar Tweets are the same or a similar format for both ITWEC and the modified algorithm. The relatively small exemplar RMSDs indicated that the clusters are very tight and highly redundant.

Table 4.110: Modified T-Information Exemplar Distances for WorldCup

Modified			Distance to:				
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3	Ex4
0	51	0.176	0.000	0.639	0.745	0.803	0.742
1	32	0.202	0.795	-0.016	0.830	0.789	0.780
2	16	0.176	0.760	0.773	0.015	0.781	0.838
3	11	0.248	0.724	0.765	0.764	0.015	0.752
4	6	0.253	0.833	0.780	0.859	0.833	0.000

Table 4.111: Modified T-Information Exemplar Distances for WorldCup

Cluster No.	Exemplar Tweet
0	@TuneMakaveli Thanks for subscribing! Up next on your World of Champions is the 2018 FIFA #WorldCup âš½ï¸• Connect withâ€¦ https://t.co/Xbc7yZVgnD
1	@ethanthegoat Score! FIFA #WorldCup reminders are coming your way! Download the FOX Sports App now and never miss aâ€¦ https://t.co/ZuUQMFuEec
2	888 #WorldCup 2018 Winner (Outright) ðŸ† 40/1 #Belgium ðŸ‡§ðŸ‡ª to win the World Cup â€“ MAX BET Â£5 Promo-code: BOOSTâ€¦ https://t.co/jzwI1rGTGM
3	BETFAIR 200/1 Messi to be Top Goalscorer at the #WorldCup, New Customers,Max Stake Â£1. Winnings paid in free bets.â€¦ https://t.co/JcRi25xJkF
4	My #WorldCup Winner is Spain. Predict Yours! https://t.co/Ee7raj0k7h #FIFA18

Table 4.112: ITWEC T-Information Exemplar Distances for WorldCup

ITWEC			Distance to:				
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3	Ex4
0	50	0.176	0.000	0.639	0.745	0.803	0.689
1	31	0.201	0.795	-0.016	0.830	0.789	0.752
2	15	0.175	0.760	0.773	0.015	0.781	0.789
3	10	0.253	0.724	0.765	0.764	0.015	0.720
4	5	0.249	0.825	0.804	0.882	0.807	0.000

Table 4.113: ITWEC T-Information Exemplar Distances for WorldCup

Cluster No.	Exemplar Tweet
0	@TuneMakaveli Thanks for subscribing! Up next on your World of Champions is the 2018 FIFA #WorldCup âš½ï¸• Connect withâ€¦ https://t.co/Xbc7yZVgnD
1	@ethanthegoat Score! FIFA #WorldCup reminders are coming your way! Download the FOX Sports App now and never miss aâ€¦ https://t.co/ZuUQMFuEec
2	888 #WorldCup 2018 Winner (Outright) ðŸ† 40/1 #Belgium ðŸ‡§ðŸ‡ª to win the World Cup â€“ MAX BET Â£5 Promo-code: BOOSTâ€¦ https://t.co/jzwI1rGTGM
3	BETFAIR 200/1 Messi to be Top Goalscorer at the #WorldCup, New Customers,Max Stake Â£1. Winnings paid in free bets.â€¦ https://t.co/JcRi25xJkF
4	My #WorldCup Winner is Germany. Predict Yours! https://t.co/T61TDvYBaA #FIFA18

Table 4.114 and Table 4.115 show a sample of unclustered tweets and their T-Information distances to the exemplar Tweets determined by each algorithm. As can be seen, the unique Tweets are relatively distant from each exemplar Tweet which is the expected behaviour.

Table 4.114: Modified Unclustered Exemplar T-Information Distances WorldCup

Modified	Distance to:				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
#WorldCup is just two days away. How big your dream is at the #WorldCup2018 ? 6" or 13.3" ? Select the right-sizeâ€¦ https://t.co/Ml04TvQBD2	0.764	0.818	0.853	0.840	0.745
Are you supporting Egypt in the 2018 World Cup? #Egypt #worldcup #worldcup2018Â #â€¦ https://t.co/7TJ2MkPoYa https://t.co/Cx4ZrM6VnQ	0.747	0.783	0.836	0.806	0.747
Who's excited for the WorldCup 2018 ?!?! â˜ï¸âš½ï¸ ðŸ‡¦ðŸ‡·all the way!! Argentina #WorldCupWithGary	0.838	0.889	0.890	0.964	0.917
@LuluuDeCartoon Can't wait for the #WorldCup ..	0.935	0.918	1.024	0.975	0.904
Our economists created a machine-learning model to predict the #WorldCup winner. Does your prediction match ours?â€¦ https://t.co/6s9GGIJeFc	0.875	0.786	0.870	0.842	0.756

Table 4.115: ITWEC Unclustered Exemplar T-Information Distances WorldCup

ITWEC	**Distance to:**				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
Quem serÃ¡ o vencedor #WorldCup #FIFAWC2018 #russia2018	0.937	0.937	1.005	0.977	0.859
Ø§Ø³Øªà¹Ø¯Ø§Ø§ Ù…Ø¯ÙŠÙ†Ø© #ÙŠÙƒØ§ØªÙŠÙ†Ø¬Ù†Ø¨ Ù„Ø§Ø³Øªà¹Ø§Ø¯Ø© Ù…Ø¹Ø±Ø¶ Ø¥ÙƒØ³Ø¨Ùˆ 2025 #WorldCup | #ThePharaohs | #Russia_2018 #Ø±Ù‚Ø³Ø§ #ÙƒØ£Ø³_Ø§Ù„Ø§Ù„Ù…Ù…â€¦ https://t.co/qew787lUE0	0.943	0.989	0.956	0.935	0.981
#WorldCup âš½ BACK THE #ThreeLions TO WIN THE ðŸ†n ðŸ‘€ IF THEY DON'T WIN IT GET YOUR MONEY BACK! RISK FREE BET! ðŸ’ª Betâ€¦ https://t.co/4mOztb15A8	0.867	0.960	0.772	0.918	0.949
WATCH: Winging it with Zakuani | Yedlin on #WorldCup qualification https://t.co/0ywT6El8pw	0.829	0.826	0.859	0.877	0.677
@pappahapa the worldcup is here, who are supporting? or I should not ask, its obvious!!!	0.995	0.932	1.113	1.049	1.087

Table 4.116 and Table 4.117 list the aggregate cluster T-Information distances for the WorldCup validation. As can be seen, many of the clusters are very far apart. Greater than 1 distances are likely a result of how T-Information handles emoji's and special characters.

Table 4.116: Modified Aggregate Cluster T-Information Distances WorldCup

Modified	**Distance to:**				
Aggregate Cluster	**C0**	**C1**	**C2**	**C3**	**C4**
C0	0.00	1.00	1.01	0.99	1.01
C1	1.00	0.01	1.02	1.00	0.98
C2	1.00	1.00	0.00	0.90	0.96
C3	1.05	0.98	0.91	-0.01	0.93
C4	1.01	1.01	0.91	0.88	0.03

Table 4.117: ITWEC Aggregate Cluster T-Information Distances WorldCup

ITWEC	Distance to:				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.99	1.01	1.00	1.00
C1	1.00	0.01	1.02	1.00	0.97
C2	1.00	0.99	0.00	0.90	0.96
C3	1.01	0.98	0.93	-0.01	0.92
C4	1.00	1.00	0.91	0.90	0.03

4.4.2.4.2 World Cup Jaccard

Table 4.118 - Table 4.125 show the validation results for the Jaccard distance measure on the WorldCup search. Similar to the previous validation results, and as can be see in Table 4.118 - Table 4.122, the exemplar Tweets are either the same or of a similar format across the two thresholding methods. Similarly, the distances between the exemplar Tweets are relatively high indicating that these clusters are spread out.

Table 4.118: Modified Jaccard Exemplar Distances for WorldCup

Modified			Distance to:				
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3	Ex4
0	51	0.254	0.000	0.889	0.892	0.974	0.926
1	32	0.273	0.889	0.000	0.949	0.974	0.964
2	16	0.266	0.892	0.949	0.000	0.921	0.929
3	6	0.287	0.974	0.974	0.921	0.000	1.000
4	6	0.337	0.926	0.964	0.929	1.000	0.000

Table 4.119: Modified Jaccard Exemplar Tweets for WorldCup

Cluster No.	Exemplar Tweets
0	@AdamsBakuli1 Thanks for subscribing! Up next on your World of Champions is the 2018 FIFA #WorldCup âš½ï¸• Connect withâ€¦ https://t.co/CRKHMT1Qak
1	@_marrrrrria_ Score! FIFA #WorldCup reminders are coming your way! Download the FOX Sports App now and never miss aâ€¦ https://t.co/okW5gSvG9V
2	888 #WorldCup 2018 Winner (Outright) ðŸ† 40/1 #Belgium ðŸ‡§ðŸ‡ª to win the World Cup â€“ MAX BET Â£5 Promo-code: BOOSTâ€¦ https://t.co/aTc3xQVAtQ
3	BETFAIR 100/1 Brazil to win the #WorldCup, New Customers,Max Stake Â£1. Winnings paid in free bets. Full T&Cs Apply.â€¦ https://t.co/FaOavnRLnv
4	My #WorldCup Winner is Brazil. Predict Yours! https://t.co/cdi1H0J40L #FIFA18

Table 4.120: ITWEC Jaccard Exemplar Distances for WorldCup

ITWEC			Distance to:				
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3	Ex4
0	50	0.254	0.000	0.889	0.892	0.974	0.926
1	31	0.272	0.889	0.000	0.949	0.974	0.964
2	15	0.262	0.892	0.949	0.000	0.921	0.929
3	5	0.317	0.974	0.974	0.921	0.000	1.000
4	5	0.330	0.926	0.964	0.929	1.000	0.000

Table 4.121: ITWEC Jaccard Exemplar Tweets for WorldCup

Cluster No.	Exemplar Tweets
0	@AdamsBakuli1 Thanks for subscribing! Up next on your World of Champions is the 2018 FIFA #WorldCup âš½ï¸• Connect withâ€¦ https://t.co/CRKHMT1Qak
1	@_marrrrrria_ Score! FIFA #WorldCup reminders are coming your way! Download the FOX Sports App now and never miss aâ€¦ https://t.co/okW5gSvG9V
2	888 #WorldCup 2018 Winner (Outright) ðŸ† 40/1 #Belgium ðŸ‡§ðŸ‡ª to win the World Cup â€“ MAX BET Â£5 Promo-code: BOOSTâ€¦ https://t.co/aTc3xQVAtQ
3	BETFAIR 100/1 Brazil to win the #WorldCup, New Customers,Max Stake Â£1. Winnings paid in free bets. Full T&Cs Apply.â€¦ https://t.co/D0G425r4aE
4	My #WorldCup Winner is Brazil. Predict Yours! https://t.co/cdi1H0J40L #FIFA18

Table 4.122 and Table 4.123 show the Jaccard distances from a sample of unclustered Tweets to the exemplar Tweets as defined by each thresholding algorithm. As can be seen the unique Tweets are significantly distant from the exemplar Tweets.

Table 4.122: Modified Unclustered Exemplar Jaccard Distances WorldCup

Modified	**Distance to:**				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
All the cyber threats that fans and footballers face at the #WorldCup https://t.co/HxB1e5MQhH	0.933	0.897	0.935	0.96	0.95
@LuluuDeCartoon Can't wait for the #WorldCup ..	0.875	0.920	0.923	0.96	0.93
#Russia vs #Egypt | #FIFA #WorldCup 2018 Group A | Match 17 Predictions #FIFA18 #ThePharaohs #EGY #Egipto #RUSvEGYâ€¦ https://t.co/m2mNZ0UERW	0.944	0.973	0.946	1.00	0.92
England to be eliminated at the quarter finals is 9/4 they've a good young side this year #EnglandSquad #WorldCup	0.917	0.946	0.919	0.94	0.92
DeNet: BLOCKCHAÄ°N ALT YAPILI WEB HOSTÄ°NG https://t.co/loL3mI7e1F Microsoft Bethesda #E32018 #WorldCup #Ð¡Ð²Ð¾Ð±Ð¾Ð´ÑƒÐ¢Ð²Ð¸,Ñ,Ð¾Ñ€Ñ‡Ð°Ð½Ð•Ð¾ÐœÑ‹Ð¢Ð¾Ð¶ ÐµÐ›ÑŽÐ´Ð¸	0.968	0.968	0.969	1.00	0.95

Table 4.123: ITWEC Unclustered Exemplar Jaccard Distances WorldCup

ITWEC	**Distance to:**				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
My Predictions for the #WorldCup #WorldCup2018 #FIFAWorldCup #FIFAWorldCup2018 #Russia2018 https://t.co/eK0D791Ptb	0.889	0.929	0.931	0.966	0.882
#WorldCup #FanPredict https://t.co/rVQmtparRi	0.955	0.955	0.957	1.000	0.909
#WorldCup #TrumpKimSummit #UNITEDF4 #2018BTSFESTA #Singapore #CanadianGP #Ð¿Ñ‚Ð¸Ñ…Ð¾Ð»Ð³Ð¸Ñ‚ https://t.co/2BXfJjokQx	0.963	0.963	0.964	1.000	0.938
#Serbia vs #Switzerland | #FIFA #WorldCup 2018 Group E | Match 26 Predictions #FIFA18 #Orlovi #FSS #SRB #SUIâ€¦ https://t.co/BL2zjAhYP3	0.944	0.973	0.946	1.000	0.920
#DieMannschaft #WorldCup	0.952	0.952	0.955	1.000	0.900

Table 4.124 and Table 4.125 detail the aggregate cluster distances as measured using the Jaccard similarity measure. As can be seen, the aggregate of the clusters are significantly far apart indicating they share very little common information.

Table 4.124: Modified Aggregate Cluster Jaccard Distances WorldCup

Modified	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.98	0.98	0.99	0.99
C1	0.98	0.00	0.99	0.99	0.99
C2	0.98	0.99	0.00	0.95	0.97
C3	0.99	0.99	0.95	0.00	1.00
C4	0.99	0.99	0.97	1.00	0.00

Table 4.125: ITWEC Aggregate Cluster Jaccard Distances WorldCup

ITWEC	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.98	0.98	0.99	0.99
C1	0.98	0.00	0.98	0.99	0.99
C2	0.98	0.98	0.00	0.95	0.97
C3	0.99	0.99	0.95	0.00	1.00
C4	0.99	0.99	0.97	1.00	0.00

4.4.2.4.3 World Cup Levenshtein

Table 4.126 to Table 4.133 present the results for the Levenshtein distance validation of a WorldCup search. Table 4.126 - Table 4.129 show the exemplar Tweets and their respective distances to each other. As can be seen, and similar to the Jaccard section, the Tweets share a common format and have a very low exemplar RMSD indicating that

the data is clusters contain highly redundant information. The exemplar Tweets are also seen to be significantly distant from each other.

Table 4.126: Modified Levenshtein Exemplar Distances for WorldCup

Modified			Distance to:				
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3	Ex4
0	51	0.154	0.000	0.950	1.000	1.000	0.950
1	32	0.176	0.950	0.000	0.952	1.000	0.950
2	16	0.158	1.000	0.952	0.000	1.000	0.905
3	6	0.196	1.000	1.000	1.000	0.000	1.000
4	6	0.205	0.950	0.950	0.905	1.000	0.000

Table 4.127: Modified Levenshtein Exemplar Tweets for WorldCup

Cluster No.	Exemplar Tweets
0	@AdamsBakuli1 Thanks for subscribing! Up next on your World of Champions is the 2018 FIFA #WorldCup âš½ï¸• Connect withâ€¦ https://t.co/CRKHMT1Qak
1	@_marrrrrria_ Score! FIFA #WorldCup reminders are coming your way! Download the FOX Sports App now and never miss aâ€¦ https://t.co/okW5gSvG9V
2	888 #WorldCup 2018 Winner (Outright) ðŸ† 40/1 #Belgium ðŸ‡§ðŸ‡ª to win the World Cup â€“ MAX BET Â£5 Promo-code: BOOSTâ€¦ https://t.co/aTc3xQVAtQ
3	BETFAIR 100/1 Brazil to win the #WorldCup, New Customers,Max Stake Â£1. Winnings paid in free bets. Full T&Cs Apply.â€¦ https://t.co/FaOavnRLnv
4	My #WorldCup Winner is Brazil. Predict Yours! https://t.co/cdi1H0J40L #FIFA18

Table 4.128: Modified Levenshtein Exemplar Distances for WorldCup

ITWEC			Distance to:				
Cluster No.	Cluster Size	Exemplar RMSD	Ex0	Ex1	Ex2	Ex3	Ex4
0	50	0.154	0.000	0.950	1.000	1.000	0.950
1	31	0.175	0.950	0.000	0.952	1.000	0.950
2	15	0.155	1.000	0.952	0.000	1.000	0.905
3	5	0.218	1.000	1.000	1.000	0.000	1.000
4	5	0.200	0.950	0.950	0.905	1.000	0.000

Table 4.129: ITWEC Levenshtein Exemplar Tweets for WorldCup

Cluster No.	Exemplar Tweets
0	@Donaldovwildout Thanks for subscribing! Up next on your World of Champions is the 2018 FIFA #WorldCup âš½ï¸• Connect wâ€¦ https://t.co/WbbzQeGI32
1	@mescobedo11 Score! FIFA #WorldCup reminders are coming your way! Download the FOX Sports App now and never miss aâ€¦ https://t.co/vM4TEfE0jM
2	888 #WorldCup 2018 Winner (Outright) ðŸ† 40/1 #Belgium ðŸ‡§ðŸ‡ª to win the World Cup â€“ MAX BET Â£5 Promo-code: BOOSTâ€¦ https://t.co/aTc3xQVAtQ
3	BETFAIR 100/1 Brazil to win the #WorldCup, New Customers,Max Stake Â£1. Winnings paid in free bets. Full T&Cs Apply.â€¦ https://t.co/D0G425r4aE
4	My #WorldCup Winner is Brazil. Predict Yours! https://t.co/cdi1H0J40L #FIFA18

Modified	Distance to:				
Unclustered Tweets	Ex0	Ex1	Ex2	Ex3	Ex4
#WorldCup 2018: Traveling to #Russia? Here's what you need to know; Russia has some very restrictive #cybersecurityâ€¦ https://t.co/JT95vNvDfG	1.00	1.00	0.95	0.95	1.00
@DudleyWFEN I believe there was some tension between the two players before the session Dudley. Some form of verbal exchange. #WorldCup	0.95	1.00	1.00	1.00	1.00
A look back on my thoughts on 2014's opening ceremony - FAIL! https://t.co/G0YZ8tnslR #WorldCup #Russia2018â€¦ https://t.co/1iaPfRwXnQ	0.90	1.00	1.00	1.00	1.00
Got ETH guessing on Australlia vs Brazil! #11q #WorldCup https://t.co/97K3JUP3fZ https://t.co/gAOEQjRj4n	0.90	1.00	1.00	1.00	1.00
That can have an impact on the future #WorldCup #WorldCup2018 #baseH #Dante #AIwriting https://t.co/3LfRFDL8Bj	0.90	0.95	0.95	1.00	1.00

Table 4.130 and Table 4.131 show the Levenshtein distances between a set of unclustered Tweets and the exemplar Tweets. As can be seen, they follow the expected behaviour and are not quantitatively similar to the exemplar Tweets.

Table 4.130: Modified Unclustered Exemplar Levenshtein Distances WorldCup

Modified	**Distance to:**				
Unclustered Tweets	Ex0	Ex1	Ex2	Ex3	Ex4
#WorldCup 2018: Traveling to #Russia? Here's what you need to know; Russia has some very restrictive #cybersecurityâ€¦ https://t.co/JT95vNvDfG	1.00	1.00	0.95	0.95	1.00
@DudleyWFEN I believe there was some tension between the two players before the session Dudley. Some form of verbal exchange. #WorldCup	0.95	1.00	1.00	1.00	1.00
A look back on my thoughts on 2014's opening ceremony - FAIL! https://t.co/G0YZ8tnslR #WorldCup #Russia2018â€¦ https://t.co/1iaPfRwXnQ	0.90	1.00	1.00	1.00	1.00
Got ETH guessing on Australlia vs Brazil! #11q #WorldCup https://t.co/97K3JUP3fZ https://t.co/gAOEQjRj4n	0.90	1.00	1.00	1.00	1.00
That can have an impact on the future #WorldCup #WorldCup2018 #baseH #Dante #AIwriting https://t.co/3LfRFDL8Bj	0.90	0.95	0.95	1.00	1.00

Table 4.131: ITWEC Unclustered Exemplar Levenshtein Distances WorldCup

ITWEC	**Distance to:**				
Unclustered Tweets	**Ex0**	**Ex1**	**Ex2**	**Ex3**	**Ex4**
#CostaRica vs #Serbia | #FIFA #WorldCup 2018 Group E | Match 9 Predictions #FIFA18 #Orlovi #FSS #SRB #CRCSRBâ€¦ https://t.co/q1c177RGsi	1.00	1.00	1.00	1.00	0.89
Imagine watching 24 hours of Le Mans over the #WorldCup this weekend Football all the way	0.95	0.95	0.95	1.00	1.000
4) An England fan in the stands at Euro 2016, Harry Maguire will get a much closer look at this #WorldCup #ENGâ€¦ https://t.co/g43LkInOWl	1.00	1.00	1.00	0.95	1.00
On June 15 #Iran will play its first #WorldCup game as the ONLY country to ban women from entering sports stadiums.â€¦ https://t.co/0Mhv4ZVGFM	1.00	0.95	0.95	1.00	0.95
Romelu #Lukaku has been in great form for @BelRedDevils, scoring 12 goals in his past 9 games in preparation for the #WorldCup	1.00	1.00	1.00	1.00	1.00

Table 4.132 to Table 4.133 show the intercluster distances of the aggregate clusters. As can be seen the cluster are all significantly spread from each other in the measure space.

Table 4.132: Modified Unclustered Exemplar Levenshtein Distances WorldCup

Modified	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.90	0.95	0.99	0.99
C1	0.90	0.00	0.95	0.99	0.99
C2	0.95	0.95	0.00	0.94	0.96
C3	0.99	0.99	0.94	0.00	1.00
C4	0.99	0.99	0.96	1.00	0.00

Table 4.133: ITWEC Unclustered Exemplar Levenshtein Distances WorldCup

ITWEC	**Distance to:**				
Aggregate Cluster	C0	C1	C2	C3	C4
C0	0.00	0.90	0.95	1.00	0.99
C1	0.90	0.00	0.95	0.99	0.99
C2	0.95	0.95	0.00	0.94	0.97
C3	1.00	0.99	0.94	0.00	1.00
C4	0.99	0.99	0.97	1.00	0.00

4.4.2.5 Word Cloud Validation

The final form of validation carried out was a word cloud of each simulation permutation. As the world clouds are highly redundant, only the T-Information for the largest clusters and the unclustered dataset are shown here. Word clouds corresponding to the other validation tests are included in Appendix A. The samples used to generate the word clouds do not necessarily match the samples used to generate the quantitative validation results in the previous section.

4.4.2.5.1 Vancouver Word Cloud Validation

Figure 4.48 and Figure 4.49 show the word clouds for the largest cluster and the unclustered Tweets, respectively. Both word clouds have are relatively dense, indicating

there is a lot of content in each set. Interestingly, there is a significant amount of content overlap between the two word clouds indicating the clusters are highly similar subsamples of Tweets. It also suggests that a topic based approach may not be an effective technique to cluster posts.

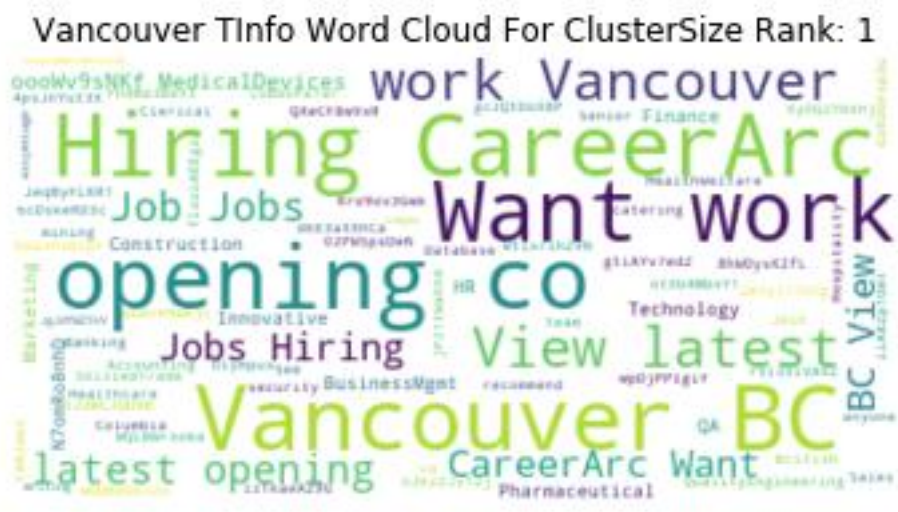

Figure 4.48: Vancouver T-Information Largest Cluster Word Cloud

Figure 4.49: Vancouver T-Information Unclustered Content Word Cloud

4.4.2.5.2 London Word Cloud Validation

Figure 4.50 and Figure 4.51 show the word clouds for the largest cluster and the unclustered Tweets, respectively. As can be seen, there is far more content in the unclustered Tweets word cloud than the cluster because there are far more unique words across the unclustered content. The clustered word cloud very clearly includes a link oriented content that has a number of unique short links.

Figure 4.50: London T-Information Largest Cluster Word Cloud

Figure 4.51: London T-Information Unclustered Content Word Cloud

4.4.2.5.3 RoyalWedding Word Cloud Validation

As can be seen in Figure 4.52 and Figure 4.53, the RoyalWedding word clouds of the top rank and the unclustered content follow the same format as London. The largest cluster has redundant content that includes lots of variable short links, while the unclustered word cloud has a significant amount of unique information.

Figure 4.52: RoyalWedding T-Information Largest Cluster Word Cloud

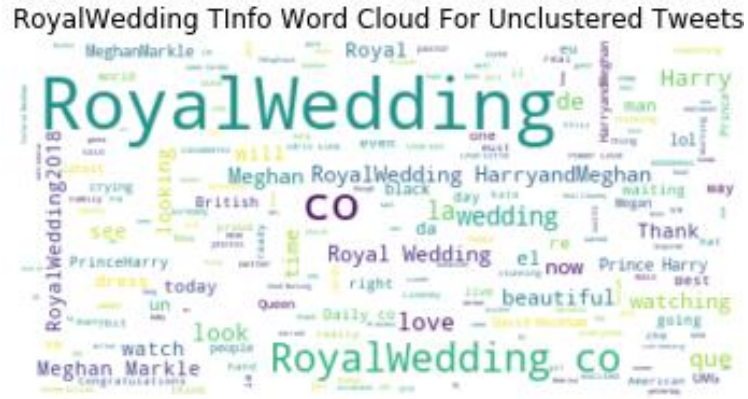

Figure 4.53: RoyalWedding T-Information Unclustered Content Word Cloud

4.4.2.5.4 WorldCup Word Cloud Validation

Figure 4.54 and Figure 4.55 are the word clouds for the largest cluster and the unclustered Tweets. The WorldCup largest cluster also follows the format of repetitive content and short link strings, but seems to include more content in the Tweet than the previous examples. The unclustered content has a significant amount of unique content and some overlap with the largest cluster.

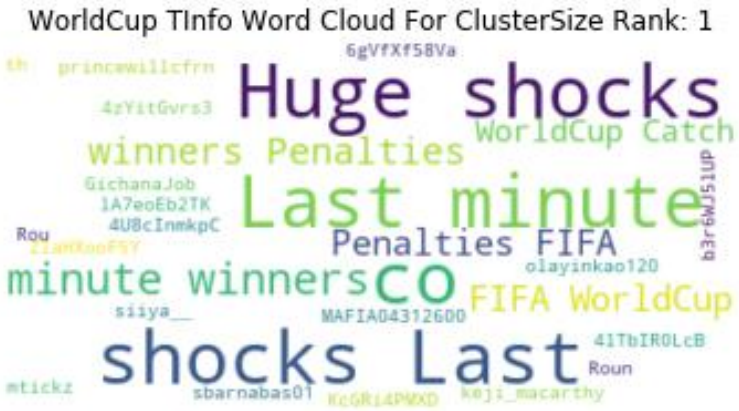

Figure 4.54: WorldCup T-Information Largest Cluster Word Cloud

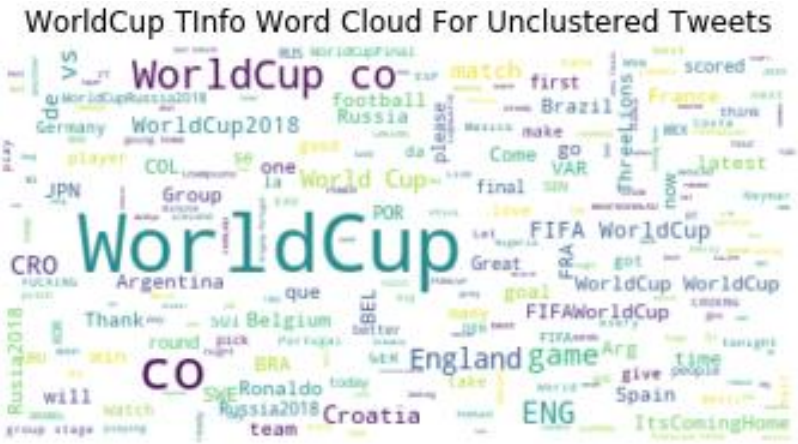

Figure 4.55: WorldCup T-Information Unclustered Content Word Cloud

See Appendix A for more word cloud examples.

4.5 Real-Time Data Stream Clustering Simulation

The following tables present the results from running the real-time data streaming simulation. Comparisons to the streaming data statistics from Section 3.5.4 are also discussed.

4.5.1 Run Statistics for Data Stream Clustering Simulation

Tables show the run statistics for the real-time data stream simulation. The mean metrics presented in the tables are explained in Table 4.1. For the purposes of this evaluation cluster characteristics refer to the number of clusters, the max cluster size and the average RDSD, the reduction characteristics refer to the Reduction and the number of unclustered posts, and the complexity characteristics refer to the total time and number of calculations. The value presented in the Mean Percent Difference column is the percent difference between the quick cluster method and the full cluster method for the given metric as outlined in the methodology. The calculation for percent difference is provided in Equation 3.5.

4.5.1.1 Vancouver Stream Results

Table 4.134 - Table 4.140 show the Vancouver stream simulation results for cluster characteristics, reduction characteristics, and complexity characteristics. For cluster characteristics including number of clusters, max cluster size and RMSD in Table 4.134, Table 4.135, and Table 4.136 there is a significant difference between the full cluster method and the quick cluster method in the number of clusters for both algorithms and all similarity measures. However, it is also seen that the maximum cluster size is consistent across both the full cluster and the quick cluster methods for all simulations. This indicates, for Vancouver, the compressible nature of the data set can be exploited. It is also noted that the mean cluster distance is smaller for the full cluster method, so the full cluster method is generating tighter clusters. T-Information has the least variance in cluster density as it generates the largest clusters and its continuous nature allows for more similar content to be clustered.

Table 4.134: Vancouver Stream Simulation Number of Clusters

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Number of Clusters	Jaccard	27	91.3	70.4%	24.2	81.8	70.5%
	LevToken	23.2	77.3	70.0%	18	60	70.0%
	TInfo	18.7	115.3	83.8%	13.2	87.3	84.9%

Table 4.135: Vancouver Stream Simulation Max Cluster Size

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Max Cluster Size	Jaccard	163.5	182.7	10.5%	160.7	181.7	11.6%
	LevToken	166.8	167	0.1%	165.8	166	0.1%
	TInfo	165.7	167	0.8%	165.5	166	0.3%

Table 4.136: Vancouver Stream Simulation Average RMSD

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Average Cluster RMSD	Jaccard	0.38	0.32	19.2%	0.4	0.35	16.1%
	LevToken	0.26	0.19	38.8%	0.27	0.2	35.7%
	TInfo	0.26	0.19	37.6%	0.29	0.2	46.4%

Table 4.137 and Table 4.138 show the reduction characteristics for Vancouver including the Reduction and number of unclustered Tweets. While the quick cluster method does not reduce the data set as significantly as the full cluster method, the resulting data set is between 60% and 45% of it's original size. The Levenshtein string token method can be seen to perform the best on reduction, closely followed by T-Information. This is likely a result of Vancouver's heavy career related Tweet content favouring the Levenshtein edit distance and the continuous nature of T-Information over the Jaccard similarity distance.

Table 4.137: Vancouver Stream Simulation Reduction

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Reduction	Jaccard	0.47	0.35	36.6%	0.5	0.37	33.7%
	LevToken	0.43	0.33	29.1%	0.45	0.37	23.0%
	TInfo	0.58	0.42	38.2%	0.6	0.47	26.5%

Table 4.138: Vancouver Stream Simulation Unclustered Tweets

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Unclustered Tweets	Jaccard	1159.7	777.7	49.1%	1223.2	851.3	43.7%
	LevToken	1045.5	750.7	39.3%	1115.3	861.2	29.5%
	TInfo	1428.3	931.8	53.3%	1486.8	1098.2	35.4%

Finally, when observing Vancouver's results for complexity including total time and number of calculations, in Table 4.139 and Table 4.140, the quick cluster method does not outperform the full cluster for all Modified thresholding algorithm cases. This is the result of the added overhead of the quick cluster method. As the full cluster method doesn't scale as well as the quick cluster method, the quick cluster method show performance gains for large sample sizes. Vancouver's worst-case data throughput from Table 3.4 was 750 posts per day. As a result, any method for clustering Vancouver's content would be effective, therefore, quick clustering would not be beneficial.

Table 4.139: Vancouver Stream Simulation Time

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Time (s)	Jaccard	74.3	46.6	59.3%	66.5	94.4	29.5%
	LevToken	74	89.2	17.0%	63.6	185.7	65.8%
	TInfo	53.3	60.6	12.1%	42.5	132.7	68.0%

Table 4.140: Vancouver Stream Simulation Total Calculations

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Total Calculations	Jaccard	84957	430438	80.3%	111004	856119	87.0%
	LevToken	74124	391939	81.1%	93818	808198	88.4%
	TInfo	78674	606016	87.0%	118525	1307609	90.9%

4.5.1.2 London Stream Results

Table 4.141 to Table 4.147 show the run characteristics of the real-time simulation for the London search. Included in Table 4.141, Table 4.142, Table 4.143, are the cluster characteristics of the streaming simulation. As observed, there is a significant difference in the number of clusters generated by the full cluster method. However, there appears to be little difference in cluster quality or the maximum cluster size between the two methods. The one exception appears to be the Jaccard-ITWEC clustering method, where the quick cluster had a significantly higher RMSD than the full cluster method. This is likely due to the ITWEC algorithm clustering a group of Tweets into two or more dense clusters and the quick cluster method grouping the same set into a single, less dense cluster.

Table 4.141: London Stream Simulation Number of Clusters

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Number of Clusters	Jaccard	10.0	40.0	75.0%	6.3	27.0	76.5%
	LevToken	11.5	51.0	77.5%	7.0	35.3	80.2%
	TInfo	12.2	51.8	76.5%	8.8	41.3	78.6%

Table 4.142: London Stream Simulation Max Cluster Size

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Max Cluster Size	Jaccard	26.0	26.8	3.1%	25.2	25.8	2.6%
	LevToken	37.3	39.0	4.3%	36.3	38.0	4.4%
	TInfo	42.7	44.5	4.1%	42.8	43.5	1.5%

Table 4.143: London Stream Simulation RMSD

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Average Cluster RMSD	Jaccard	0.292	0.267	9.4%	0.388	0.288	34.7%
	LevToken	0.255	0.239	6.5%	0.260	0.261	0.4%
	TInfo	0.297	0.307	3.3%	0.303	0.331	8.4%

As can be see in Table 4.144 and Table 4.145 for reduction characteristics Reduction and unclustered Tweets, the full clustering method still reduce the dataset by significantly more than the quick cluster method. Notably however, the quick cluster method for the modified thresholding algorithm outperforms the ITWEC version. This is likely due to the modified algorithm generating a large number of small clusters, which gives the stream data more clusters to measures against.

Table 4.144: London Stream Simulation Reduction

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Reduction	Jaccard	0.97	0.92	4.8%	0.97	0.94	3.5%
	LevToken	0.95	0.89	7.2%	0.96	0.91	5.4%
	TInfo	0.95	0.89	6.8%	0.96	0.91	5.4%

Table 4.145: London Stream Simulation Unclustered Tweets

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Unclustered Tweets	Jaccard	2403.2	2262.5	6.2%	2427.2	2325.3	4.4%
	LevToken	2372.3	2173.7	9.1%	2402.8	2251.2	6.7%
	TInfo	2366.3	2174.8	8.8%	2391.7	2235.3	7.0%

As can be see in Table 4.146 and Table 4.147 for total time and calculations, the quick cluster method is significantly faster than the full cluster method for all cases. When comparing the execution time to the industry constraints from Table 3.4, London's worst-case daily stream rate was about 3.88 posts per second, or 644 seconds for 2500 posts. By comparison, it appears that each full cluster method would be adequate to cluster the content in real-time, except for the Levenshtein string-token pair with the I-TWEC thresholding algorithm. Of course, as the sample size of the stream increases, the full cluster method will not be capable of clustering it in real-time and only the quick cluster methods will be viable. It does suggest, however, that the seed sample size in a real-world scenario could increase to about 2000 before becoming restrictive. As the stream size increases, quick clustering would be a beneficial method for filtering content for a London style search.

Table 4.146: London Stream Simulation Time

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Time (s)	Jaccard	40.54	261.44	84.5%	39.59	536.04	92.6%
	LevToken	55.10	489.33	88.7%	64.07	1021.83	93.7%
	TInfo	45.40	261.54	82.6%	46.35	545.34	91.5%

Table 4.147: London Stream Simulation Total Calculations

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Total Calculations	Jaccard	129184	2681494	95.2%	238926	5508472	95.7%
	LevToken	128410	2519350	94.9%	234657	5188845	95.5%
	TInfo	128138	2510646	94.9%	233867	5169349	95.5%

4.5.1.3 RoyalWedding Stream Results

Table 4.148 to Table 4.154 detail the RoyalWedding stream simulation results for cluster, reduction, and complexity characteristics. For the cluster characteristics in Table 4.148, Table 4.149, and Table 4.150 it can be seen there is a significant difference between the full cluster method and the quick cluster method in both the number of clusters and the maximum cluster size for all algorithms and measures. Notably, the average cluster distance, as measured by the root mean square distance, is similar between the full cluster and quick cluster methods. The modified thresholding algorithm paired with T-Information was the most accurate. T-Information had the least dense clusters indicating it may include more results in each cluster, which is supported by the max cluster size being highest. This is likely also a result of RoyalWedding's containing a modest number of small clusterable Tweets that T-Information is capable of clustering, while Jaccard and Levenshtein are not.

Table 4.148: RoyalWedding Stream Simulation Number of Clusters

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Number of Clusters	Jaccard	8.33	38.5	78.4%	7.5	33.5	77.6%
	LevToken	4.33	16.83	74.3%	3.33	12.17	72.6%
	TInfo	4	13.17	69.6%	3	9.17	67.3%

Table 4.149: RoyalWedding Stream Simulation Max Cluster Size

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Max Cluster Size	Jaccard	27.33	42.83	36.2%	29.5	42.17	30.0%
	LevToken	14	16.83	16.8%	10.5	15.83	33.7%
	TInfo	13.17	15.83	16.8%	10.83	14.83	27.0%

Table 4.150: RoyalWedding Stream Simulation Cluster RMSD

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Average Cluster RMSD	Jaccard	0.37	0.38	1.1%	0.39	0.42	8.4%
	LevToken	0.17	0.26	33.9%	0.25	0.27	7.2%
	TInfo	0.15	0.21	29.2%	0.26	0.25	4.8%

From Table 4.151 and Table 4.152, it can be seen RoyalWedding does not have a significant amount of clusterable content and, therefore, we see only modest improvements from the full clustering algorithm over the quick cluster in the reduction characteristics. Again, T-Information reliably finds the most reduction because of presence of small clusterable Tweets in RoyalWedding.

Table 4.151: RoyalWedding Stream Simulation Reduction

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Reduction	Jaccard	0.96	0.91	5.0%	0.96	0.92	4.7%
	LevToken	0.98	0.97	1.9%	0.99	0.97	1.4%
	TInfo	0.99	0.97	1.3%	0.99	0.98	1.0%

Table 4.152: RoyalWedding Stream Simulation Unclustered Tweets

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Unclustered Tweets	Jaccard	2379.8	2235.2	6.5%	2391.8	2259	5.9%
	LevToken	2453	2395.7	2.4%	2461.5	2419.3	1.7%
	TInfo	2463.2	2422.8	1.7%	2469.8	2440.3	1.2%

When looking at RoyalWedding's time and calculation complexity in Table 4.153 and Table 4.154, it can be seen that there is a dramatic improvement from the quick clustering method over the full cluster method, especially for the ITWEC thresholding algorithm. The worst-case data throughput for RoyalWedding from Table 3.4 was 258 posts per second and required that 2500 post be clustered in roughly 9 seconds. As a result, neither the quick cluster nor the full cluster method would capable of clustering the RoyalWedding content in real-time.

Table 4.153: RoyalWedding Stream Simulation Time

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Time (s)	Jaccard	37.2	269.2	86.2%	42.9	556	92.3%
	LevToken	38.2	533.9	92.8%	53.5	1096.4	95.1%
	TInfo	27.2	290.4	90.6%	33.5	593.5	94.4%

Table 4.154: RoyalWedding Stream Simulation Total Calculations

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Total Calculations	Jaccard	122439	2551351	95.2%	230907	5263306	95.6%
	LevToken	123640	2886833	95.7%	240693	5902838	95.9%
	TInfo	125096	2950819	95.8%	242748	5994490	96.0%

4.5.1.4 WorldCup Stream Results

Table 4.155 to Table 4.161 present the results for the WorldCup real-time stream simulation. Similar to the previous simulations, the quick cluster method does not find as many clusters as the full cluster method as seen in Table 4.155. Interestingly, for WorldCup the two methods find the same max cluster for each run, as reflected in the mean max cluster size. This is due to a significant presence of a semi-automated set of posts that appears through the data set and are obviously clusterable.

Table 4.155: World Cup Stream Simulation Number of Clusters

		Modified			**ITWEC**		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Number of Clusters	Jaccard	7.2	45.4	84.1%	5.2	29.4	82.3%
	LevToken	6.2	38.8	84.0%	4.8	20.8	76.9%
	TInfo	5.8	36.2	84.0%	4.8	20	76.0%

Table 4.156: World Cup Stream Simulation Max Cluster Size

		Modified			**ITWEC**		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Max Cluster Size	Jaccard	540.2	540.2	0.0%	539.4	539.2	0.0%
	LevToken	539.6	539.6	0.0%	538.6	538.6	0.0%
	TInfo	539.8	539.8	0.0%	538.8	538.8	0.0%

Table 4.157: World Cup Stream Simulation Cluster RMSD

		Modified			**ITWEC**		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Average Cluster RMSD	Jaccard	0.23	0.24	5.7%	0.21	0.26	19.0%
	LevToken	0.12	0.17	27.2%	0.09	0.18	50.3%
	TInfo	0.16	0.21	24.3%	0.18	0.23	21.8%

With respect to the reduction characteristics in Table 4.158 and Table 4.159, it was observed that the reduction of the original data set did not vary significantly for algorithm type and was better for the full cluster methods. The similarity was likely due to the large cluster of similar semi-automated posts directly influencing the reduction characteristics. This result supports the idea that a quick cluster method would be an equally effectual technique for removing heavy redundancies from data.

Table 4.158: World Cup Stream Simulation Reduction

		Modified			**ITWEC**		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Reduction	Jaccard	0.74	0.66	12.6%	0.75	0.68	10.5%
	LevToken	0.74	0.67	10.8%	0.75	0.69	8.5%
	TInfo	0.74	0.67	10.2%	0.75	0.69	8.2%

Table 4.159: World Cup Stream Simulation Unclustered Tweets

		Modified			**ITWEC**		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Unclustered Tweets	Jaccard	1837.2	1593.2	15.3%	1863.4	1662	12.1%
	LevToken	1844.2	1631	13.1%	1867.8	1705.4	9.5%
	TInfo	1849	1646.2	12.3%	1870.2	1713	9.2%

Lastly, a reduction in the overall time and complexity for clustering the WorldCup content was observed, as seen in Table 4.160 and Table 4.161. The worst-case data throughput for WorldCup from Table 3.4 was 109 posts per second or 2500 posts in 22.9 seconds. Therefore, neither the quick cluster nor the full cluster method as currently implemented would be sufficient to cluster it in real-time. It is possible, however, that an industry scale implementation could reduce the calculations per second to a point that the quick cluster methods would be effective.

Table 4.160: World Cup Stream Simulation Time

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Time (s)	Jaccard	29.7	164.4	81.9%	27.9	339.9	91.8%
	LevToken	35.9	318.8	88.7%	41.7	666.3	93.7%
	TInfo	25.9	163.2	84.1%	26.9	337.9	92.0%

Table 4.161: World Cup Stream Simulation Total Calculations

		Modified			ITWEC		
	Measure	Quick Cluster	Full Cluster	Percent Difference	Quick Cluster	Full Cluster	Percent Difference
Mean Total Calculations	Jaccard	81584	1572715	94.8%	145359	3275557	95.6%
	LevToken	80606	1637675	95.1%	146015	3397178	95.7%
	TInfo	80289	1655048	95.1%	146633	3421323	95.7%

4.6 Chapter Summary

Chapter 4 presented the results of the thesis. The effects of thresholding, sample size, and minimum cluster size were explored. Then an industry relevant simulation of 500 Tweet samples was presented with a quantitative and qualitative validation. Finally, a streaming simulation was presented.

It was observed that cluster quality was directly influenced by the threshold selected for clustering. A reasonable value of 0.4 was determined to be used to create meaningful clusters and avoid multi-modal clustering. It was also seen that T-Information formed continuous clusters while the token based similarity measures found narrowly banded clusters.

Sample size was also determined to influence clustering in expected manners. Cluster size, total calculations and max cluster size all increased with sample size. For small clusters sizes, it was found that each search had different clustering behaviours. Most notably, RoyalWedding frequently did not find clusters for sample sizes 500 or smaller for Jaccard or Levenshtein string token similarity. T-Information was still able to determine clusters, in part, due to frequent two term Tweets in RoyalWedding that the other measures could not detect. It was also seen that the Modified algorithm was significantly faster than the ITWEC algorithm for larger sample sizes.

Minimum cluster size had the expected result of reducing the number of clusters in the set and reducing the amount of time for an experiment to run. A high minimum cluster value was also found to inhibit the clustering process and should not exceed 10% of the sample size as an upper bound.

In the 500 Tweet search simulation several differences were observed between the thresholding algorithm and the similarity measures. Generally speaking, the ITWEC thresholding algorithm generated better clusters and found more clusters than the modified algorithm. However, the modified algorithm executed significantly faster and the difference in cluster quality was modest. For similarity measures, it was seen that T-

Information and Levenshtein distance found similar quality clusters that were better than the Jaccard measure. However, the Levenshtein distance measure was seen to operate significantly slower than both Jaccard and T-Information. Jaccard similarity was not seen to have discernable advantage over T-Information or Levenshtein distances in a meaningful measure.

The validation process confirmed that each clustering method was finding similar cluster results. Clusters also appeared to be significantly different from each other and unclustered content. Upon inspection Levenshtein string token distance incorrectly classified posts with some similarity as very distant, likely a result of significant edits and the left to right nature of the algorithm, whereby the similarity exists at the right side of the respective strings.

Finally the streaming simulation showed that in industry contexts these algorithms would perform for some searches but not all. A quick search method could be used to circumvent computational complexity, however, it would not be fast enough for very high throughput searches like WorldCup or RoyalWedding.

Chapter 5

5 Conclusions and Future Work

This chapter presents an evaluation of the results, some limitations experiment methodology, potential future works, and the implications of this thesis.

5.1 Conclusions

Given the significant amount of social media content that is generated every day, it is necessary to establish methods for reducing the amount of information required to be reviewed and understood before important decisions are made. In social media, it is common to see redundant or irrelevant information return from each search that is unique to each search topic or area. Therefore, it is necessary to find techniques to remove as much of the redundant information in the search dataset as possible that are not preterminal or set. Hence, it is necessary to establish techniques for the purposes of clustering similar content within a dataset, or intra-set clustering. Ultimately, this thesis developed and evaluated several intra-set clustering techniques for practical implementation and effectiveness.

5.1.1 Evaluation of the Results

The purpose of this thesis was to evaluate different, intelligent filtering techniques for content rich social data in an industry relevant context. Specifically, this thesis focused on finding intra-set clusters for a given social media search, instead of the prevailing academic goal of separating two topics from a pool of Tweets. To this end, two basic algorithms were explore for clustering content, one presented by Arin et al in I-TWEC [17] and a modified thresholding algorithm version that improved on the computational

complexity in exchange for reduced clustering performance. Each of these algorithms required a similarity measure to cluster the content. For this purpose, three measures were explored. Two measures, the Jaccard similarity measure and the Levenshtein similarity measure, are well established in academia and effective tools for string and document similarity comparisons. The final measure, T-Information similarity, is a recently established technique from information theory that can be adapted to measure the information distance between two strings. It's applications in measuring similarity between social media posts has not been explored. Each similarity measure and threshold algorithm are dependent on various parameters to cluster content including a minimum cluster size, a distance threshold. Ultimately, experiments were designed to test each threshold method and similarity measure to practically establish appropriate values for the parameters. Then tests were run against search dataset to simulate an industry relevant application and the results were evaluated.

The effects of distance thresholding, sample size, and minimum cluster size were explored to determine how the clustering algorithms' performance would change as a result. From these experiments, values were selected to carry out the industry relevant simulations. Sample size was selected to by 500, as a results of industry constraints. An appropriate similarity distance threshold for all measures was determined to be 0.4. And the minimum cluster size was selected to be 3 Tweets per cluster.

The 500 Tweet search simulation was run using the selected clustering parameters. It was found that some dataset clustered better than others. Notably, Vancouver was easily clusterable and frequently was reduced to 60% of the original set, whereas it was common to not find any clusters in the RoyalWedding dataset for a sample of 500 Tweets.

The real-time data stream clustering simulation was also run using the same parameters for the seed dataset. Similar clustering characteristics were observed for the streaming simulation. It was also determined that the experimental setup may provide a search dependent, effective method for managing real-time streaming content to reduce

some data and there may be performance improvements for larger seed data sample sizes.

Overall, both thresholding algorithms were found to be effective techniques for filtering content rich social media in an industry context. There were, however, practical differences between each implementation that could promote one over the other depending on use case. The ITWEC thresholding algorithm was found to generally provided higher cluster quality due its feedback loop of Tweets that did not meet the minimum cluster requirements. The ITWEC algorithm, however, operated at a higher computational complexity that was a significant drawback for the searches cases where few clusters were found. The modified clustering algorithm, by comparison, found lower quality clusters but at a far greater rate. It was also seen that the quality disparity between the two algorithms was not significant enough to justify the added complexity. For an industry application with the goal of reducing the perceived amount of redundant content in a social media search, the modified thresholding algorithm provides more than enough capability at the complexity that can be managed for small samples.

For similarity measures, it was also found that each distance measure could provided adequate results with different advantages for each. T-Information provided a fast solution that produced good clusters. The continuous, character-wise nature of the measure allowed it to cluster content at low thresholds for short Tweets as well as more standard-length Tweets. Jaccard string token similarity was equally fast as the T-Information but suffered clustering performance especially for short tweets. Levenshtein string token similarity was computationally expensive and so provided slow but good clustering in most cases. As a string token-based measure, however, it did not perform for small Tweets. Ultimately, for an industry application T-Information provides the best solution as it had the best performance for small Tweets, similar performance to the Levenshtein distance greater speed for larger Tweets, and better clustering performance at similar speeds to Jaccard.

5.1.2 Limitations

The limitations for this methodology largely come from the sample size and scope of the experiment, the parameter selection, similarity distance normalization, and data biases.

Firstly, the sample sizes and the scope of runs used in the experiments were limited by the computational complexity of the algorithms and the performance of the computing machine used for the simulation. Several experiments had single runs that took many hours to complete and would not scale to larger systems with out a significant increase in time and memory requirements. Despite having access to a large data set, only a relatively small subsample of the content was used in any one experiment. To improve upon this, a larger computer cluster or parallel testing system could be used increase results consistency by either running additional samples of the experiment or larger sample sizes.

By definition, the similarity distances measure denoted distinct measures spaces and do not represent the same value. Consequently, each threshold parameter and absolute distance measure meant something different and it could not be directly normalized. Practically speaking this meant a given threshold, for example 0.4, could be more or less permissive when applied to each different similarity measure. This could be improved on by using a ground truth data set to measure against and then developing a normalize threshold values based on the clustering results of the ground truth data set.

The algorithm parameters threshold, minimum cluster, and sample size were selected based on several experiments to evaluate appropriate values. It is possible that better thresholds could selected for each search data set that could be more or less permissive and allowed for different clusters to form. This could be accomplished by clustering very large sample sizes for small thresholds increments and evaluating the results for the best cluster quality. However, the process would be computationally expensive and subject to the training data set. With real-world content, data will change over time and the trained threshold values will need to adapt, rendering the expense

moot. Additional experimentation or information theory may be applicable to find more appropriate values for the clustering methods.

Lastly, the data itself can provide limitations to assessment methodologies due to inherent biases. A simple characterization was used to understand potential biases the data might have, including hashtag, term length, and composition. Further characterization of the data set could be carried out to understand other potential biases better. Finally, the dataset was restricted to 140 characters where Twitter now allows for 280 characters for each publication. As the Echosec platform only allowed for the inclusion of 140 characters during the time of data collection, the disparity may have injected bias resulting from longer Tweets only containing partial messages.

5.2 Future Work

In the future, larger scale testing of these methods should be undertaken. The confidence of the results could be improved on by increasing the data sample size, the increments of variation, and the number of runs used per experiment. Based on the results for some searches there seems to exist only a small amount of clusterable content at small sample sizes. It was also observed the number of clusters and content in each cluster increased as sample size increased. It would be highly beneficial to design and run a significantly large experiment to determine the extent of this relationship. In addition, it would be beneficial to conduct a thorough exploration of the clusters across all simulations to understand the degree to which the largest clusters contain the same Tweets.

Alternative clustering algorithms could be explored for improving intra-set clustering effectiveness. The thresholding algorithm implemented has high computational complexity, whereas new, alternate methods may be able to improve upon it. Specifically, I-TWEC uses a suffix tree implementation that can be carried out in near O(N) time that could be explored.

Additional experimentation into other social media types could provide interesting results. Twitter is a great industry standard for its ubiquity and accessibility, however, other social media platforms are equally important and may have different clustering characteristics. Further research could be carried out to cluster disparate types of social media as well.

Finally, additional research into T-Information as a similarity measure could be carried out. Specifically, T-Information's properties could be analyzed to understand how it is influenced by, or not influenced by, various languages, left-to-rightness, emoji's and other common social media characteristics.

5.3 Implications

There are several direct implications of this research for applications in industry. Firstly, it showed there exists effective methods for intra-set clustering of social media content. This thesis also showed that some searches are more clusterable than others and a classification of search type may provided additional context for clustering. It also determined that there is a trade-off between cluster quality and computational complexity, however, a small loss in quality can result in a significant reduction in the practical computational complexity. T-Information was also determined to be an excellent technique for clustering social media content. Ultimately, industry applications implementing noise reduction features through intra-set clustering would be well advised to review the qualities of a modified thresholding algorithm implementing T-Information as a similarity distance measure.

5.4 Chapter Summary

This thesis evaluated various intra-set clustering techniques for social media content in an industry relevant context. It was determined that the techniques evaluated could effectively cluster social media content and reduce the amount of noise an end user would have to review to make a decision. The ITWEC thresholding algorithm was seen to produce the best clusters at the expense of computational complexity. The modified thresholding algorithm worked quickly and its cluster quality was slightly worse than the ITWEC algorithm. T-Information and Levenshtein similarity distance measures were found to perform the best on clustering, however, T-Information was considerably faster. The primary limitations with this research were the small sample sizes used for clustering, the lack of a normalized information space, and the selection of parameters for the threshold algorithms. There remains a significant amount of research to be carried out carry these findings forward. This research can easily be adopted in industry to reduce the noise present in social media applications. Ultimately, this thesis concludes there are a number of effective techniques for intra-set clustering social media content, however, an implementation of T-Information similarity measure and the modified clustering algorithm could provide an effective solution that delivers good clustering and high performance.

6 Appendix A

Appendix A includes Word Cloud validations for Levenshtein distance and Jaccard distance. As the thresholding algorithms produce qualitatively indistinguishable wordclouds, only one set is presented.

Figure 6.1: RoyalWedding Levenshtein WordCloud Unclustered

Figure 6.2: RoyalWedding Levenshtein WordCloud Largest Cluster

Figure 6.3: Jaccard WordCloud Unclustered

Figure 6.4: Jaccard WordCloud Largest Cluster

Figure 6.5: WorldCup Levenshtein WordCloud Unclustered

Figure 6.6: WorldCup Levenshtein WordCloud Largest Cluster

Figure 6.7: WorldCup Jaccard WordCloud Unclustered

Figure 6.8: WorldCup Jaccard WordCloud Largest Cluster

Figure 6.9: Levenshtein WordCloud Largest Cluster

Figure 6.10: Levenshtein WordCloud Unclustered

Figure 6.11: London Jaccard WordCloud Unclustered

Figure 6.12: London Jaccard WordCloud Largest Cluster

Figure 6.13: Vancouver Levenshtein WordCloud

Figure 6.14: Vancouver Levenshtein WordCloud Unclustered

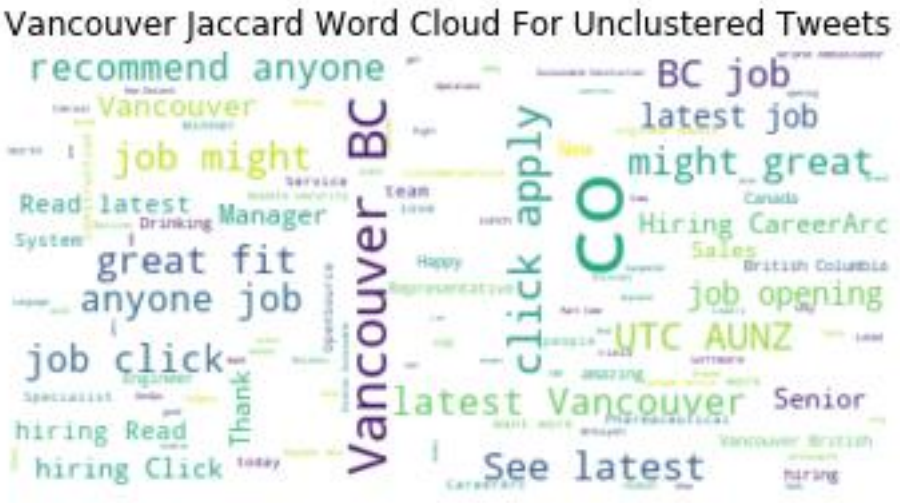

Figure 6.15: Vancouver Jaccard WordCloud Unclustered

Figure 6.16: Vancouver Jaccard WordCloud Largest Cluster

7 Bibliography

[1] "Top 20 Facebook Statistics - Updated April 2018," *Zephoria Inc.*, 30-Apr-2018. .

[2] "Echosec," *Echosec*. [Online]. Available: https://www.echosec.net/. [Accessed: 28-Sep-2018].

[3] "Social Media Usage: 2005-2015 | Pew Research Center," 08-Oct-2015. .

[4] "Global Social Media Statistics Summary 2017," *Smart Insights*, 27-Apr-2017. [Online]. Available: https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/. [Accessed: 07-Jan-2018].

[5] H. Margetts, "Why Social Media May Have Won the 2017 General Election," *Polit. Q.*, vol. 88, no. 3, pp. 386–390, Jul. 2017.

[6] P. T. Metaxas and E. Mustafaraj, "Social Media and the Elections," *Science*, vol. 338, no. 6106, pp. 472–473, Oct. 2012.

[7] "Best Social Media Monitoring Software in 2018 | G2 Crowd." [Online]. Available: https://www.g2crowd.com/categories/social-media-monitoring. [Accessed: 28-Sep-2018].

[8] W. W. Moe and D. A. Schweidel, "Opportunities for Innovation in Social Media Analytics," *J. Prod. Innov. Manag.*, vol. 34, no. 5, pp. 697–702, Sep. 2017.

[9] "Dataminr | Real-Time Information Discovery | Dataminr." [Online]. Available: https://www.dataminr.com/. [Accessed: 28-Sep-2018].

[10] "Resources | Dataminr." [Online]. Available: https://www.dataminr.com/resources?tag=Case+Study. [Accessed: 28-Sep-2018].

[11] H. M. Inc, "Social Media Marketing & Management Dashboard," *Hootsuite*. [Online]. Available: https://hootsuite.com/. [Accessed: 28-Sep-2018].

[12] "Sysomos | Social Media Management and Analytics Software." [Online]. Available: https://sysomos.com/. [Accessed: 28-Sep-2018].

[13] F. A. Pozzi, E. Fersini, E. Messina, and B. Liu, *Sentiment Analysis in Social Networks*. Morgan Kaufmann, 2016.

[14] U. Farooq, T. P. Dhamala, A. Nongaillard, Y. Ouzrout, and M. A. Qadir, "A word sense disambiguation method for feature level sentiment analysis," in *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 2015, pp. 1–8.

[15] R. A., A. J., and C. J. M. Tauro, "A Novel, Generalized Recommender System for Social

Media Using the Collaborative-filtering Technique," *SIGSOFT Softw Eng Notes*, vol. 39, no. 3, pp. 1–4, Jun. 2014.

[16] S. M. Kywe, T.-A. Hoang, E.-P. Lim, and F. Zhu, "On Recommending Hashtags in Twitter Networks," in *Social Informatics*, 2012, pp. 337–350.

[17] İ. Arın, M. K. Erpam, and Y. Saygın, "I-TWEC: Interactive clustering tool for Twitter," *Expert Syst. Appl.*, vol. 96, pp. 1–13, Apr. 2018.

[18] "Tweeting Made Easier." [Online]. Available: https://blog.twitter.com/official/en_us/topics/product/2017/tweetingmadeeasier.html. [Accessed: 13-May-2018].

[19] "Use cases." [Online]. Available: https://developer.twitter.com/en/use-cases.html. [Accessed: 22-Oct-2018].

[20] "Products Overview." [Online]. Available: https://developer.twitter.com/en/products/products-overview.html. [Accessed: 02-Oct-2018].

[21] "Enterprise search." [Online]. Available: https://developer.twitter.com/en/docs/tweets/search/overview/enterprise.html. [Accessed: 02-Oct-2018].

[22] S. Thaiprayoon, A. Kongthon, P. Palingoon, and C. Haruechaiyasak, "Search result clustering for Thai Twitter based on Suffix Tree Clustering," in *2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2012, pp. 1–4.

[23] "Things every developer should know." [Online]. Available: https://developer.twitter.com/en/docs/basics/things-every-developer-should-know.html. [Accessed: 02-Oct-2018].

[24] "Twitter Bots Boosted Donald Trump's Votes by 3.23%: Study | Time." [Online]. Available: http://time.com/5286013/twitter-bots-donald-trump-votes/. [Accessed: 22-May-2018].

[25] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg?," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 6, pp. 811–824, Nov. 2012.

[26] "Social Media Management Platform | Buffer." [Online]. Available: https://buffer.com/. [Accessed: 28-Sep-2018].

[27] "Developer Agreement." [Online]. Available: https://developer.twitter.com/en/developer-terms/agreement.html. [Accessed: 02-Oct-2018].

[28] K. Lee, "What analyzing 1 million tweets taught us," *The Next Web*, 03-Nov-2015. [Online]. Available: https://thenextweb.com/socialmedia/2015/11/03/what-analyzing-1-million-

tweets-taught-us/. [Accessed: 13-May-2018].

[29] E. S. Ristad and P. N. Yianilos, “Learning string-edit distance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 5, pp. 522–532, May 1998.

[30] A. Bookstein, V. A. Kulyukin, and T. Raita, “Generalized Hamming Distance,” *Inf. Retr.*, vol. 5, no. 4, pp. 353–375, Oct. 2002.

[31] G. A. Stephen, *String searching algorithms*. Singapore: World Scientific, 1994.

[32] J. Yang and U. Speidel, “String parsing-based similarity detection,” in *IEEE Information Theory Workshop, 2005.*, 2005, pp. 5 pp.-.

[33] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2012.

[34] M. U. S. Shameem and R. Ferdous, “An efficient k-means algorithm integrated with Jaccard distance measure for document clustering,” in *2009 First Asian Himalayas International Conference on Internet*, 2009, pp. 1–6.

[35] A. Z. Broder, “Identifying and Filtering Near-Duplicate Documents,” in *Combinatorial Pattern Matching*, vol. 1848, R. Giancarlo and D. Sankoff, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–10.

[36] H. Ahmed, M. A. Razzaq, and A. M. Qamar, “Prediction of popular tweets using Similarity Learning,” in *2013 IEEE 9th International Conference on Emerging Technologies (ICET)*, 2013, pp. 1–6.

[37] B. Sriram, D. Fuhry, E. Demir, and H. Ferhatosmanoglu, "Short Text Classification in Twitter to Improve Information Filtering." *SIGIR* (2010).

[38] M. K. Erpam, “Tweets on a tree: Index-based clustering of tweets,” Sabanci University, Apr. 2017.

[39] S. Poomagal, P. Visalakshi, and T. Hamsapriya, “A novel method for clustering tweets in Twitter,” *Int. J. Web Based Communities*, vol. 11, no. 2, pp. 170–187, Jan. 2015.

[40] Y. Fang, H. Zhang, Y. Ye, and X. Li, “Detecting hot topics from Twitter: A multiview approach,” *J. Inf. Sci.*, vol. 40, no. 5, pp. 578–593, Oct. 2014.

[41] M. R. Titchener, “Generalised T-codes: extended construction algorithm for self-synchronising codes,” *IEE Proc. - Commun.*, vol. 143, no. 3, pp. 122–128, Jun. 1996.

[42] A. Lempel and J. Ziv, “On the Complexity of Finite Sequences,” *IEEE Trans. Inf. Theory*, vol. 22, no. 1, pp. 75–81, Jan. 1976.

[43] M. R. Titchener, “Digital encoding by means of new T-codes to provide improved data

synchronisation and message integrity," *IEE Proc. E - Comput. Digit. Tech.*, vol. 131, no. 4, pp. 151–153, Jul. 1984.

[44] N. Rebenich, U. Speidel, S. W. Neville, and T. A. Gulliver, "FLOTT - A Fast, Low Memory T-TransformAlgorithm for Measuring String Complexity," *IEEE Trans. Comput.*, vol. 63, no. 4, pp. 917–926, Apr. 2014.

[45] N. Rebenich, T. A. Gulliver, and S. W. Neville, "Counting prime polynomials and measuring complexity and similarity of information," University of Victoria, Victoria, British Columbia, 2016.

[46] H. Tu and J. Ding, "An Efficient Clustering Algorithm for Microblogging Hot Topic Detection," in *2012 International Conference on Computer Science and Service System*, 2012, pp. 738–741.

[47] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.

[48] R. S. Perdana, T. H. Muliawati, and R. Alexandro, "BOT SPAMMER DETECTION IN TWITTER USING TWEET SIMILARITY AND TIME INTERVAL ENTROPY," *J. Ilmu Komput. Dan Inf.*, vol. 8, no. 1, pp. 19–25, Mar. 2015.

[49] "13.1. csv — CSV File Reading and Writing — Python 2.7.15 documentation." [Online]. Available: https://docs.python.org/2/library/csv.html. [Accessed: 03-Oct-2018].

[50] "5. Data Structures — Python 3.7.1rc1 documentation." [Online]. Available: https://docs.python.org/3/tutorial/datastructures.html. [Accessed: 03-Oct-2018].

[51] "Python Data Analysis Library — pandas: Python Data Analysis Library." [Online]. Available: https://pandas.pydata.org/. [Accessed: 03-Oct-2018].

[52] "MySQL." [Online]. Available: https://www.mysql.com/. [Accessed: 03-Oct-2018].

[53] "Docker," *Docker*. [Online]. Available: https://www.docker.com/. [Accessed: 03-Oct-2018].

[54] "Project Jupyter." [Online]. Available: http://www.jupyter.org. [Accessed: 04-Oct-2018].

[55] "Matplotlib: Python plotting — Matplotlib 3.0.0 documentation." [Online]. Available: https://matplotlib.org/. [Accessed: 04-Oct-2018].

[56] A. Mueller, *A little word cloud generator in Python. Contribute to amueller/word_cloud development by creating an account on GitHub*. 2018.

[57] "Distance · PyPI." [Online]. Available: https://pypi.org/project/Distance/. [Accessed: 04-Oct-2018].

[58] “GitHub - ardeego/libflott: A linear time and space string complexity library.” [Online]. Available: https://github.com/ardeego/libflott. [Accessed: 04-Oct-2018].

[59] “GitHub - mike-anderson/libflott-python: c-python bindings for nti_dist and ntc_dist of libflott.” [Online]. Available: https://github.com/mike-anderson/libflott-python. [Accessed: 04-Oct-2018].

[60] “Google Translate.” [Online]. Available: https://translate.google.com/. [Accessed: 04-Oct-2018].