Intra-Topic Clustering for Social Media

by

Uttej Reddy Gondhi

B.Tech, Jawaharlal Nehru Technological University, 2016

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

Supervisory Committee

Intra-Topic Clustering for Social Media

by

Uttej Reddy Gondhi

B.Tech, Jawaharlal Nehru Technological University, 2016

Supervisory Committee

Dr. Stephen W. Neville, Department of Electrical and Computer Engineering

Supervisor

Dr. Michael L. McGuire, Department of Electrical and Computer Engineering

Departmental Member

Abstract

With the social media platforms leading the internet in terms of user base and the average time spent, significant amount of data is being generated by these platforms every day. This makes social media platforms a go-to place to understand the reviews, trends, and opinions of the people. Any regular search for a popular topic would result in an abundance of information and thus it is impossible to go through these large amounts of data manually to understand the trends.

This thesis discusses techniques for the intra-topic clustering of such social media data and discusses how social media noise increases the redundancy of the search results. Our goal is to filter the amount of redundant information an end-user must review from a regular social media search. The research proposes clustering models based on two string similarity measures Jaccard word token and T-Information distance. Evaluation parameters are introduced and the models are evaluated on clustering a set of current and historical topics to determine which techniques are the most effective.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank:

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

my supervisor **Dr. Stephen Neville**, for his constant guidance, support, and motivation for my project and throughout my program.

Dedication

Dedicated to my dad ***Mr. Venkat Ram Reddy***, my mom ***Mrs. Vani Reddy***, my brother ***Mr. Ujwal Reddy*** and ***Choco*** for their endless support, love, motivation, and guidance.

Also dedicated to the all the front-line workers combating COVID-19.

Chapter 1 Introduction

This chapter reviews the need for effective social media clustering tools. The problem statement and the objective will be introduced. The context of the problem will be reviewed with a brief discussion of the current industry practices and tools.

1.1 Problem Statement

Internet expansion has led to rapid growth in the social media industry. According to [1], people spent about 135 minutes a day on average, browsing and interacting on social media in 2017. Given the scale of social media data, there is a high commercial demand for data clustering tools, techniques, and methods to understand social media data. Social media data is being generated on scales infeasible for manual inspection with more people joining the platform. Social media data is not only being used on a personal level but also on an industry level to understand the conversations and people behind them. This thesis aims to present an approach to cluster redundant data on social media platforms.

1.2 Social Media Background

This section discusses the evolution, trend of internet usage, impact, and the reach of social media platforms on the users and events around the world.

1.2.1 Evolution of social media:

Social media began initially as a platform to communicate among users. It has become one of the most influential platforms in our day to day life including creating jobs across the marketing sector. Social media is a growing phenomenon and has become a medium of communication in the modern age. The trend of social media usage is shown in the Figure 1.1. The data shows the plot of percentage of US adults using at least one social media platform in a year. As of 2019, more than 72% of adults, aged 18 and above in the US use at least one social media site, and these numbers are growing every year with more contribution from young adults as seen from the Figure 1.2 [2]. As per Pew Research Center's demographic stats, it is evident that young adults were the dominant adopters and users of social media and the usage by older adults has increased in recent years [2].

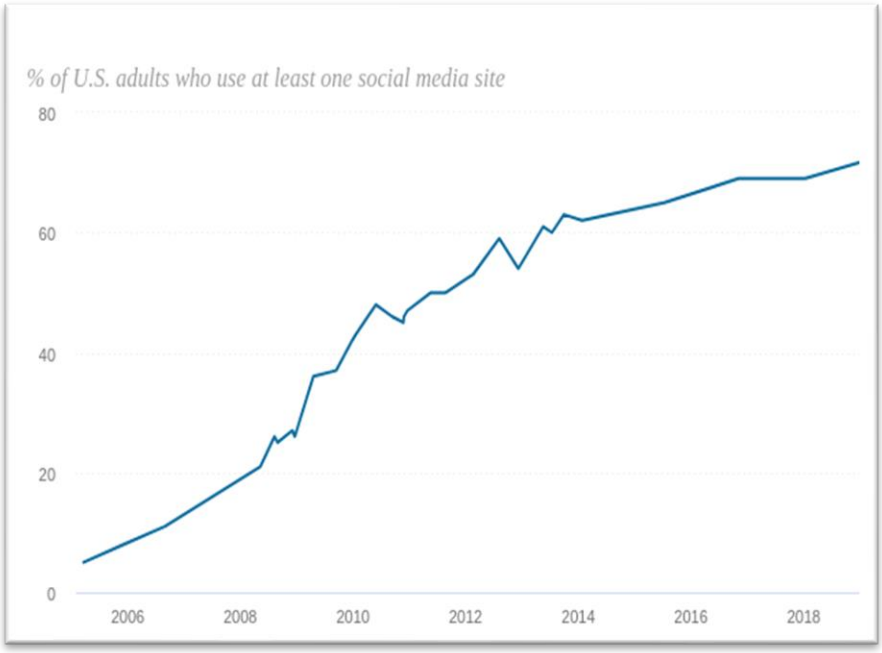

Figure 1.1 Percentage of US adults on social media by the time [2]

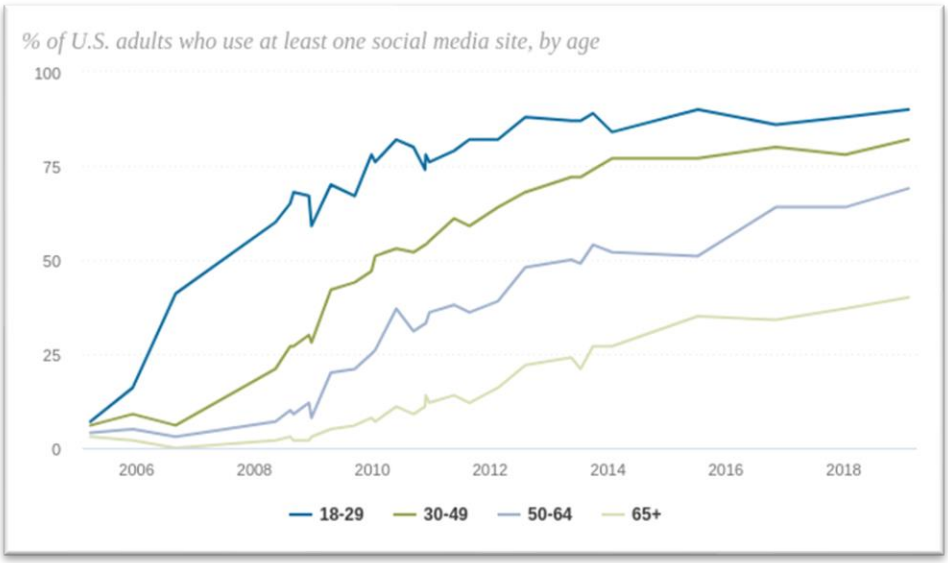

Figure 1.2 Demographics of US adults on social media [2]

1.2.2 Impact and Reach

Activism is coined as, “Taking action to effect social change which can occur in a myriad of ways and in a variety of forms. Often it is concerned with ‘how to change the world’ through social, political, economic or environmental change. This can be led by individuals but is often done collectively through social movements”. [3]

A snippet of hashtag activism on Twitter is shown in the Figure 1.3. Lately, activism is being incorporated in social media to reach a large audience and to spread awareness. One of the key aspects of this is hashtag activism. “Hashtag activism is the act of fighting for or supporting a cause through social media platforms like Facebook, Twitter, Google+, and other networking websites. The term gets its name from the liberal use of hashtags (#) that are often used to spread the word about a cause over Twitter”.[4]

Figure 1.3 A snippet of hashtag(#) activism [5]

Some of the most powerful social movements and their impact are listed below.

1. **#ALSIceBucketChallenge**

   Goal: This hashtag challenge was made to promote consciousness and to raise donations for the research of ALS disease. It involved people pouring a bucket of ice water on themselves, donating to ALS and further nominating people.

   Timeline: Circa 2014

   Impact: The Ice Bucket Challenge was a successful campaign. The ALS association raised over $115M worldwide, a mere 187% increase in annual funding [6].

2. **US Presidential elections**

   Goal: This section talks about the impact of social media on the 2016 US presidential elections. According to the Pew Research Center, social media played a pivotal role in the presidential election. It claimed that over 44% of Americans have admitted to getting their information regarding the 2016 presidential election from social media [7].

   Timeline: 2016

   Figure 1.4 and Figure 1.5 show the tweeting pattern and Twitter usage across the years by two presidential candidates for 2016 US elections, Mr. Donald Trump and Mrs. Hillary Clinton, respectively. Each point on the graph represents a tweet. The horizontal axis denotes the timeline by year and the vertical axis represents the time in hours (EST). We see there is a clear shift in frequency and tweeting patterns of Trump from 2013 and early 2015 for Hillary.

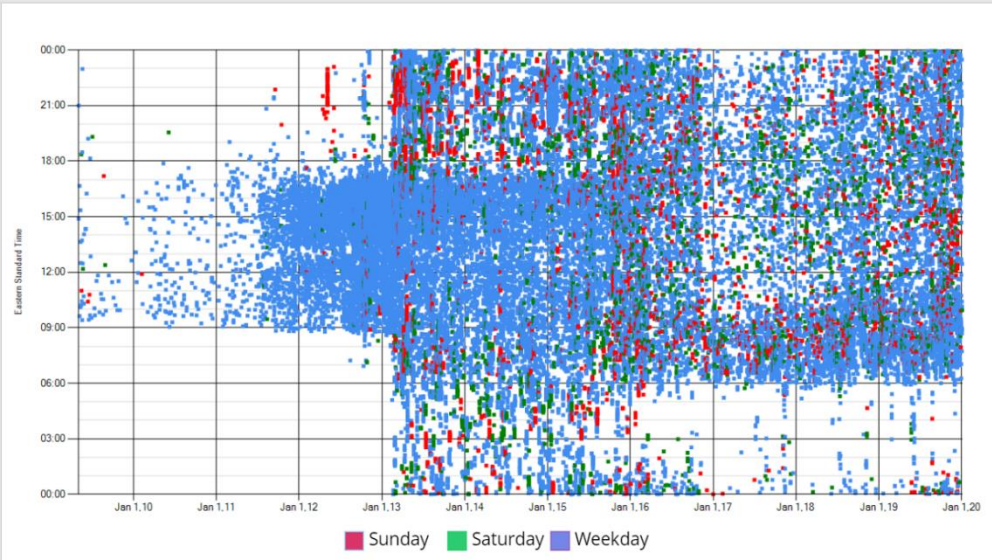

Figure 1.4 Tweeting pattern of Mr. Donald Trump [8]

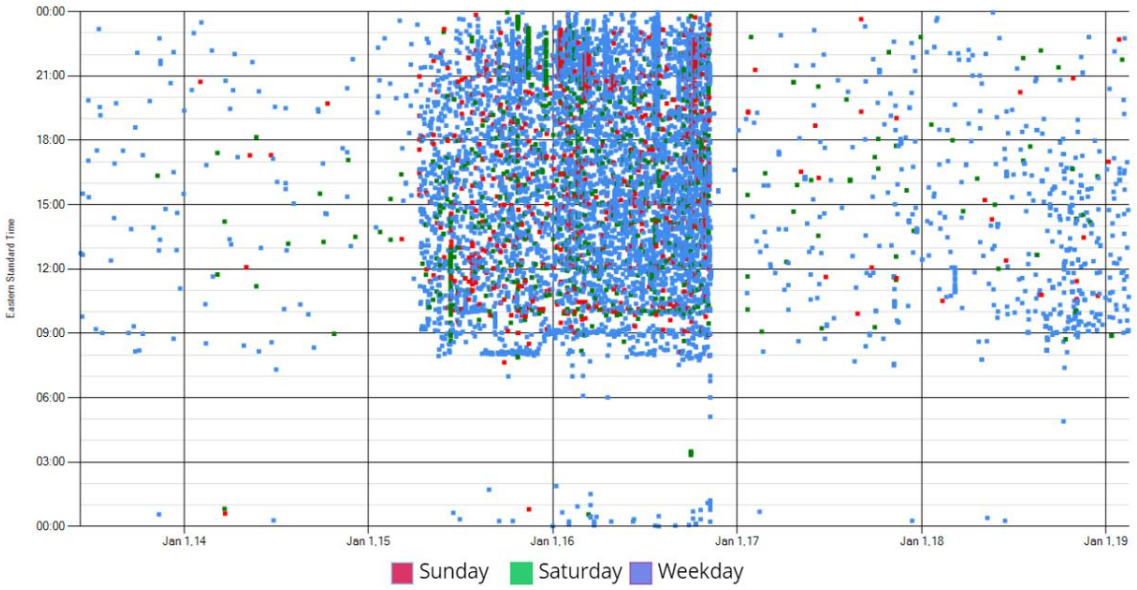

Figure 1.5 Tweeting pattern of Mrs. Hillary Clinton [9]

Frank Speiser, the co-founder of SocialFlow, said, "This is the first true social media election." He added that before the 2016 presidential primaries, social media were a mere "auxiliary method of communication," but in this new era, "folks on social media to act on your behalf by just sharing it around. You don't have to buy access to reach millions of people anymore". [10]

The Guardian compared Internet memes to political cartoons, arguing, "For the first time in a US election cycle, community-generated memes have grown to play a significant role in political discourse, similar to the classic printed cartoon." While an Internet meme is unlikely to destroy a political career, lots of memes targeting a candidate might". [11]

These above snippets show how powerful social media is and what it can achieve. With people spending hours on social media platforms, there is an abundance of data including noise, which will be discussed in the later sections. This calls for the need of tools and methodologies to filter out the data we need.

1.3 Use of Social Media in Industry:

There are multiple companies that work on gathering and analyzing the way people communicate, their interests, and opinions. Some of the major domains and how they use this information is outlined below.

1. **Marketing**: Marketing companies use this information to create campaigns to increase their client's product reach and to target specific audience.

2. **Political campaigns**: Political parties and agencies rely heavily on social media data in designing their political campaign strategies. This enhanced knowledge can influence the reach and can target specific audience groups for their campaigns.
3. **Brands**: Today's brands rely heavily on social media in several ways. Primarily for managing brand reputation online, discovering influencers, local trends and public opinion.
4. **Security**: With globalization, companies now have employees working from across the globe and the organisation can use social media to study about the events happening in a place, and if people or their organization will be impacted [12].
5. **Journalism**: Social media is the fastest way to learn out about events of interest in real-time. One of the existing solutions that aggregate social media data and generates alerts in case of high impact events is DataMinr [13].
6. **Finance**: Financial analysts and insurance agencies, use social data for staying on top of events that may impact the market and to analyze current market trends and, assets.

1.4 Noise in Social Media

Noise in social media is defined as any social media content that does not contribute or distracts the ultimate purpose of the search carried out by a social media user. It is often subjective as it depends on the context of the search. Some primary forms of social media noise are as follows.

Type I: Non-informative results.

An end-user is interested in learning about Apple and its device reviews. The search for keyword "#apple" was carried out on Twitter as shown in the Figure 1.6 and it shows three tweets.

The first tweet talks about the evolution of Apple and in this case, is relevant to the search and is adding value to the user's search, while the second tweet appears to be part of an Amazon's marketing post and in this case, is noise as it does not help the end-user. The last tweet shows a user's critique of Apple's marketing skills this case does not help the end-user and is considered noise.

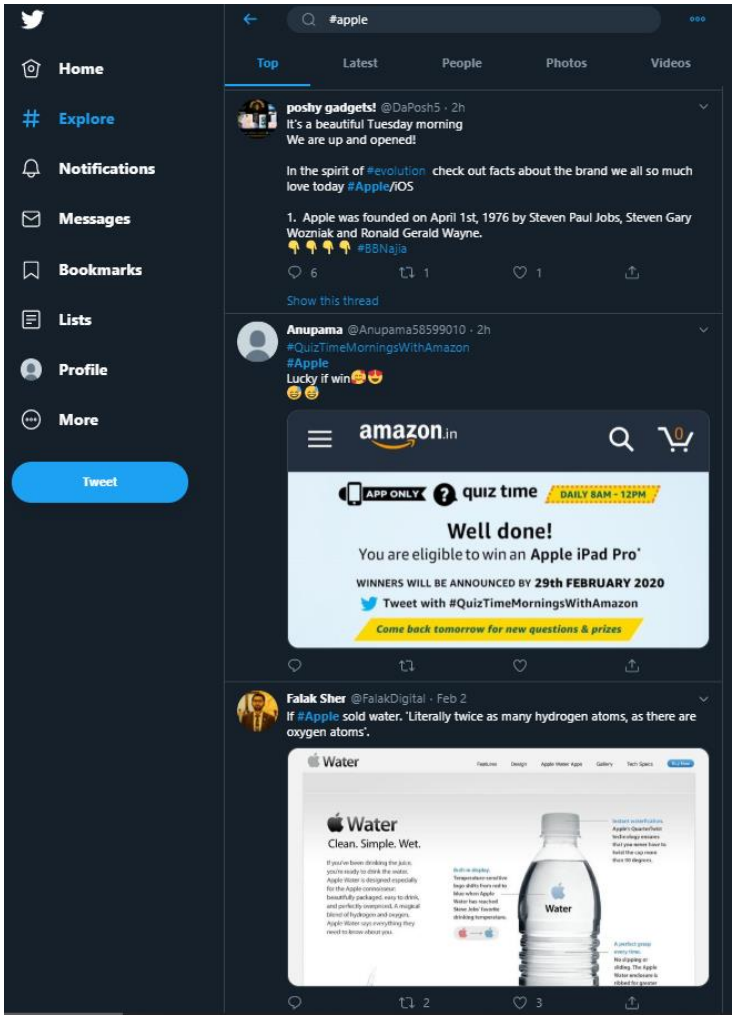

Figure 1.6 Type-I noise[5]

Type II: Different contexts.

An end-user is interested in learning about the Raptors, Canada's basketball team based out of Toronto, Ontario and their progress in the NBA. The search for keyword "#raptors" was carried out on Twitter as shown in the Figure 1.7 and shows two tweets.

The first tweet refers to a photograph of a bird in the family of raptors and in this case, is considered noise. the second tweet rightly refers to the basketball team, Toronto Raptors and hence a meaningful tweet.

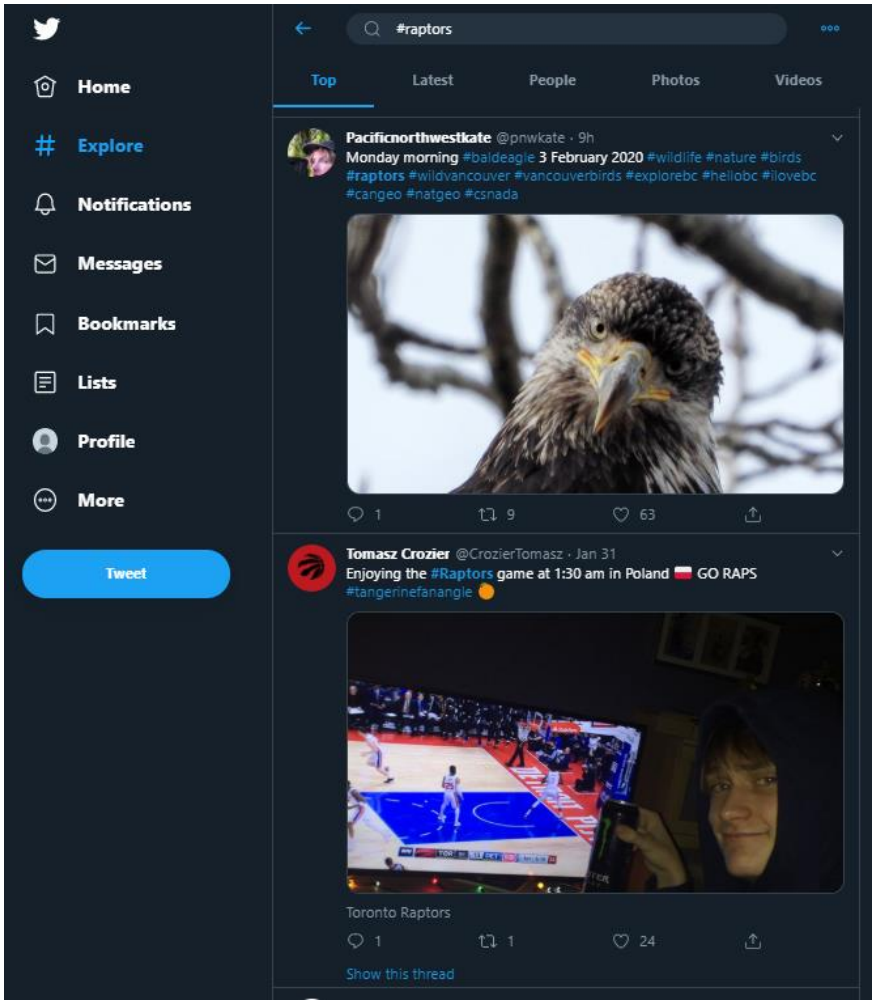

Figure 1.7 Type-II Noise [5]

Type III: Non – relevant content.

The search was performed for the “trump” keyword on Twitter and the result in the Figure 1.8.

The third kind of social media noise are the ones that are irrelevant to the search topic and this case usually happens when the tweet author includes the trending keywords to boost their post’s reach.

Figure 1.8 Type-III Noise [5]

Type IV: Bot generated posts

Figure 1.9 shows a case wherein the tweets generated by bots are shown. This usually includes postings about breaking news, live scores, weather updates, etc.

Figure 1.9 Type-IV Noise [5]

To address the Type-I type of noise, there is no general solution that can be developed since the noise in social media is very subjective and is highly dependent on the user's scope of search. While the Type-II type of noise could be filtered out by using specific keywords or custom searches related to the

search we anticipate which in this case could be using #TorontoRaptors for NBA related search. Examples of complex search parameters is shown in Table 1.1. Type-III type of noise is difficult to filter out as they are usually dynamic and the Type-IV type of noise usually, these posts tend to follow patterns and can be easily filtered out by clustering algorithms. This thesis aims to specifically address Type-III and Type-IV social media noise.

Table 1.1 Complex search examples [14]

Custom String Example	Finds Tweets…
watching now	containing both “watching” and “now”. This is the default operator.
“happy hour”	containing the exact phrase “happy hour”.
love OR hate	containing either “love” or “hate” (or both).
beer -root	containing “beer” but not “root”.
#haiku	containing the hashtag “haiku”.
from:interior	sent from Twitter account “interior”.
list:NASA/astronauts-in-space-now	sent from a Twitter account in the NASA list astronauts-in-space-now
to:NASA	a Tweet authored in reply to Twitter account “NASA”.
@NASA	mentioning Twitter account “NASA”.

1.5 Social media analysis techniques used in Industry

The three most common solutions used across industry and academia for social media analytics are sentiment analysis, recommendation engines, and clustering.

1.5.1 Sentiment Analysis

Sentiment analysis can be coined as “interpretation and classification of emotions within voice and text data using text analysis techniques, allowing businesses to identify customer sentiment toward products, brands or services in online conversations and feedback“ [15].

Sentiment analysis methodologies use natural language processing (NLP) algorithms, with naïve approaches using a pre-defined list of words or expressions. These words are assigned either positive or negative sentiments. For example, words like happy or cheerful are associated with positive sentiment

while the words like angry or awful are tagged with negative sentiment. The system counts the occurrence of these words and based on which set of words dominate in the given text, it returns the corresponding sentiment and returns neutral when there is an equal number of positive and negative sets of words [15]. Stop words are the list of words that are commonly used, and which do not add any meaning. Stop words are often deemed as neutral to prevent bias. Examples of stop words are articles. Although these systems are easy to implement, they lack understanding of human emotions and do not understand irony and sarcasm.

1.5.2 Recommender Systems

Recommender systems work to serve relevant, interesting, and engaging content to social media users. In general terms, these systems are either content-based or collaborative filtering [16]. The former model makes recommendations based on the user's profile characteristics while the latter model makes recommendations based on the user's preferences that are calculated by analyzing the user interactions with the content, which is usually tracked by either tags or metadata.

1.5.3 Clustering Algorithms

As discussed in Section 1.2 with the evolving social media industry, the volume of social media data is becoming challenging to manage, clustering methodologies are one of the key areas of interest across the industry and academia [17]. Noise in the social media industry is also increasing with the volume of the data and this calls for a method that can effectively cluster the social media data and help remove the redundant data.

1.6 Twitter

Twitter began in 2006 and is one of the leading social media platforms. Twitter's content is known as tweets and this platform is considered microblogging given the initial character count restriction of 140 characters and later doubled to 280 characters for non-Asian characters [18].

Each social media platform has its own character limits in place, as shown in Table 1.2.

Table 1.2 Social media platforms with character limits [19]

Social Account	Character Limit
Facebook page	5000
Instagram	2200
Twitter	280
Pinterest boards	500

Twitter significance:

Twitter's user base includes the POTUS (President of the United States), celebrities and journalists. Some of the top reasons why Twitter is preferred over other social media platforms are.

- **Character limit of 280**: This is a focus of Twitter to deliver content that is crisp and to the point.
- **The integrity of content**: Tweets once posted cannot be edited, unlike the other social media platforms.
- **No Followers limitation**: Twitter does not have a restriction on the number of followers one can have, unlike the Facebook user profiles which restrict to a maximum of 5000 people.
- **Verified profiles**: Twitter has verified profile programs to indicate the accounts of public interest that are authentic and is indicated by a blue checkmark beside the profile name as shown in the Figure 1.10.

Figure 1.10 Verified profile on a Twitter profile [5]

1.6.1 Twitter APIs

Twitter offers APIs for developers to interact with their data. They offer three categories of APIs as shown in Figure 1.11.

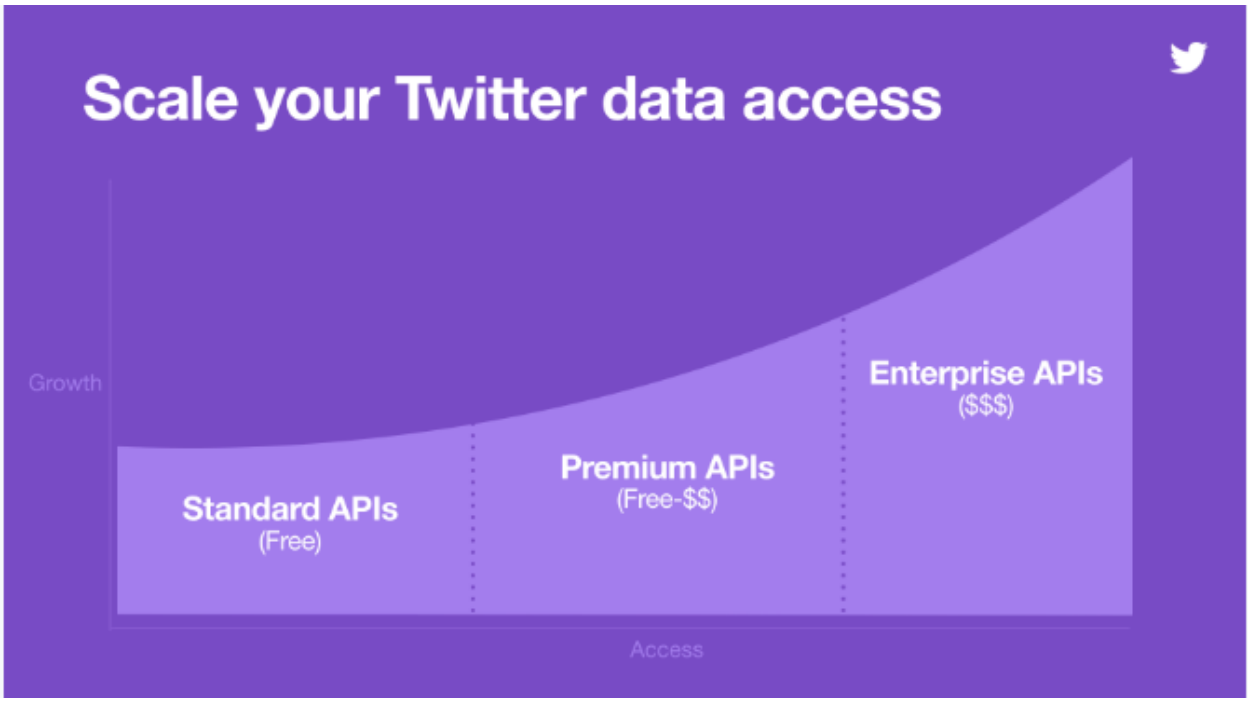

Figure 1.11 Twitter API tiers [20]

The category varies by price, data fidelity, and search capability. The standard APIs are free and limit the number of tweets returned in a time window. These are commonly used across academia due to budget constraints and they can search tweets dating back to seven days from the time query is run. They are part of the 'public' set of APIs.

The premium category API offers two versions that can search for tweets dating back to 30 days or full historical data, while the Enterprise category API gives complete access to historical tweets dating back to the first tweet ever made in 2006. It can handle complex queries and the query length can be as long as 1024 characters. Premium category API's are generally affordable for the industry.

Search API can return from 15 – 500 tweets per request based on type of API category as shown in Figure 1.12 and are rate limited by seconds and minutes based on the API tier being used [21]. Different instances of the same search can and does generate different return results.

Category	Product name	Supported history	Query capability	Counts endpoint	Data fidelity
Standard	Standard Search API	7 days	Standard operators	Not available	Incomplete
Premium	Search Tweets: 30-day endpoint	30 days	Premium operators	Available	Full
Premium	Search Tweets: Full-archive endpoint	Tweets from as early as 2006	Premium operators	Available	Full
Enterprise	30-day Search API	30 days	Premium operators	Included	Full
Enterprise	Full-archive Search API	Tweets from as early as 2006	Premium operators	Included	Full

Figure 1.12 Twitter API tiers capabilities [22]

Each search API request needs authorization header along with a search query of varying complexity based on a category and have per second, per minute rate limitations. The search query has multiple variables it can process, and the search flexibility is based on the API category. A search endpoint for enterprise API query parameters is shown in Table 1.2.

Table 1.3 Enterprise API Query search endpoint toggle parameters [23]

Parameters	Description	Required	Sample Value
query	The equivalent of one PowerTrack rule, with up to 2,048 characters (and no limits on the number of positive and negative clauses).	Yes	(snow OR cold OR blizzard) weather
tag	Tags can be used to segregate rules and their matching data into different logical groups. If a rule tag is provided the rule tag is included in the 'matching_rules' attribute.	No	8HYG54ZGTU
fromDate	The oldest UTC timestamp (back to 3/21/2006 with Full-Archive search) from which the tweets will be provided. The timestamp is in minute granularity	No	201207220000

	and is inclusive (i.e. 12:00 includes the 00 minutes). *Specified:* Using only the fromDate with no toDate parameter will deliver results for the query going back in time from now( ) until the fromDate.		
toDate	The latest, most recent UTC timestamp to which the Tweets will be provided. Timestamp is in minute granularity and is not inclusive (i.e. 11:59 does not include the 59th minute of the hour).	No	201208220000
maxResults	The maximum number of search results to be returned by a request. A number between 10 and the system limit (currently 500). By default, a request response will return 100 results.	No	500
next	This parameter is used to get the next 'page' of results as described HERE. The value used with the parameter is pulled directly from the response provided by the API and should not be modified.	No	NTcxODIyMDMyODMwMjU1MTA0

1.6.2 Sample Twitter API Request

This section shows a sample GET request and response to Twitter's search endpoint [24]. The API response is usually in JSON format and the API calls can be scripted in various programming languages with the most popular choice being Python.

Sample GET Request

```
curl -u<username> "http://gnip-api.Twitter.com/search/:product/accounts/:acco
unt_name/:label.json?query=TwitterDev%20%5C%22search%20api%5C%22&maxResults=5
00&fromDate=<yyyymmddhhmm>&toDate=<yyyymmddhhmm>"
```

SAMPLE GET Response

```
{
    "results":
      [
            {--Tweet 1--},
            {--Tweet 2--},
            ...
            {--Tweet 100--}
      ],
    "next":"NTcxODIyMDMyODMwMjU1MTA0",
    "requestParameters":
      {
        "maxResults":100,
        "fromDate":"202002010000",
        "toDate":"202002200000"
      }
 }
```

1.7 Semi-automated Solutions

With the choice of having an API endpoint, posting social media content by means of a script has made it easy to broadcast information in real-time. However, with more data comes more noise or redundant data in social media.

There are many commercial solutions available in the market, that work with social media APIs to schedule a post/tweet as per the user's convenience. Some of the popular solutions are Hootsuite and Sprout social. These semi-automated solutions contribute to the generation of large amounts of social media data. A solution is indeed needed to reduce the social media noise to focus on high-value data.

Chapter 2 Literature Review

This chapter discusses and reviews some of the common methodologies and filtering approaches for social media data.

2.1 Content similarity:

Tweets contain text (including links, emojis) and may include images with most of the tweets being textual. Hence, we aim to find similarities by analyzing how similar the tweet contents are. There are several similarity measures that are being used in the literature and industry including Jaccard [25], Hamming [26], Levenshtein distances. Some of the approaches to find similar content are bag of words, lexical and semantic similarity, and suffix trees, as described below.

2.1.1 Hamming and Levenshtein distance:

Hamming and Levenshtein distances are the two most used edit distance measures. The edit distance of two strings is defined as the number of characters in a string that needs to be changed to convert it into the second-string [27]. These distance measures are most commonly used in telecommunication, information theory, error-correcting codes, and as a general similarity measure [28].

Hamming distance measure works only if the two strings being compared are of the same length, while the Levenshtein distance measure allows for omissions and insertions. The strings to be compared can be bits, characters, or strings. Both these measures operate on strings from left to right and are affected by the order of tokens in the string. Word tokens are generated by splitting the string into space-separated tokens commonly comprising of words, hashtags, links, etc.

2.1.2 Jaccard similarity:

Mathematically, the Jaccard similarity measure (or the coefficient of similarity) of two strings is defined as the ratio of the number of shared attributes across the strings to the total number of unique attributes in the strings [25]. This measure does not need two strings to be of the same length and the strings can be bits, characters, or sets. As this measure calculates the shared attributes, it does not depend on the order unlike the edit distances (Hamming or Levenshtein distances). The Jaccard coefficient of two strings A and B is denoted mathematically in Equation 2.1.

$$Jaccard_{Coeff}(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.1}$$

Where |.| is the set cardinality operator. The intersection of sets refer to the number of common word tokens in the strings A and B. Likewise the union operator refers to the number of unique word tokens in the strings A and B. Research by Shameem et al. [29] talks about using Jaccard similarity measure for a modified K-means approach to cluster documents, the authors proposed using Jaccard measure to find a set of dissimilar documents to be used as the initial 'K' centres rather than selecting 'K' random documents as in traditional K-means approach. Modified K-means and the traditional algorithm were evaluated, and the former approach saw a mere 42% reduction in the sum of square error value [29]. The paper concludes that by considering dissimilar set as initial centres for a K-means algorithm, its clustering performance increases, and the sum of square error value decreases substantially. The evaluation was performed on documents which usually contain far more information than the tweets.

2.2 Bag of Words Content Similarity:

Bag of words (BoW) is one of the natural language processing (NLP) algorithms used to transform the text into vectors by using vocabulary and corpus, which are in turn used to generate learning models. The BoW vectors can then be compared to determine the relative similarity between the vectors. This approach is usually used to cluster documents and the intuition is that the documents are similar if they have similar content [30]. BoW constructs the vocabulary of unique words from the data set and transforms each document to vectors by representing each document by the measure of the presence of these words. It does not consider the structure or order of the words. Example 2.1 shows a naïve BoW implementation.

Let there be three documents, Document 1, Document 2, and Document3. The documents have the following text.

Example 2.1:

Document1 = "I am the father"

Document2= "I am the mother"

Document3= "How are you doing"

The process begins by constructing the vocabulary of all the unique words from the data set. The data set is the collection of documents. The vocabulary in our data set is shown along with the transformed vector representation of the documents. Each vector shows the measure of the presence of words from the constructed vocabulary.

Vocabulary = ['I' , 'am' , 'the' ,'father' ,'mother', 'How' ,'are' ,'you' ,'doing']

VectorD1 = [1, 1 ,1 ,1 ,0 ,0 ,0, 0, 0]

VectorD2 = [1, 1 ,1 ,0 ,1 ,0 ,0, 0, 0]

VectorD3 = [0, 0,0 ,0 ,0 ,1 ,1, 1, 1]

The research by Sriram et al. [31] presents an 8-feature classifier approach of classifying over 5400 tweets into five topic bins of news, events, opinions, deals and private messages. The paper compares the bag of words model and the 8-feature approach where 8 features are extracted from the tweets. The features considered were authorship metadata, presence of shortening words, slangs, time-

event phrases, opinion words, currency, percentage sign, @username either at the beginning or in the tweet and used a naïve Bayes classifier for classification.

From the conclusions of the research by Sriram et al. [31], the naïve Bayes classification using 8-feature method has outperformed the traditional bag of words approach by over 32% in overall accuracy measure. The time to build the training models were 37.2 sec for BoW and 0.8 seconds for the 8-feature approach [31]. Their research concluded that the traditional BOW does not perform well over tweets as they are usually short-text due to the character limit and, as they are based on emotions, they do not follow standard grammar, may contain short forms, slangs, typos. Using BoW on tweets might need stemming and lemming, increasing the overall time and computations. Stemming is the process of reducing words to their stems while lemming is a way of reducing words to their lemma or to their dictionary form.

2.3 Suffix Tree Clustering (STC):

Suffix tree is a compacted tree that holds the suffixes of a string and is extensively used in measuring string similarity and pattern matching [32]. Research by Santipong et al. [33] introduced STC which is a clustering algorithm based on suffix trees that clusters the documents that share common phrases or the suffix of a phrase. This research performed clustering on 160k tweets of 12 topics related to flash floods in Thailand. Suffix tree clustering was used in conjunction with the label merging approach, tweets were clustered by using Carrot2's STC implementation [33]. Carrot2 is an open-sourced result clustering algorithm using STC to form clusters. To improve the original algorithm in Carrot2, an approach was proposed to merge and create a two-level label structure called Suffix Tree Clustering with Label Merging (STC-LM). This algorithm could merge partially overlapped labels, which can be combined into one label.

Arin et al. developed a web-based tweet clustering tool using lexical and semantic similarities [34]. The data set composed of sixty thousand tweets from four topics, #Trump, #Jesuischarlie, #Christmas (2016) and #NBA. The research compared LCS-Lex (Longest common subsequence based lexical clustering of tweets) to ST-TWEC (Suffix tree-based clustering method). The former method uses a lexical approach and focuses on clustering by subsequence while the latter method combines a semantic approach and uses common sub-strings to cluster the tweets.

2.3.1 T-Information Flott distance:

T-codes, a variable length, prefix-free code introduced by Titchener [35], have been used for various applications including error detection, malware detection, cryptography, data compression, and basic information classification. 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. [17] N. Rebenich et al. [36] developed a fast T-code decomposition FLOTT, which has performance increase over previous implementations of T-codes in both speed and effective memory utilization.

The thesis of Jubinville [17] introduced a baseline of ten string modifications of tweets including omitting, adding, replacing part of the string etc. as shown in Table 2.1. Multiple similarity measures (Hamming, Levenshtein, Jaccard, T-Information distance) were evaluated on their computational performance and relative distance measures and compared against a performance baseline [17].

Table 2.1 List of String modifications [17]

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

This prior thesis also introduced an algorithm to cluster tweets and needed a threshold between [0,1] to determine, how close the tweets must be, to be part of a cluster with 0 indicating the same tweet and 1 indicating dissimilar tweets. This thesis compared baseline results across multiple similarity measures including Hamming, Levenshtein, Jaccard and T-Information distances. Hamming distance was discarded as it needed the strings in comparison to be of the same size and this may rarely occur in tweets. Levenshtein distance was also excluded as it was computationally too expensive.

From the conclusions of research by Jubinville [17], as the character-wise Jaccard distance measures the ratio of the number of common characters in the tweets to the number of unique characters in both the tweets. As a result, it failed to properly distinguish unique tweet, and was discarded from the evaluation. T-information distance measure was proven to be effective while also being robust to minor variations that are common across social media as they are based on human emotions. This research suggested that, two distance measures, T-Information distance and word token Jaccard distance measures performed the best in the baseline tests and were considered for the algorithm [17]. Word tokens are generated by splitting the tweet into space separated tokens commonly comprising of words, hashtags, links etc. The research of this thesis explores how to replace the algorithm of [17] with an improved algorithm that does not depend on a tunable parameter.

Chapter 3 Methodology:

This chapter discusses data acquisition, clustering methodologies, implementation, and performance measures.

3.1 Data Acquisition

The data sets used in this research were obtained during different timeframes and using different platforms. A Twitter public API was used for three data sets about recent events/topics (Corona, Oscar and Wetsuweten), referred to as the current data set and for the two other data sets (London and Seattle), referred to as the historical data sets, for these data sets tweets were obtained using an enterprise-level Twitter API in collaboration with Echosec Inc. [12], a leading social media aggregation company based out of Victoria, BC, Canada.

3.2 Data Composition

This section discusses the data set(s) types and presents the data set format and a sample tweet from both the historical and current topic data sets.

3.2.1 Historical Data set:

These data sets are obtained from Echosec Inc.[12] in Structured Query language (SQL) format and all personally identifiable information (PII) was removed to be compliant with Echosec's terms of service [37].

Each data set has four columns, namely:

1. Post ID
2. Timestamp
3. User ID
4. Tweet content

The Post ID and User ID are randomized IDs generated by the Echosec Platform and hence the tweets cannot be tracked to a specific Twitter user. The timestamp is when the tweet was created and the 'tweet_content' field contains the raw content of the tweet including text, emojis, and URLs. An example of plain-text data schema and associated content is seen in the Figure 3.1 below.

Figure 3.1 Sample tweet from the historical data set

3.2.2 Current Data set:

For the current data set, a free tier Twitter API was used. A Twitter developer account is needed to obtain the keys to authenticate these API requests. The code is built on Python using Tweepy [38] library , to extract tweets from the Twitter's endpoint. The data was obtained in a data frame and was exported to .xlsx(spreadsheet) format and the following two fields are returned for each of the searches.

1. Timestamp
2. Tweet Content

Using the free tier Twitter API, tweets can be retrieved as old as 7 days from the time of API call execution. The data was collected on 17th Feb 2020. The timestamp field indicates the time of tweet creation and the 'tweet_content' field indicates the raw tweet including URLs. A sample tweet is shown in the Figure 3.2.

Time_stamp	Tweet_content
2/16/2020 8:17	@showmekittys Hahahahaha. We've heard that one before! #Wetsuweten #ShutCanadaDown #rcmpgtfo

Figure 3.2 Sample tweet from the current data set

3.3 Exploratory data analysis (EDA)

To better understand the data set(s), several parameters are evaluated including: Wordcloud, number of words per tweet, top 10 frequent words and percentage composition of hashtag (#), and mention (@) usage in tweets by topic. The top 10 frequent word tokens and the Wordcloud are generated after removing the stop words.

3.3.1 London Data Set Analysis

The Figure 3.3 shows the percentage composition of hashtags (#) and mentions (@) in the London data set. The Figure 3.4 shows the number of words per tweet. The Figure 3.5 shows the 10 most frequent words in the data set while the Figure 3.6 shows a Wordcloud for this data set.

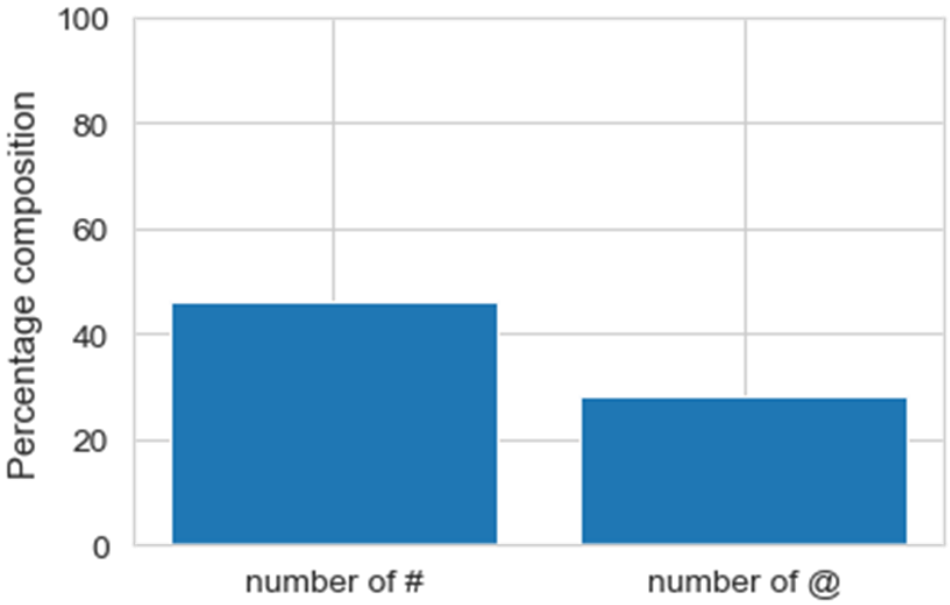

Figure 3.3 Percentage composition of London data set

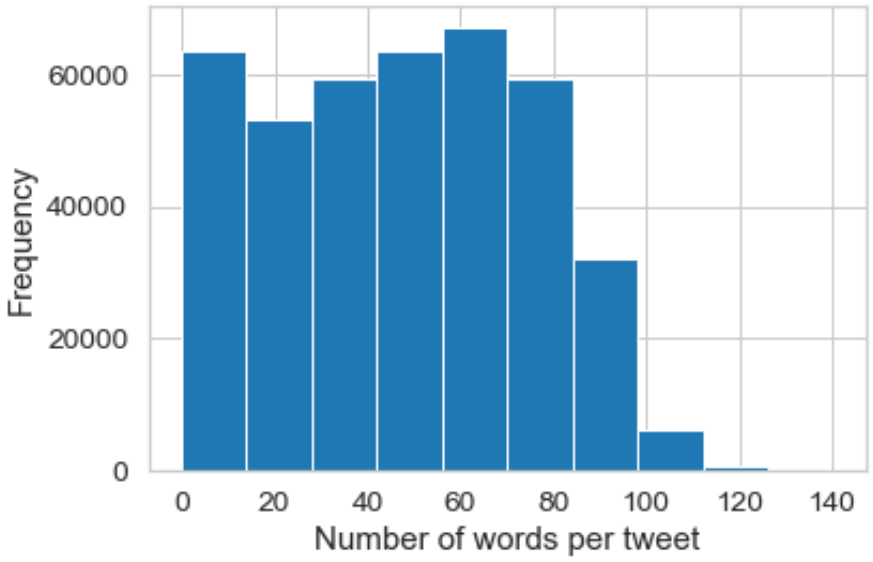

Figure 3.4 Words per tweet – London data set

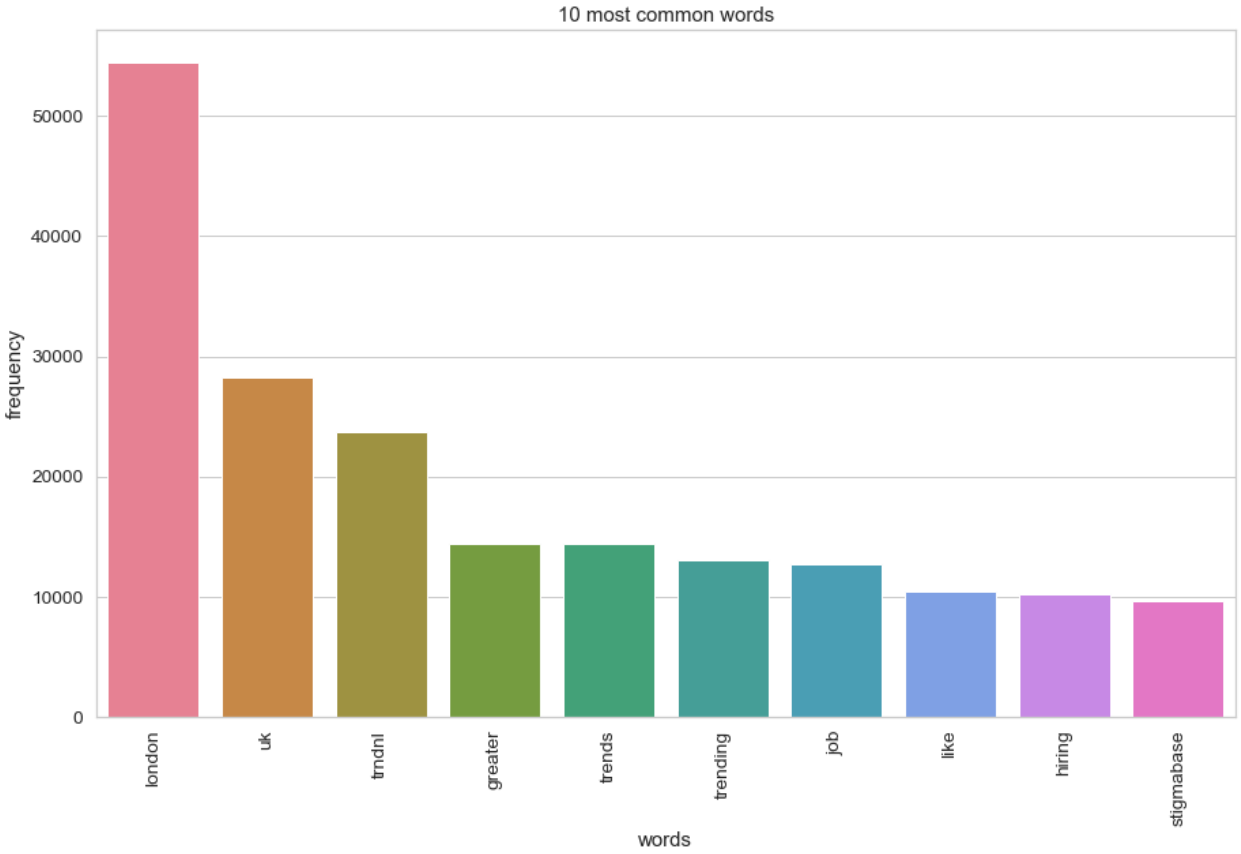

Figure 3.5 10 Frequent words in London data set

Figure 3.6 Wordcloud - London data set

3.3.2 Seattle Data Set Analysis

The Figure 3.7 shows the percentage composition of hashtags (#) and mentions (@) in the Seattle data set. The Figure 3.8 shows the number of words per tweet. The Figure 3.9 shows the 10 most frequent words in the data set while the Figure 3.10 shows a Wordcloud for this data set.

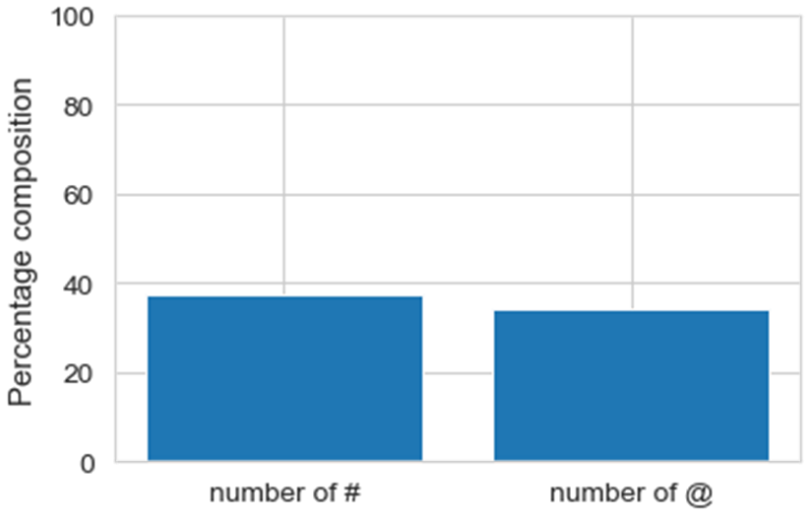

Figure 3.7 Percentage composition of Seattle data set

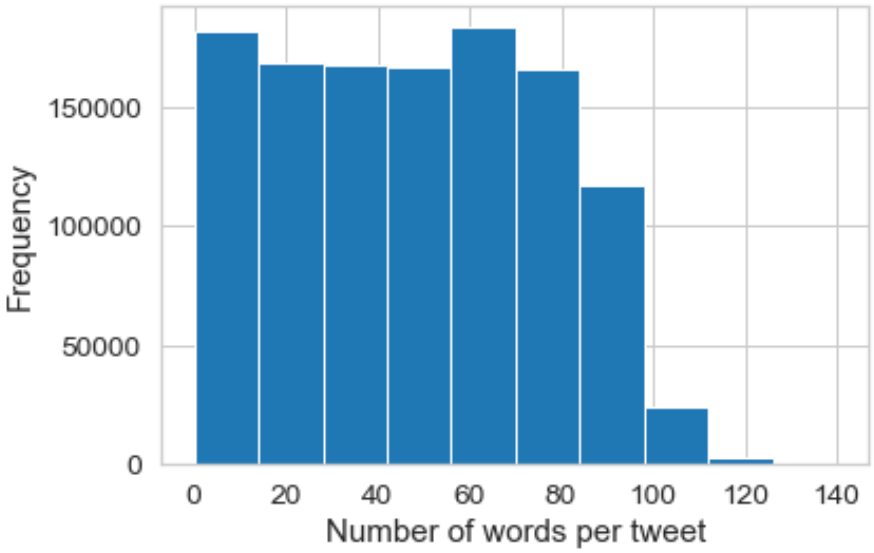

Figure 3.8 Word tokens per tweet – Seattle data set

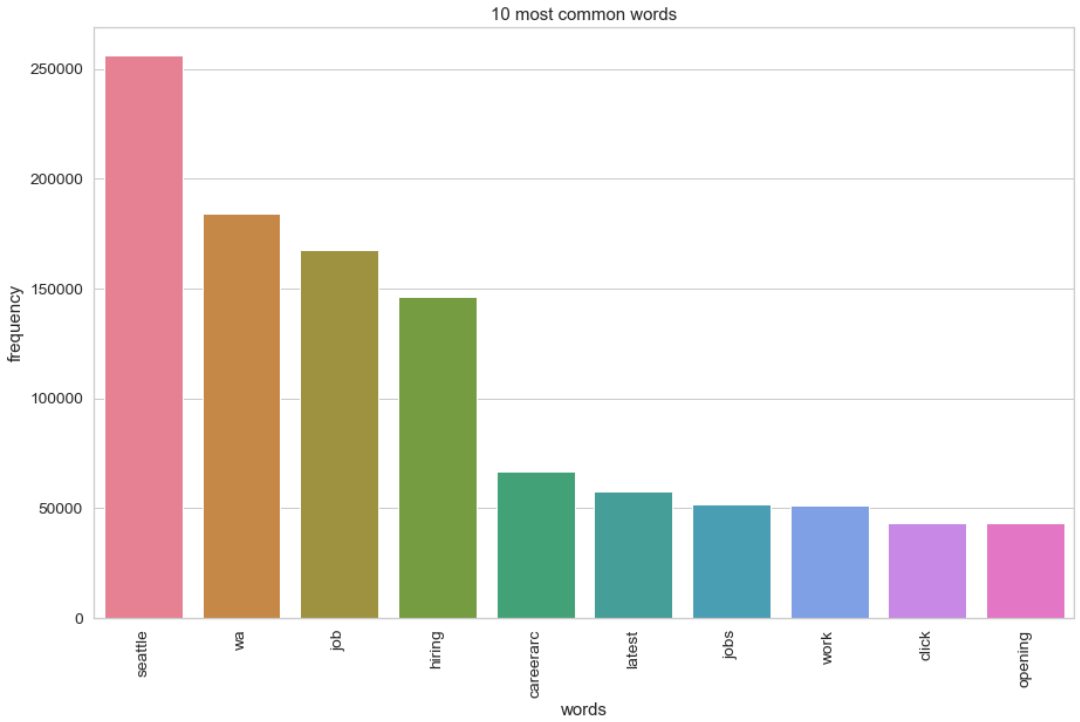

Figure 3.9 10 Frequent words in Seattle data set

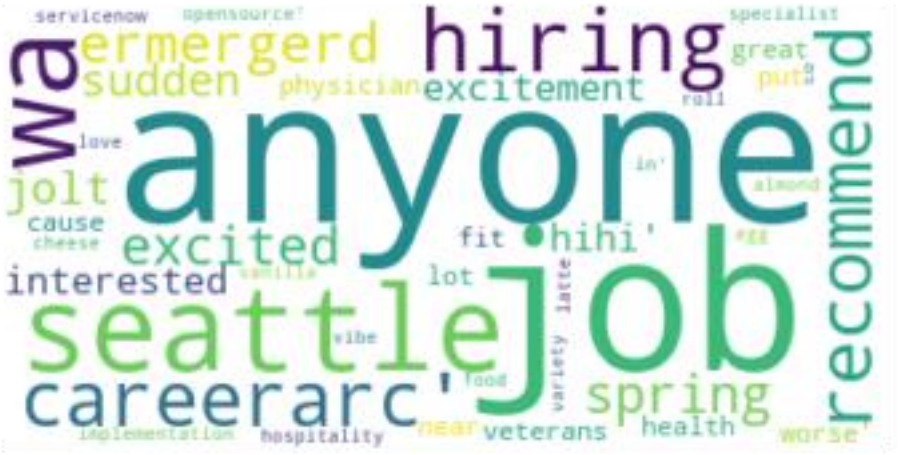

Figure 3.10 Wordcloud - Seattle data set

3.3.3 Oscar Data Set Analysis

The Figure 3.11 shows the percentage composition of hashtags (#) and mentions (@) in the Oscar data set. The Figure 3.12 shows the number of words per tweet. The Figure 3.13 shows the 10 most frequent words in the data set while the Figure 3.14 shows a Wordcloud for this data set.

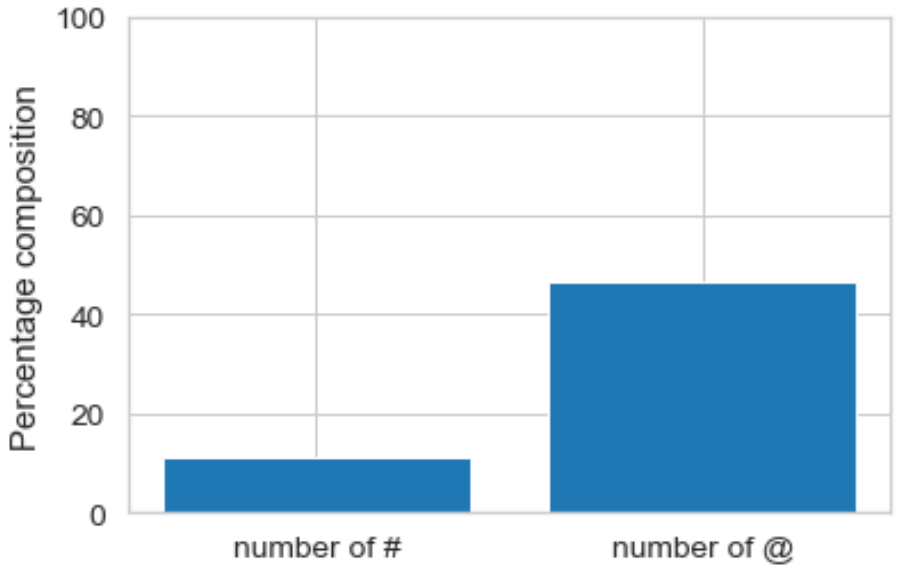

Figure 3.11 Percentage composition of Oscar data set

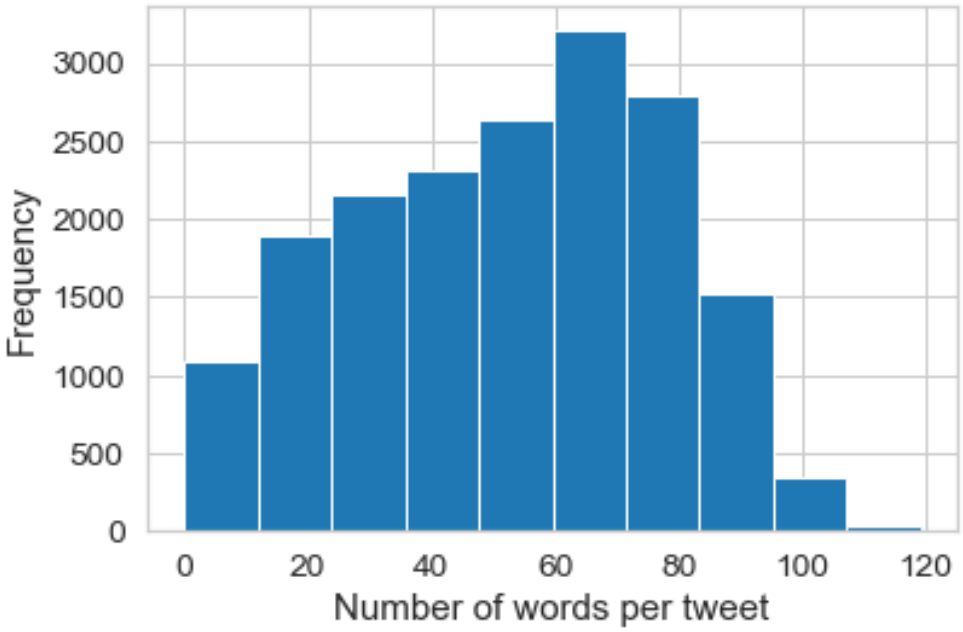

Figure 3.12 Word tokens per tweet – Oscar data set

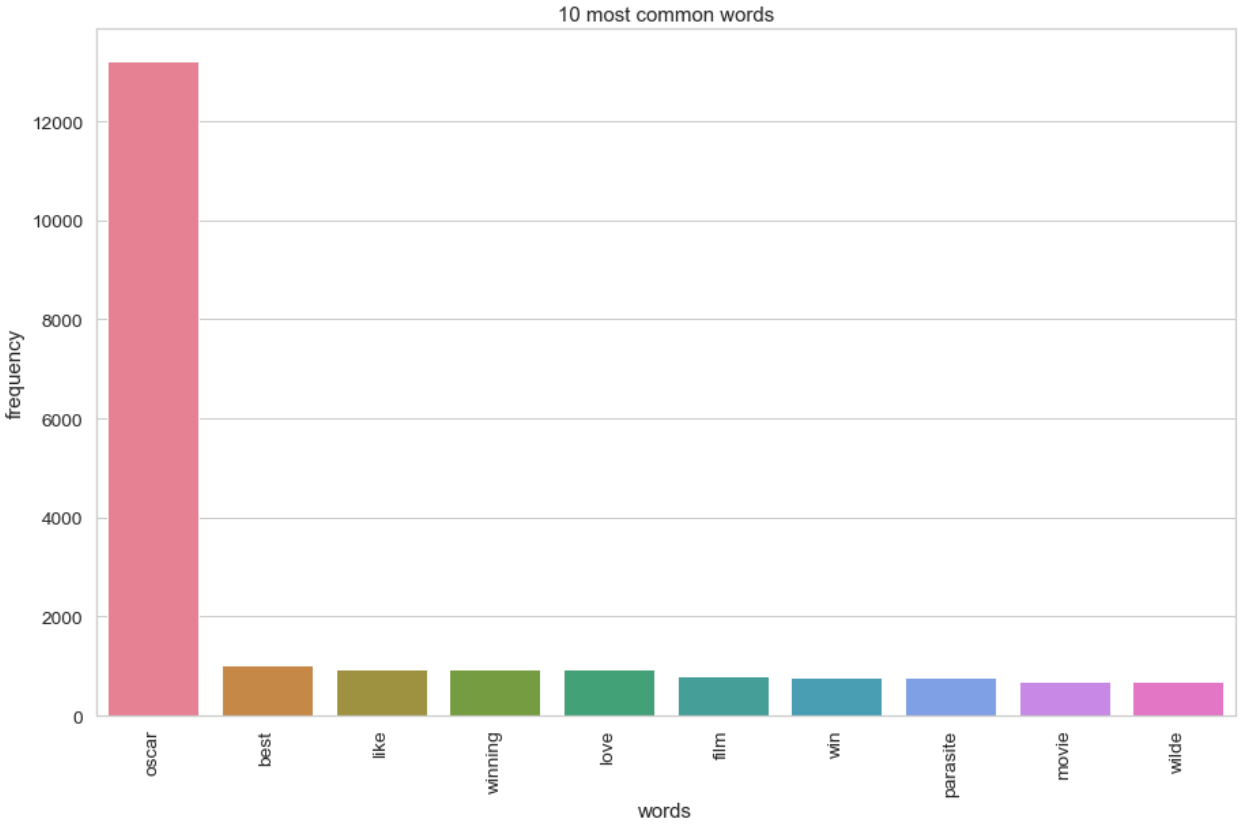

Figure 3.13 10 Frequent words in Oscar data set

Figure 3.14 Wordcloud - Oscar data set

3.3.4 Wetsuweten Data Set Analysis

Figure 3.15 shows the percentage composition of hashtags (#) and mentions (@) in the Westsuweten data set. Figure 3.16 shows the number of words per tweet. Figure 3.17 shows the 10 most frequent words in the data set while Figure 3.18 shows a Wordcloud for this data set.

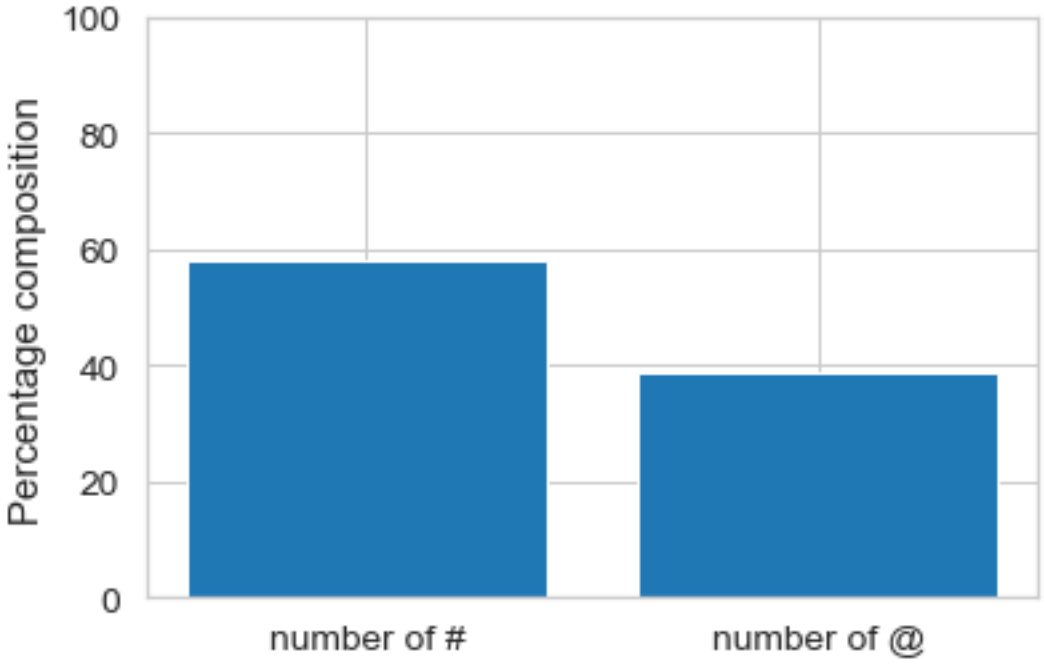

Figure 3.15 Percentage composition of Wetsuweten data set

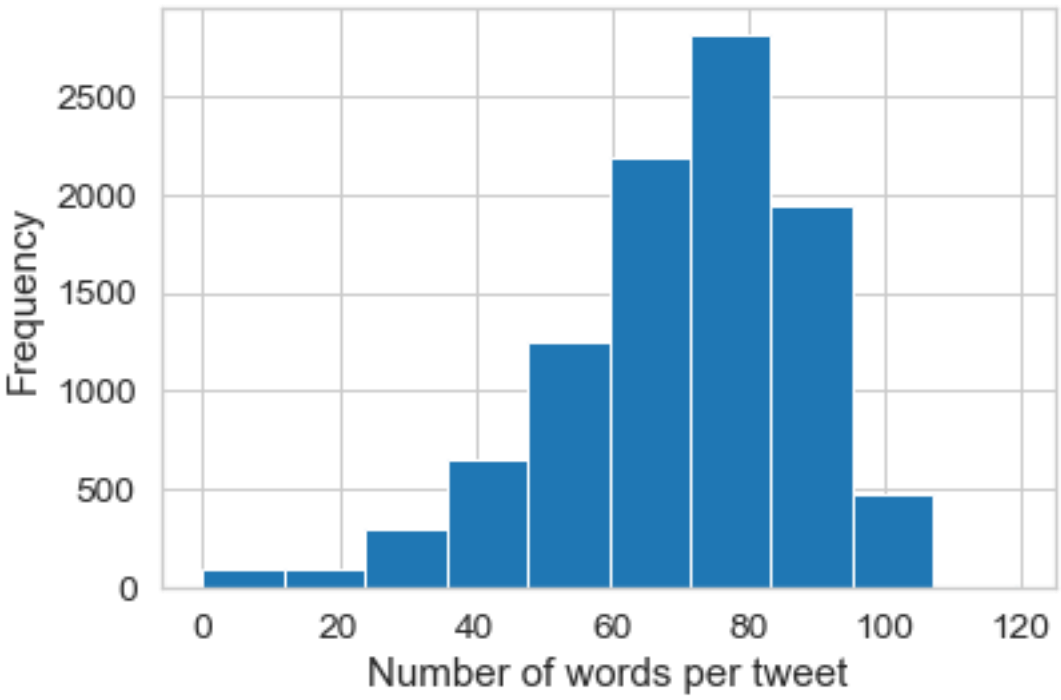

Figure 3.16 Word tokens per tweet – Wetsuweten data set

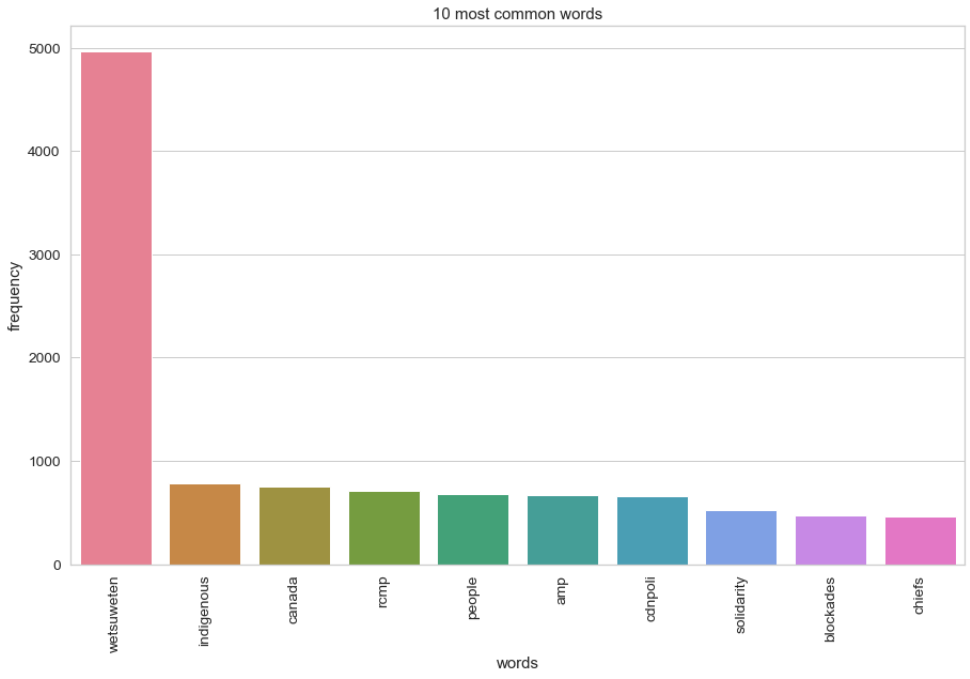

Figure 3.17 10 Frequent words in Wetsuweten data set

Figure 3.18 Wordcloud - Wetsuweten data set

3.3.5 Corona Data Set Analysis

Figure 3.19 shows the percentage composition of hashtags (#) and mentions (@) in the Corona data set. The Figure 3.20 shows the number of words per tweet. The Figure 3.21 shows the 10 most frequent words in the data set while the Figure 3.22 shows a Wordcloud for this data set.

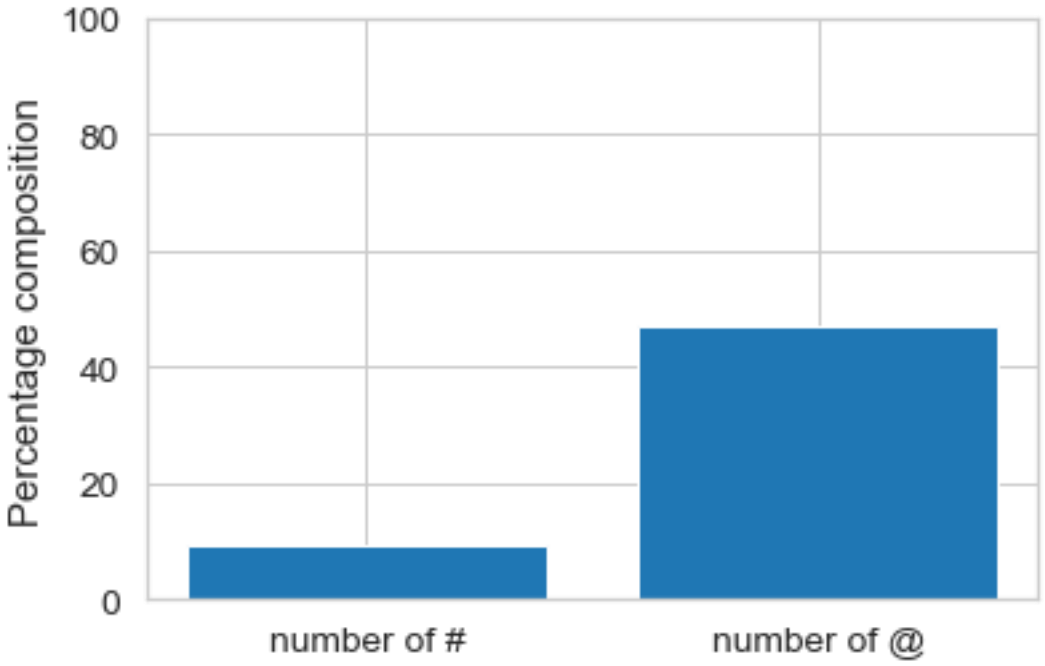

Figure 3.19 Percentage composition of Corona data set

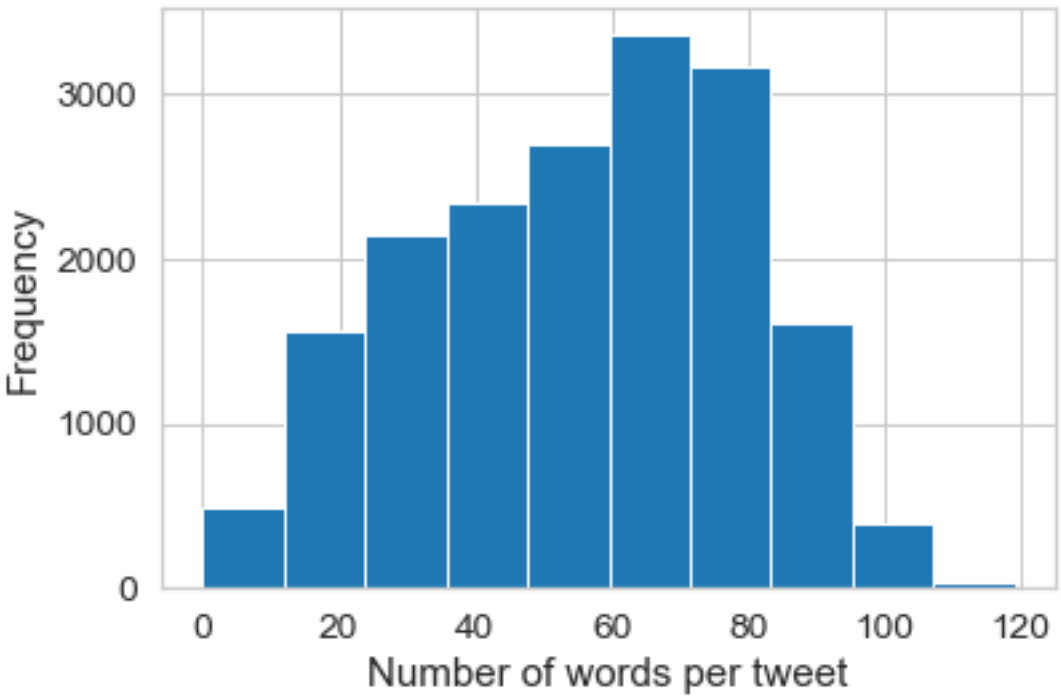

Figure 3.20 Word tokens per tweet – Corona data set

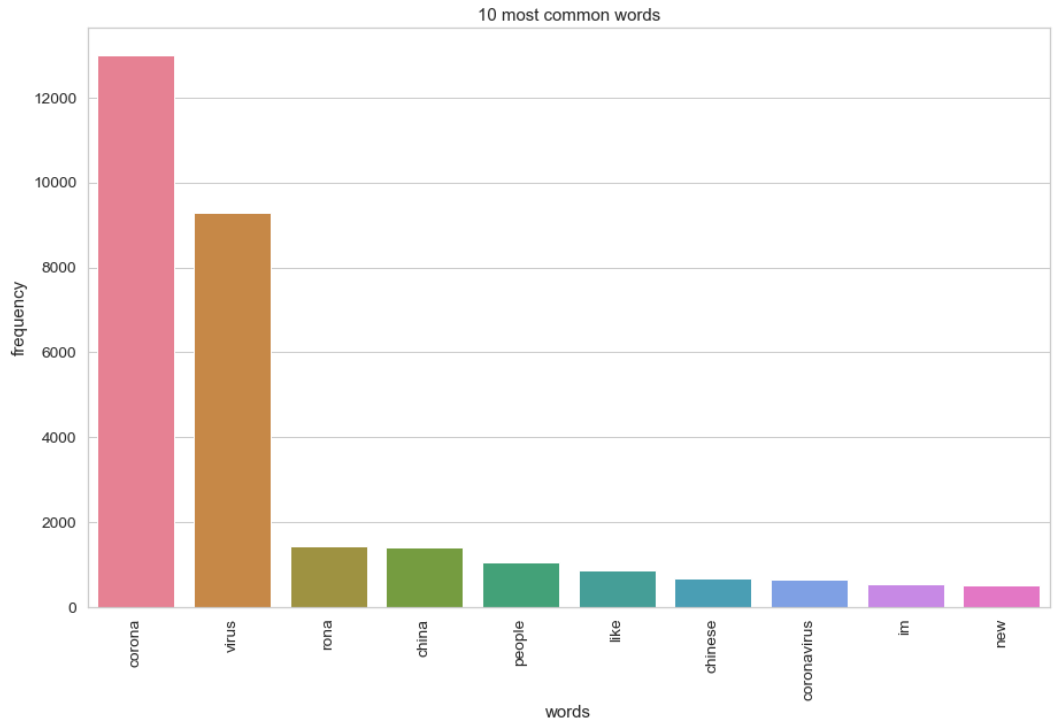

Figure 3.21 10 Frequent words in Corona data set

Figure 3.22 Wordcloud - Corona data set

3.4 Similarity measures:

Building on the conclusions from the research on effective similarity measures by Jubinville [17], T-Information distance and Jaccard word token distances will be considered as similarity measures in this thesis for generating the intra-set clusters.

3.4.1 Jaccard distance:

The Jaccard similarity measure or Jaccard coefficient is defined as the ratio of the number of shared attributes of two sets to the number of elements in the union of two sets [25]. A set can be a group of bits, words, strings, or functions. If the set consists of characters, it is known as character token or word token if the set contains words. The type of token determines which attribute to compare to compute the distance. Sets need not be of the same size and it is mathematically represented in Equation 3.1 and the measure lies between 0 and 1. Let A, B be two non-empty sets, where $|.|$ denotes set cardinality.

$$Jaccard_{Coeff}(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.1}$$

The Jaccard distance is defined as the complement of the Jaccard coefficient and is expressed below as shown in Equation 3.2.

$$Jaccard_{dist}(A,B) = 1 - Jaccard_{Coeff}$$

$$= 1 - \frac{|A \cap B|}{|A \cup B|} \in [0,1] \ \forall (A,B) \tag{3.2}$$

By observing the Equation 3.2 and considering the extreme cases, where the two sets A and B are equal. Then $|A \cap B| = |A \cup B|$ and $Jaccard_{dist}(A,B) = 0$. In the second extreme case, if A and B are dissimilar such that, $A \cap B = 0$ then $Jaccard_{dist}(A,B) = 1$. Hence, Jaccard distance always lies between [0,1] and the value of 0 indicates exactly similar sets and the value of 1 indicates dissimilar sets. Example 3.1 shows a sample calculation of Jaccard word token distance of two tweets X and Y.

Example3.1:

Let the tweets be X="Happy" and Y="Happier"

As the tweets in this example consist of single words, no further tokenization (process of splitting a string by space delimiter) is needed.

$$|X| = |Y| = 1$$

The next step involves evaluating the number of common tokens and the number of unique tokens in the tweets X and Y.

As we see, the tokens are both unique and have no common tokens among them,

The number of common tokens, $(X \cap Y) = \emptyset \implies |X \cap Y| = 0$

The number of unique tokens, $(X \cup Y) = \{happy, happier\} \implies |X \cup Y| = 2$

From the Equation 3.2, $Jaccard_{Dist}(X,Y) = 1 - \frac{|X \cap Y|}{|X \cup Y|}$

$$Jaccard_{Dist}(X,Y) = 1$$

3.4.2 T-Information distance:

T-Information is a measure of string complexity developed by Titchener [35]. It works by decomposing strings into base T-Codes and it measures the information contained in one string as compared to information in the second string. The T-information for two tweets, X and Y can be computed by effectively compressing string X using the basis strings generated by a T-code decomposition of string Y. The information distance then becomes a function of how well one string's base strings compresses the second string [17]. This compression system is robust to small variations and hence, is robust to common issues in tweets like character level errors including spellings. Rebenich [36] defined the T-Information distance mathematically in Equations 3.3 and 3.4.

$$TInfo_{Dist}(X,Y) = \frac{\max\{C_T(X \mid Y), C_T(Y \mid X)\}}{\max\{C_T(X), C_T(Y)\}} \tag{3.3}$$

$$TInfo_{Dist}(X,Y) = \frac{C_T(XY) - \min\{C_T(X), C_T(Y)\}}{\max\{\, C_T(X)\,, C_T(Y)\,\}} \tag{3.4}$$

Where X and Y are two strings, $C_T(X \mid Y)$ is the conditional T-complexity which is defined as the additional decomposition effort required on string X after application of all copy factor/copy pattern combinations encountered during the decomposition of string Y. The T-Information decomposition generates a set of copy factors by filtering the information in a string with a set of copy patterns. Equation 3.4 is an implication of Equation 3.3 where,

$$C_T(X \mid Y) \approx C_T(XY) - C_T(Y) \text{ and } C_T(XY) \approx C_T(YX)$$

From the Equation 3.4, the strings of different lengths can produce values outside of [0,1] as the applied normalization is with respect to the maximum of the individual string complexities and if the combined string, XY has higher complexity and one of the individual strings has lower complexity, then T-Information distance can be > 1. This calls for a normalization function to restrict the measures to [0,1].

A sigmoid function is used to smoothen out the trailing values and to ensure the value always lies between [0,1]. For this research, N. Rebenich's FLOTT C-code implementation[36], [39] was used along with python bindings written by Michael Anderson [40].

$$TInfo_{Norm}(X,Y) = \begin{cases} d, & 0 \le d < 0.95 \\ \dfrac{1}{1 + e^{-(d+3.77)}}, & d \ge 0.95 \end{cases} \tag{3.5}$$

Equation 3.5 shows the normalized T-Information distance measured used in this thesis, where 'd' is the obtained T-Information distance measure. In other words, the T-information distance is split into two ranges, where for values [0,0.95) it is a linear function and for values ≥ 0.95, a sigmoid function is used to restrict the T-Information distance value to [0,1]. The calculation of the parameters in the sigmoid function is shown in Appendix A. Figure 3.23 shows a snippet of evaluating T-Information distance and the Example 3.2 shows a sample T-Information distance calculation for two tweets.

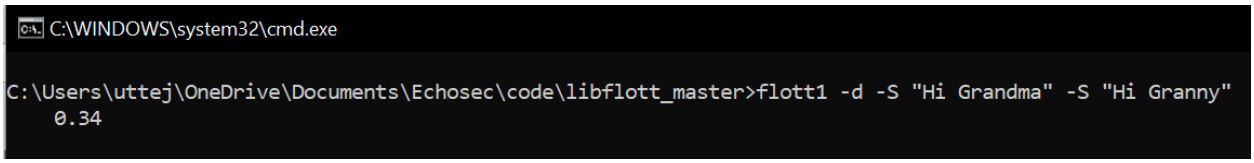

Figure 3.23 Sample T-info distance measure

Example 3.2:

Let the two tweets be X=”Happy” and Y= ”Happier”. The evaluation of T-Information distance, begins by decomposition of the tweets, as shown in the Table 3.1 and Table 3.2.

Table 3.1 T-transform of string XY

Decomposition		i : level	k_i : copy factor	p_i : copy pattern	
				length	value
$C_T(XY)$	$C_T(Y)$	1	1	1	r
		2	1	1	e
		3	1	1	i
		4	2	1	p
		5	1	1	a
		6	1	1	H
	$C_T(X\|Y)$	7	1	5	Happy

Rebenich [36] defined the conditional T-complexity of a string Y, $C_T(Y)$ as the log-weighted sum of copy factors, k_i associated with copy patterns, p_i that contain information belonging to the string Y as

$$C_T(Y) = \sum_i log_2(k_i + 1)$$

From the definition above, calculating the conditional complexities of $C_T(XY)$ and $C_T(Y)$ from the Table 3.1.

$$C_T(Y) = \sum_{i=1}^{6} log_2(k_i + 1) = 6.58$$

$$C_T(XY) = \sum_{i=1}^{7} log_2(k_i + 1) = 7.58$$

Table 3.2 T-transform of string YX

Decomposition		i : level	k_i : copy factor	p_i : copy pattern	
				length	value
$C_T(YX)$	$C_T(X)$	1	1	1	y
		2	2	1	p
		3	1	1	a
		4	1	1	H
	$C_T(Y\|X)$	5	1	1	r
		6	1	1	e
		7	1	5	Happi

Now, calculating the conditional T-complexity of $C_T(X)$ and $C_T(YX)$ from Table 3.2.

$$C_T(X) = \sum_{i=1}^{4} log_2(k_i + 1) = 4.58$$

$$C_T(YX) = \sum_{i=1}^{7} log_2(k_i + 1) = 7.58$$

From Equation 3.4, the T-Information distance is given by

$$TInfo_{Dist}(X,Y) = \frac{C_T(XY) - \min\{C_T(X), C_T(Y)\}}{\max\{ C_T(X), C_T(Y) \}} = \frac{3.0}{6.58} = 0.45$$

3.5 Clustering methods

This section introduces the naïve clustering methodology and discusses our proposed algorithm to cluster tweets based on the similarity distance measures.

Notation:

The following notation is used throughout this section. The number of tweets in a topic is represented by *N*. Assuming we have a cluster C_k, one of a set of *K* clusters. The tweets in this cluster are represented by C_k= { $tweet_j^k$ | j=1, 2,..,J_k }. The centre of this cluster is represented by $centre_k$ and is shown in the Figure 3.24.

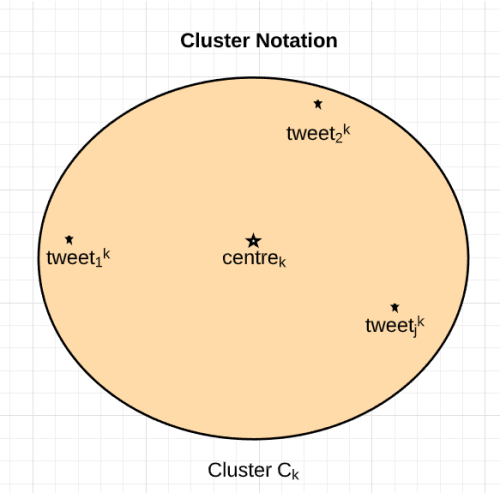

Figure 3.24 Sample cluster notation

3.5.1 Naïve K–means Algorithm

The following section discusses the methodology used in a naïve K – means clustering algorithm [41]. The idea is to define *K* initial centres, one for each of the K clusters. Data points being considered as centres are selected randomly from the data set. The next step involves assigning data points to the closest cluster centre. Each data point is taken and the distance to all the cluster centres is calculated and is assigned to the centre with the closest distance measure. When no data point is pending, the first step is completed, and the initial assignment is done. The next step involves calculating new cluster centres, one for each of the clusters. The new centres are usually either the mean or median of the existing data points in the cluster. After we have these *K* new centres, the data samples are assigned to the nearest new centre and the process is iterated. As a result, we notice that the *K* cluster centres change their location step by step until no more changes are observed. This algorithm aims at minimizing the objective function, in this case, a sum of square error function. The objective function is given in Equation 3.6 [41].

$$Objective\ Function = \sum_{k=1}^{K} \sum_{j=1}^{J_k} \left\| x_j^k - c_k \right\|^2 \qquad (3.6)$$

,

where $\left\| x_j^k - c_k \right\|^2$ is the square of distance measure between a data point, x_j^k of the cluster *k* and its cluster centre c_k.

3.5.2 Proposed Modified K-Means clustering

This subsection introduces the proposed modified K-means algorithm used in this thesis. A detailed description of the methodology along with the algorithm is discussed.

3.5.2.1 Description

As discussed in the Section 3.5.1, K-means clustering is an unsupervised clustering algorithm that requires *K* number of initial centres to form *K* clusters and it works by minimizing the sum of square error value on every iteration until the desired threshold is reached. But this approach cannot be applied in our case to cluster the tweets as our data is of string datatype, and hence we use distance measures based on context similarity to transform the character set to number space. In particular, our cluster centres are tweets and not means or medians taken from the numerical data set.

Table 3.3 Proposed Models

Model Name	Initial tweets	Distance measure
T-Information – Random	Randomly chosen.	T-Information distance
T-Information – Dissimilar	Dissimilar tweets based on T-Information distance measure.	
Jaccard – Random	Randomly chosen.	Jaccard distance
Jaccard – Dissimilar	Dissimilar tweets based on Jaccard distance measure.	

Four variations of the modified K-means algorithm were developed and tested. These four variations are listed in Table 3.3. Two of the above models align closely with the traditional approach of selecting random data points as centres, while the other two models use a distance measure to select the most dissimilar tweets as initial centres. This ensures that the clusters are placed far away from each other. The objective function, *J* of these models follows the traditional K-means objective function and is represented in Equation 3.7.

$$J = \sum_{k=1}^{K} \sum_{j=1}^{J_k} \left[d\left(tweet_j^k, centre_k\right) \right]^2 \qquad (3.7)$$

where $d\left(tweet_j^k, centre_k\right)$ is the distance between a tweet, $tweet_j^k$ of the cluster *k* and its cluster centre, $centre_k$.

3.5.2.2 Clustering Algorithm Pseudo Code

This sub-section presents the platform independent pseudo code of the proposed algorithm and the detailed description of the algorithm is presented in the next section. Figure 3.25 shows the pseudo code of the modified K-means algorithm.

Pseudo Code

```
1: Initialize K ( number of centres)
2: repeat
3:      Select K tweets as the initial centres ( based on the type of centre initialization )
4:      for i=1 to max_iterations or until iteration convergence do
5:           for each tweet do
6:                Calculate distance to all centres and assign the tweet to the
                  centre with the lowest distance.
7:           end for
8            Evaluate SSE as per Equation 3.7
9:           Update the centres of each cluster as per Equation 3.8
10:     end for
11: Increment K until the convergence condition is met.
```

Figure 3.25 Pseudo code of the proposed algorithm

3.5.2.3 Proposed Algorithm

This section presents a high-level overview of the proposed modified K-means algorithm. The algorithm follows a similar structure for the four variations and differs in some steps, indicated within the algorithm. The distance measure used in this algorithm is either a Jaccard word token distance or T-Information distance measure based on the variation as indicated in Table 3.3.

Step 1:

Input: Data set, distance measure, data set size, maximum iteration count.

The model takes as input the data set and the number of tweets in the data set to be clustered. Let the data set to be clustered is 'dataset' and the number of tweets to be clustered be 'N'. Let the number of initial centres be '*K*' to form *K* clusters. A maximum iteration count is needed to ensure the model does not enter an infinite loop. If the model does not converge before the specified maximum iteration count, the last returned state of the model after the maximum iterations is considered for further evaluations.

Step 2:

Initializing variables: tweet.clusterID=-1, threshold (ε), number of centres *K* =1 and maximum_interation_count.

- In this step, we initialize the key parameters such as threshold (ε) which decides the model's convergence. If the difference between the consecutive iteration's objective function (SSE) is less than the threshold, we consider the model to have converged.
- The clusterID of all the tweets is set to -1, to indicate all the tweets are unclustered.
- The number of centres is set to 1 (*K*=1).

Step 3:

Initializing centres:

In this step, the initial centres are selected.

- A counter to track the number of iterations is initialized and set to 1, iteration_count=1.
- If the model is random initialization, proceed to Step 4.
- If the model is dissimilar initialization, proceed to Step 5.

Step 4:

Random centres initialization:

- *K* tweets are selected randomly out of the N tweets and are considered the centres of *K* clusters. The cluster ID of these tweets is set to [1,2,…,*K*].

Step 5:

Dissimilar centres initialization:

- Distance from each tweet to all the other tweets in the data set are calculated.
- The nearest neighbor of a tweet, $tweet_i$ is the tweet for which the distance between the two tweets is the smallest.
- *K* tweets with the largest distance to their nearest neighbor are selected as centres of *K* clusters. The cluster ID of these tweets is set to [1,2,3,…,*K*].
- Dissimilar centre initialization ensures that these *K* centres are spread out and far away from each other.

Step 6:

Checking for maximum iterations:

- Check, if the number of iterations is equal or greater than the pre-set, maximum iteration count.
- If the iteration_count > maximum iteration count, proceed to Step 12. Store the value of $SSE_{maxiteration}$ calculated in Step 8 in SSE_K.
- If the iteration_count ≤ maximum iteration count, proceed to the next step.

Step 7:

Assigning tweets to clusters: $\boldsymbol{tweet_i.clusterID} \leftarrow [\min_{1 \le k \le K} \boldsymbol{d(tweet_i, center_k)}].\boldsymbol{clusterID}$

- Distance from each unclustered tweet is calculated to each of the cluster centres, and the tweet is assigned to the cluster whose distance to the centre is minimum.

Step 8:

Calculating the sum of squares error value: $SSE^K_{iteration\_count}$

- The Sum of squares error is evaluated.

Step 9:

Check for convergence on iterations:

- The difference of the $SSE_{iteration}$ value calculated in Step 8 and the $SSE_{iteration}$ value in the previous iteration is compared, and if this value is less than the threshold.
    1. Store the value of $SSE_{iteration}$ calculated in Step 8 in SSE_K.
    2. Skip to Step 12.
- If the difference is greater than the threshold, proceed to the next step.

Step 10:

Finding new centres:

New centres in every iteration are updated based on the following conditions.

In each cluster, we search for a

- Tweet with minimum distance measure to all the tweets within the cluster (intra-cluster distance)
- Tweet with maximum distance measure to all the other centres of different clusters (inter-cluster distance)

The first condition ensures the clusters have a low spread (tight), while the second condition ensures the clusters are far apart. Merging both the conditions and having them in a mathematical form is represented in Equation 3.8. Assuming the model has *K* clusters, [0,1,..k,..K] and let the cluster *k* have J_k tweets in it and its cluster centre is denoted as $centre_k$ and the new centre be $centre'_k$.

$$centre'_k = \frac{\min\ (intra - cluster\ distance)}{\max\ (inter - cluster\ distance)}$$

$$centre'_k = \min_{1 \le j \le J_k} \left[\frac{\sum_{j=1, j \neq i}^{J_k} \left\| d\left(tweet_j^k,\ tweet_i^k\right) \right\|^2}{\sum_{p=1}^{K} \left\| d\left(tweet_j^k, centre_p\right) \right\|^2} \right] \quad (3.8)$$

Step 11:

Increment iteration counter: iteration_count = iteration_count+1

- The value of the iteration count is incremented by 1.
- Go to Step 6.

Step 12:

Check for convergence on the number of centres:

- The difference of the $SSE_{centres}$ value and the $SSE_{centres}$ value for the previous number of centres is compared, and if this value is less than the threshold.
    1. Skip to Step 14.
- If the difference is greater than the threshold, proceed to the next step.

Step 13:

Increase the number of cluster centres: K = K+$\frac{N}{4}$

- The number of cluster centres is increased by the nearest integer value of $\frac{N}{4}$.
- The increments of cluster centres by $\frac{N}{4}$, ensures the number of cluster centres is iterated by at least 25 equal increments.
- Go to Step 3.

Step 14:

Output: Clustered data set, analysis, and performance measures.

3.6 Analysis and Performance measures:

To evaluate the clustering model's performance and validation, the following measures are defined and briefly discussed below.

3.6.1 Cluster Size

The cluster size of a cluster is defined as the number of tweets in that cluster. We evaluate two parameters, the size of the largest cluster and the size of the smallest cluster. The largest cluster is defined as the cluster with the highest number of tweets in that cluster. Likewise, the smallest cluster is the cluster with the lowest number of tweets in that cluster.

Although these parameters are usually data-driven i.e., these values are directly impacted by the content of the tweets. In a random pick of data, there could be tweets talking about the exact same or similar content which usually occurs when the topic has a high throughput or when a high impact event occurs. A throughput value is defined as the number of tweets generated per second. An example of high throughput topic is #RoyalWedding or an event like #Blacklivesmatter where a lot of users usually re-post other user's content to update or educate their followers. In a good pick of data, the presence of a larger cluster would indicate the presence of a dominant subtopic.

3.6.2 Data Compression Rate

The data compression rate is defined as the ratio of the number of tweets that can represent the whole set to the total number of tweets and is represented mathematically in Equation 3.9. A cluster is a valid cluster if it contains at least two tweets, i.e. minimum cluster size = 2. The value of the minimum cluster size is a tunable parameter and can be adjusted to filter out single-tweet clusters. Similarly, a lone cluster is a cluster with only the cluster centre. Let *K* be the total number of clusters which includes both valid and lone clusters.

$$Data\ Compression\ Rate = \left(\frac{|N| - K}{|N|}\right) X\ 100 \tag{3.9}$$

where |. | is the set cardinality operator and N is the total number of tweets in the data set.

This is an important performance measure as it effectively measures the total reduction rate, an ideal clustering algorithm has a higher data compression rate. Cluster centres are considered exemplars which means that they can represent the entire cluster. Based on Equation 3.9, this measure generates values in the range [0,100] and denotes percentage compression. The value of 0% indicates no data compression and that all the tweets in the initial data set exist in the final clustered data set as lone clusters.

3.6.3 Valid Cluster Ratio (VCR)

We observe that the data compression rate measure by itself does not give a complete overview of the model's performance as, by Equation 3.9, the number of clusters is a combination of both valid clusters and the lone clusters. Hence, we introduce the valid clusters ratio (VCR) measure to focus on the number of valid clusters the model can generate. This measure is mathematically represented in Equation 3.10.

$$VCR = \left(\frac{K - LC}{K}\right) X\ 100 \tag{3.10}$$

where LC is the number of lone clusters, *K* is the total number of clusters and a cluster, C_k is a lone cluster if $| C_k | \leq 1$.

Based on Equation 3.10, the VCR measure ranges from [0,100] and denotes the percentage of valid clusters. An ideal clustering model would have a higher VCR measure, indicating a higher number of valid clusters.

3.6.4 Number of Clusters

The number of clusters refers to the total number of clusters generated by the model for a given data set. It consists of both valid and lone clusters.

This measure usually indicates the number of sub-topics present in the data set. The presence of a large number of lone clusters may indicate there are multiple subtopics and a low number of lone clusters may indicate the presence of a dominant subtopic. In general, the relationship between the number of clusters and the data set size is non-linear and is data-driven as it depends on the pick of the data.

3.6.5 Intra-cluster distance (MSE)

The Intra cluster distance of a cluster is computed mathematically by Equation 3.11. It measures the compactness of a cluster. The compactness of a cluster is a measure of how close the tweets in a cluster are to the cluster's centre. A low value indicates a very dense cluster with tweets very closely related to each other and the sub-topic of that cluster, while a large value indicates a sparse cluster and the tweets in this cluster might not be totally related to each other although they may belong to the same sub-topic. The MSE value is non-negative and for an ideal clustering algorithm with ideal data set, this value would be 0.

$$MSE_k = \frac{1}{J_k}\sum_{j=1}^{J_k}\left[d\left(tweet_j{}^k, center_k\right)\right]^2 \qquad (3.11)$$

$$MSE = \frac{1}{K}\sum_{k=1}^{K} MSE_k \qquad (3.12)$$

MSE_k is the mean sum of squares error, for a cluster k. The mean of these values across all the clusters is given by the MSE value, calculated by Equation 3.12. The d(x,y) is the distance measure which can be either Jaccard word token or T-Information distance depending on the chosen model.

Risk of Over-fitting:

Figure 3.26 shows a plot of mean SSE values for different values of the number of initial centres. The data set used for this evaluation is London with 138 tweets. The number of clusters was iterated from 1 to 138 and a mean SSE value was calculated for every value of the number of clusters. As seen in Figure 3.26 the value of mean SSE decreases with an increasing number of clusters and finally reaches 0 where the number of clusters is equal to the number of tweets in the data set. In that case, each tweet forms a cluster and becomes its own centre and the distance to itself is zero, resulting in mean SSE value to be zero. This is the overfitting of clusters to the data and it presents no value since there is no reduction in the size of the clustered data set.

To reduce the risk of over-fitting, we set a threshold, denoted by epsilon (ε) and is chosen to be 10^{-3} for this thesis. This is a settable parameter and is indirectly proportional to the computations required. We consider our algorithm to have converged if the difference between objective function of the current iteration to the value from the previous iteration is less than epsilon.

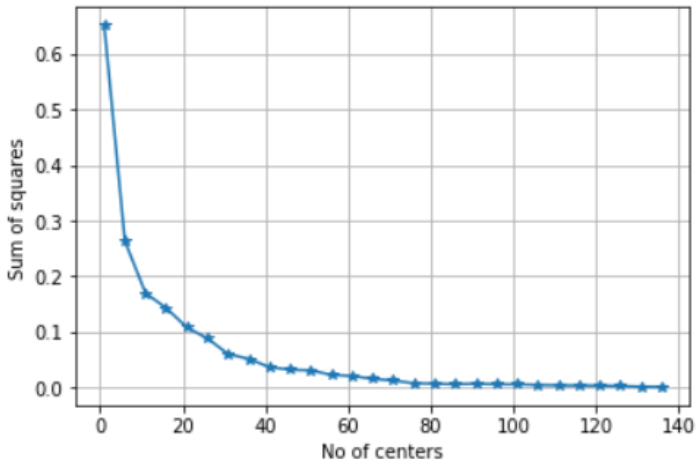

Figure 3.26 Sum of squares error vs number of centres

3.6.6 Inter-cluster Distance

The inter-cluster distance (ICD) measure, as the name suggests is the measure of how far the clusters are to each another. The inter-cluster distance of a cluster C_k is denoted by ICD_k and is mathematically represented by Equation 3.13. The average of this measure across all the *K* clusters is given by Equation 3.14 and is denoted by ICD_{mean}. For a good clustering algorithm, the mean ICD value should be as large as possible. The distance measure can be either Jaccard word token or T-Information distance.

$$ICD_k = \sum_{i=1}^{K} [d(centre_k, centre_i)]^2 \qquad (3.13)$$

$$ICD_{mean} = \frac{1}{K} \sum_{k=1}^{K} ICD_k \qquad (3.14)$$

3.6.7 Silhouette Coefficient

The silhouette index, S_i^k is calculated for each sample and the Silhouette coefficient is evaluated for the whole model. The silhouette index of a cluster S_k, is the average of the silhouette index of all the elements in that cluster. The silhouette coefficient of the model is the average of the silhouette index of all the K clusters of that model. The Silhouette index of an element of a cluster C_k, denoted as S_i^k is mathematically calculated by Equation 3.15 and is defined as the normalized difference of the inter-cluster separation to the cluster's tightness (compactness) [42]. The silhouette index of a cluster of centre K is denoted as, S_k is expressed mathematically by Equation 3.16 and the overall silhouette coefficient is expressed mathematically in Equation 3.17 [42].

$$S_i^k = \frac{(b_i - a_i)}{\max(a_i, b_i)} \qquad (3.15)$$

$$a_i = \frac{1}{J_k}\sum_{j=1}^{J_k}\left[d(tweet_i^k, tweet_j^k)\right],\ b_i = \frac{1}{J_p}\sum_{j=1}^{J_p}\left[d\left(tweet_i^k, tweet_j^p\right)\right]$$

$$S_k = \frac{1}{J_k}\sum_{j=1}^{J_k} S_j^k \qquad (3.16)$$

$$Silhouette\ Coefficient = \frac{1}{K}\sum_{k=1}^{K} S_k \qquad (3.17)$$

where,

a_i is the average distance of the i_{th} sample in the cluster C_k to the other J_k tweets in the same cluster and

b_i is the average distance of the i_{th} sample in the cluster C_k to the J_p tweets in the closest cluster C_p.

Based on Equation 3.15, the value of the Silhouette coefficient lies between [-1, 1]. For a good clustering model, the overall silhouette coefficient should be positive and be close to 1. The following discusses the possible index values and their implications.

- The Silhouette coefficient is dependent on both the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. If the Silhouette index of a tweet is close to +1, it implies that the tweet is well-clustered and is already assigned to a very appropriate cluster.
- If the Silhouette coefficient of a tweet is close to 0, the tweet could be assigned to another closest cluster and it lies equally far away from both the clusters which could indicate the possibility of overlapping clusters.
- If the Silhouette coefficient of a tweet is close to –1, the tweet is misclassified and is placed in the wrong cluster.

3.6.8 Davies–Bouldin Index

The Davies-Bouldin index (DB) is the average similarity measure between each cluster and its most similar one, averaged over all the clusters, K and is mathematically represented in Equation 3.19. DB index is the ratio of intra-cluster distance to inter-cluster distance of each cluster to its most similar neighbour, averaged over all the K clusters [43]. Thus, clusters that are farther apart and with lower within cluster scatter will result in a lower DB Index value. It is mathematically represented as follows:

$$DB\ Index = \frac{1}{K} \sum_{i,j=1,i\neq j}^{K} \max\left(\frac{IC_i + IC_j}{M_{ij}}\right) \tag{3.19}$$

$$IC_i = \frac{1}{J_k} \sum_{a,b=1,a\neq b}^{J_k} \left[\mathrm{d}\left(tweet_a^i, tweet_b^i\right)\right] \tag{3.20}$$

$$M_{ij} = \mathrm{d}\left(center_i, center_j\right) \forall i,j \in [0,K]\,, where\ i \neq j \tag{3.21}$$

where,

IC_i, IC_j is the within-cluster distances of clusters i,j respectively, and M_{ij} is the inter-cluster distance of clusters i,j.

Based on Equation 3.19, the DB index value is non-negative, and the minimum value is zero, with lower values indicating compact clusters and better separation between the clusters. This value is dependent on the data set as well as the model and it evaluates the worst-case scenario. This measure affirms the idea that no cluster has to be similar to each other [43].This measure is higher for clusters that are more distinct (or dissimilar) from each other.

In general, the clusters evaluation methodologies are broadly dependent into two types, external evaluation measures and the internal evaluation measures [44]. External evaluation measures are used

to validate the clusters in the cases where the ground truth is available. But this is rarely the case in the real world data sets in which unsupervised clustering is required as a reference ground truth is not available [45]. The data sets presented in this thesis are obtained directly from the Twitter end point and, hence, do not contain ground truth. For this reason, internal cluster evaluation methodology is used for cluster validation. Silhouette coefficient and DB index are the two internal evaluation measures used in this research for validating the clusters generated by the models.

Chapter 4 Results

This chapter presents and discusses the evaluation parameters of the proposed four different clustering models. These are compared across the different data sets and data set sizes. A snippet of a cluster from the clustered data set is shown below in Table 4.1. The cluster shown belongs to Seattle data set of 2000 tweets for the T-Information dissimilar model.

Table 4.1 Snippet of a generated cluster

	A	B	C	D
	Content	clusterID	dLead	s
49	Want to work in #Seattle, WA? View our latest opening: https://t.co/xJgtFafWkS #Nursing #Job #Jobs #Hiring #CareerArc	49	1	
10	Want to work in #Seattle, WA? View our latest opening: https://t.co/l4CLPTay0Q #Nursing #Job #Jobs #Hiring #CareerArc	49	0	
51	Want to work in #Seattle, WA? View our latest opening: https://t.co/os7cKhwgMj #CustomerService #Job #Jobs #Hiring #CareerArc	49	0	
91	Want to work in #Seattle, WA? View our latest opening: https://t.co/kWlJJzKHyf #Sales #Job #Jobs #Hiring #CareerArc	49	0	
184	Want to work in #Seattle, WA? View our latest opening: https://t.co/6yrXQswW5e #Healthcare #Job #Jobs #Hiring #CareerArc	49	0	
195	Want to work in #Seattle, WA? View our latest opening: https://t.co/seHYZBc3NC #Nursing #Job #Jobs #Hiring #CareerArc	49	0	
198	Want to work in #Seattle, WA? View our latest opening: https://t.co/JE6AFOsvrR #Nursing #Job #Jobs #Hiring #CareerArc	49	0	
206	Want to work in #Seattle, WA? View our latest opening: https://t.co/xjLZ48Cfwk #Nursing #Job #Jobs #Hiring #CareerArc	49	0	
210	Want to work in #Seattle, WA? View our latest opening: https://t.co/nYDngjMm33 #Accounting #Job #Jobs #Hiring #CareerArc	49	0	
263	Want to work in #Seattle, WA? View our latest opening: https://t.co/Aynua4YEbZ #Surgeon #Job #Jobs #Hiring #CareerArc	49	0	
264	Want to work in #Seattle, WA? View our latest opening: https://t.co/rlteYXlPQV #Retail #Job #Jobs #Hiring #CareerArc	49	0	
307	Want to work in #Seattle, WA? View our latest opening: https://t.co/UQusEft9op #Insurance #Job #Jobs #Hiring	49	0	
315	Want to work in #Seattle, WA? View our latest opening: https://t.co/77gPgfkiT9 #Retail #Job #Jobs #Hiring #CareerArc	49	0	
318	Want to work in #Seattle, WA? View our latest opening: https://t.co/4SxZB2Avla #BusinessMgmt #Job #Jobs #Hiring #CareerArc	49	0	
327	Want to work in #Seattle, WA? View our latest opening: https://t.co/mkpIgvSPo7 #Nursing #Job #Jobs #Hiring #CareerArc	49	0	
379	Want to work in #Seattle, WA? View our latest opening: https://t.co/GyTGSeQ2P8 #Automotive #Job #Jobs #Hiring #CareerArc	49	0	
418	Want to work in #SEATTLE, WA? View our latest opening: https://t.co/QTmB0p02nh #Marketing #Job #Jobs #Hiring #CareerArc	49	0	
424	Want to work in #Seattle, WA? View our latest opening: https://t.co/wvlkYS8NOG #IT #Job #Jobs #Hiring #CareerArc	49	0	
443	Want to work in #Seattle, WA? View our latest opening: https://t.co/8lylZiyUyG #IT #Job #Jobs #Hiring #CareerArc	49	0	
467	Want to work in #Seattle, WA? View our latest opening: https://t.co/XFon1q1gSf #Healthcare #Job #Jobs #Hiring	49	0	

4.1 Modified K-Means approach:

The four models of the modified k-means algorithm discussed in Table 3.3 are run on five data sets. The performance of the models is evaluated across measures discussed in Section 3.6. The models are run for four different data set sizes of 250, 500, 1000 and 2000 tweets. Each model is run on the same tweets to compare their computational performance as well as the cluster quality.

4.1.1 Data Compression Rate

Figure 4.1 shows the plot of the data compression rate to the size of the data set. It is used to study the performance of the model and is defined as the ratio of the number of tweets that can represent the data set to the total number of tweets. It gives a measure of how good the model can filter the redundant tweets in the data set without losing the information.

From the subplots in Figure 4.1, the Seattle data set had the highest average compression rate for both the Jaccard and the T-Information models. The T-Information models were able to achieve an average of around 40% to 80% data compression rate across various data set sizes while the Jaccard models had close to 15% of compression. This is due to the presence of a trend in the Seattle data set as seen in Figure 3.9 where we observed that there were dominant word tokens.

Across all the data sets, T-Information models have achieved at least 25% higher compression rate than the Jaccard distance-based models. This is because Jaccard distance compares the word attributes and is very sensitive to grammatical errors and incorrect spellings. A minor difference in similar tweets could lead to a higher Jaccard distance, especially if the number of word tokens are small as seen in the Example 3.1 and Example 3.2.

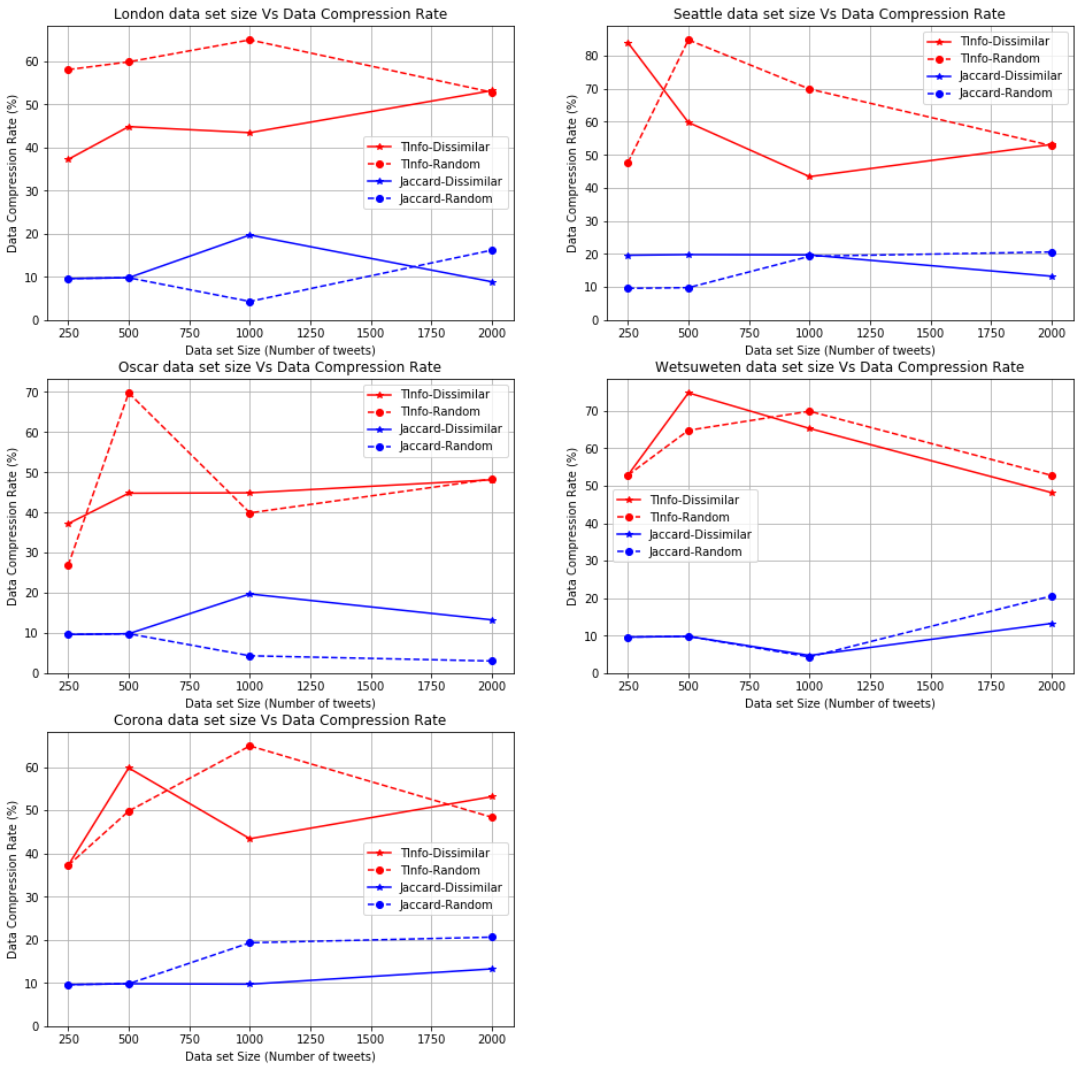

Figure 4.1 Data set size vs Data Compression Rate

4.1.2 Valid Cluster Ratio (VCR)

Valid cluster ratio of a model is defined as the ratio of the number of valid clusters to the total clusters. Figure 4.2 shows the plot of the VCR measure to the size of the data set. It is used to study the performance of the clusters by evaluating the number of valid clusters. A topic will have a higher VCR measure if it has a lower number of lone clusters.

As discussed in Section 3.6.3, for a good pick of data, an ideal clustering algorithm would have a higher value of VCR. A higher value of VCR for any given model would indicate that the pick of the data has fewer non-dominant subtopics while the inverse implies the data set is more spread.

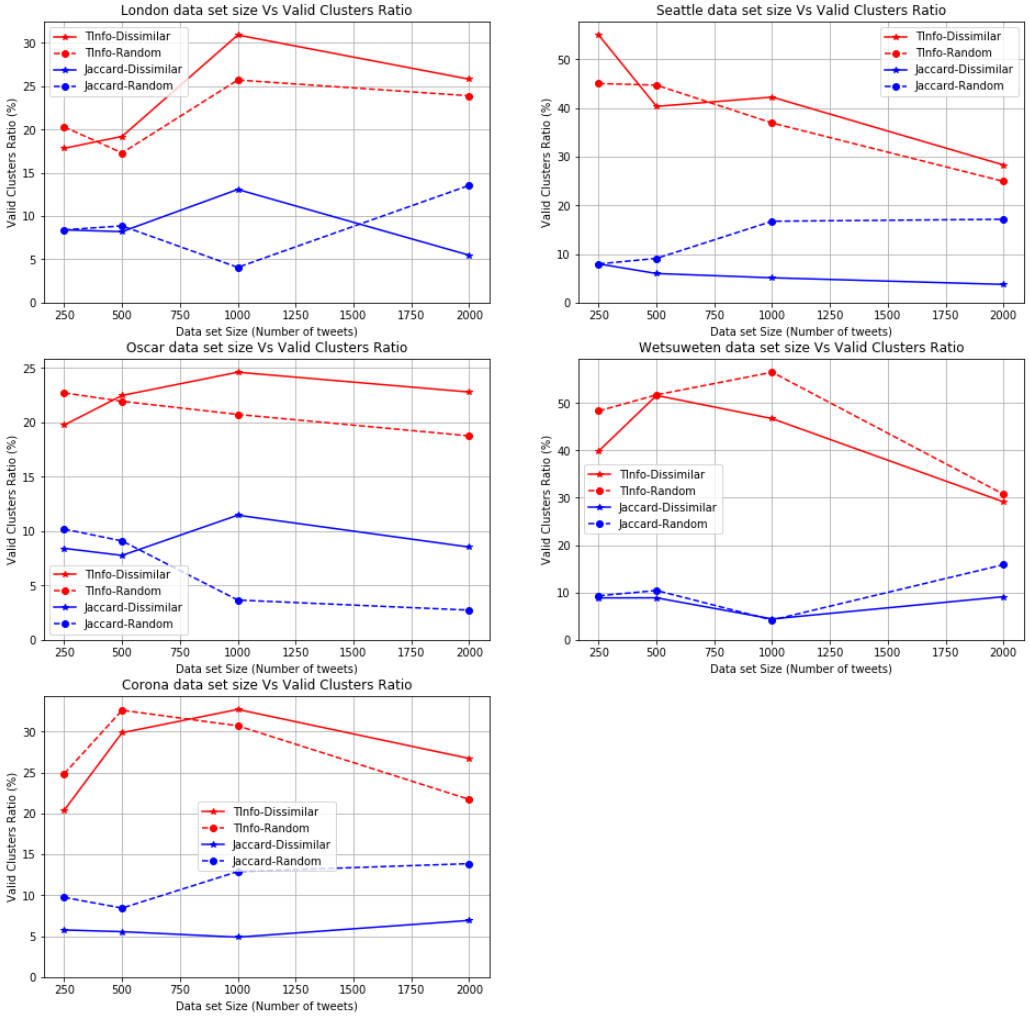

Figure 4.2 Data set size vs VCR

From the observation of the subplots in Figure 4.2, the Oscar data set had the lowest average VCR values, and this is because there are many subtopics in the data set contributing to a lot of lone clusters.

The presence of a large number of subtopics is described in Figure 3.13 shows an absence of a dominant word token in the data set.

4.1.3 Number of Clusters

From Section 3.5.2.3 of the model's algorithm, we see that the model converges at a certain number of clusters for a given data set. The plot in Figure 4.3 shows the number of clusters, where the model converges at various sizes of the data set. The four variations of the model are shown in each subplot and indicated with the legend. Each subplot shows the number of clusters in a data set. In general, each cluster in a topic would indicate a sub-topic. The number of clusters is the sum of the valid and lone clusters.

As the size of the data set increases, the number of clusters increases. This is because the addition of more tweets adds new sub-topics. This trend is observed across all the data sets. However, Jaccard-distance based models see a higher number of clusters than T-Information-distance based models and the observations from the Figure 4.2 shows the Jaccard based models having lower values of VCR measure than the T-Information models indicating that most of these clusters are lone clusters. The number of clusters measure along with the cluster sizes provides the reader with demographics such as the broadness and the presence of subtopics in the topic searched.

In a good pick of the data, a large number of clusters with small cluster sizes would usually indicate a broad topic with more subtopics while a small number of clusters with big cluster sizes would indicate a narrower and more focussed topic. Higher the number of clusters would mean longer time and the calculations required for the model to converge.

From the visual observation of Figure 4.3, we see that out of all the five data sets(topics) and for a fixed size of the data set, the current data sets returned more average clusters than the historical data sets indicating the topics are broad and have many sub-topics. Among the models, T-Information based models had higher data compression rates as seen in Figure 4.1 and hence, resulted in a lower number of clusters.

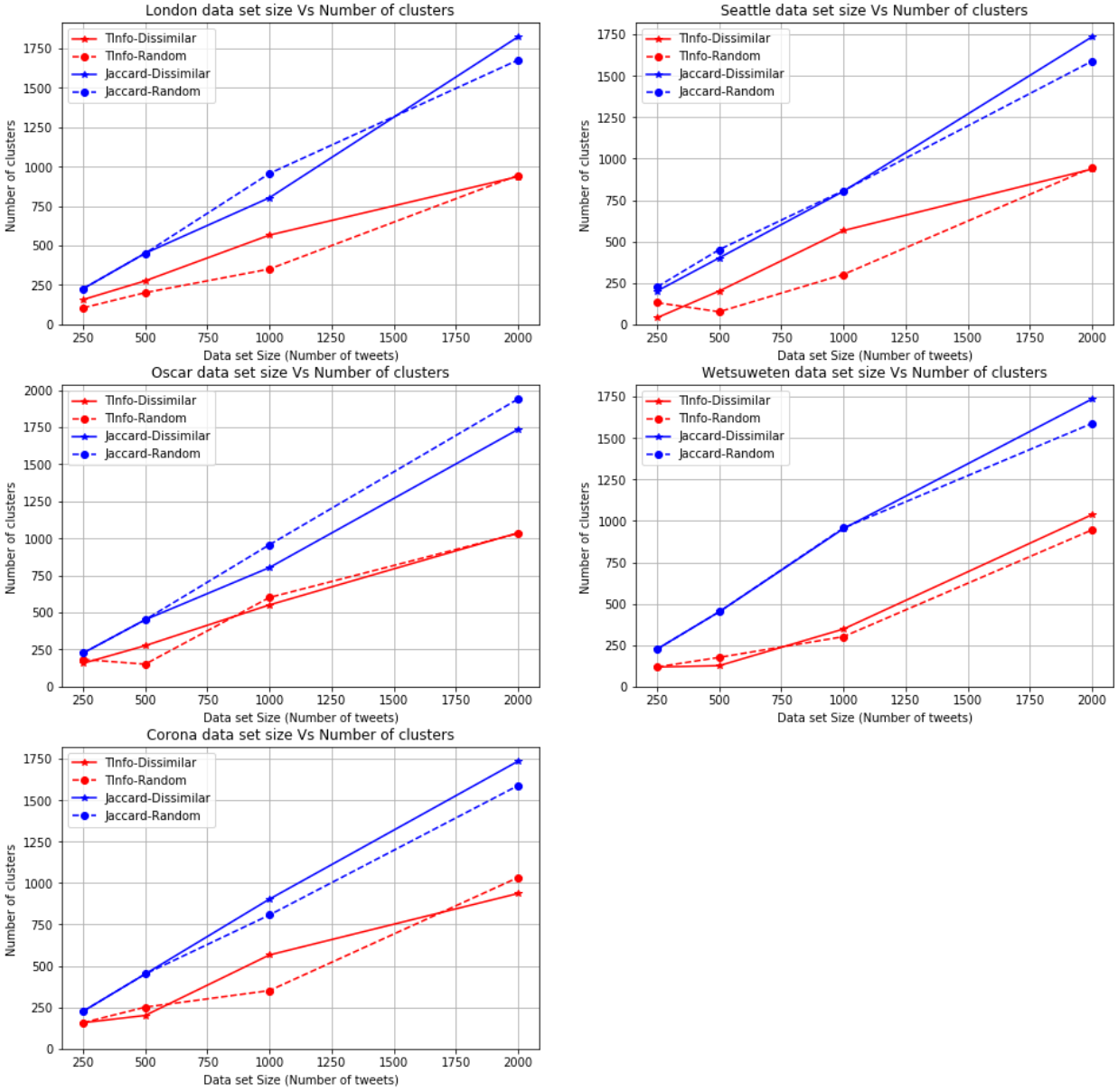

Figure 4.3 Data set size vs Number of Clusters

4.1.4 Size of clusters

The size of a cluster is defined as the total number of tweets in the cluster Figures 4.4 and 4.5 show the plots of the size of the largest cluster and the size of the smallest cluster respectively to the data set sizes for all the five data sets. The four variations of the model are shown in each subplot and indicated with the legend.

The largest cluster is defined as the one which has the highest number of tweets and likewise the smallest cluster is the one with the least number of tweets. This plot along with the number of clusters measure helps the users with the trends of the search topic, like identifying the presence of a dominant subtopic in the tweets. While Wordcloud evaluates the most frequent keyword, this measure with the help of the cluster exemplars helps to identify the whole opinion rather than just the dominant word token. The following Table 4.2 shows the exemplar tweet of the largest cluster in the Seattle data set, providing the user with the context of the cluster rather than the just the keyword, #jobs

Table 4.2 Sample exemplar tweet

Data set	Seattle
Data set size	2000
Model	Dissimilar Initialization
Measure	T-Information distance
Cluster #	49
Cluster size	56
Exemplar tweet	Want to work in #Seattle, WA? View our latest opening: https://t.co/xJgtFafWkS #Nursing #Job #Jobs #Hiring #CareerArc

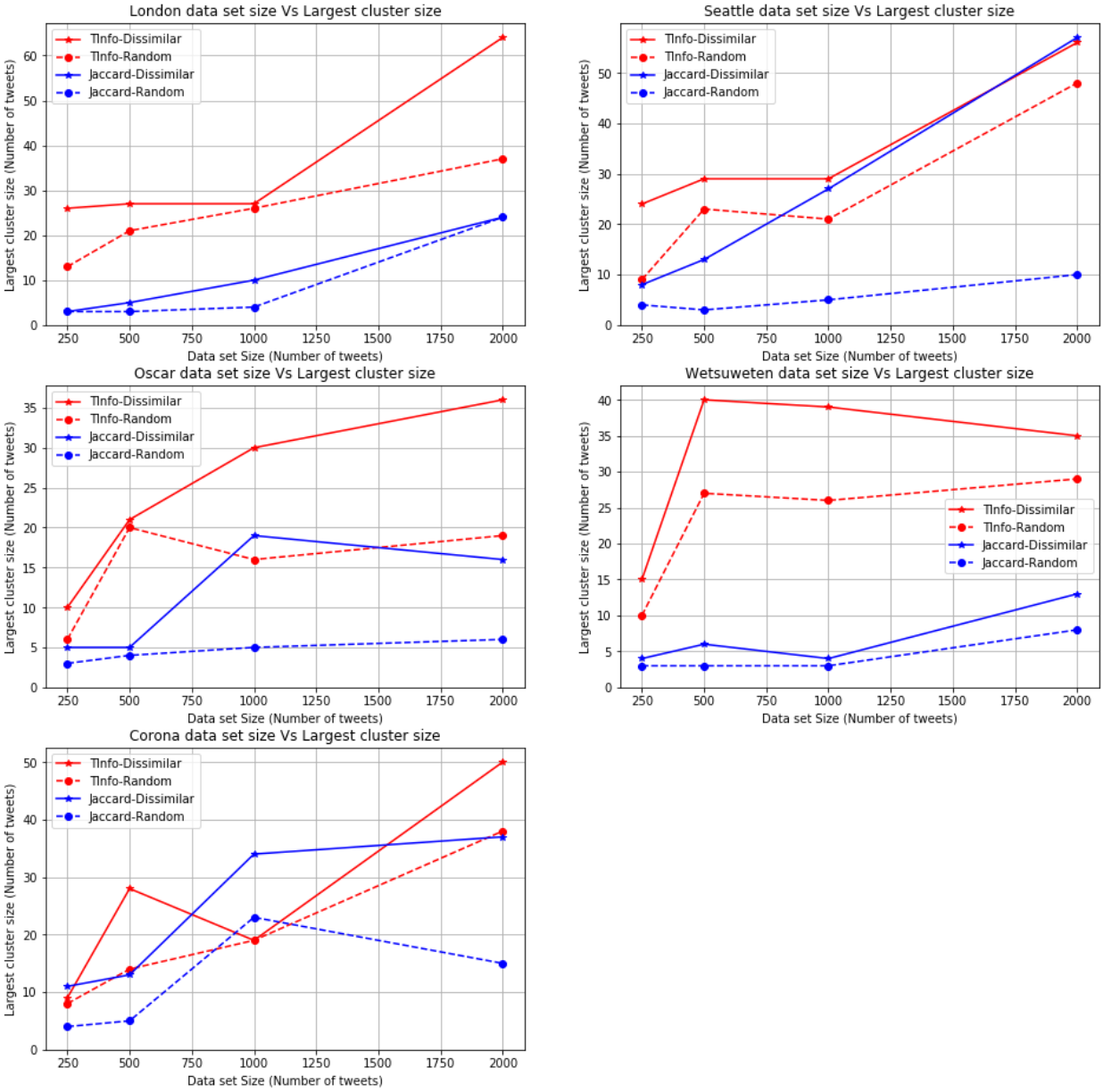

Figure 4.4 Data set size vs Largest Cluster size

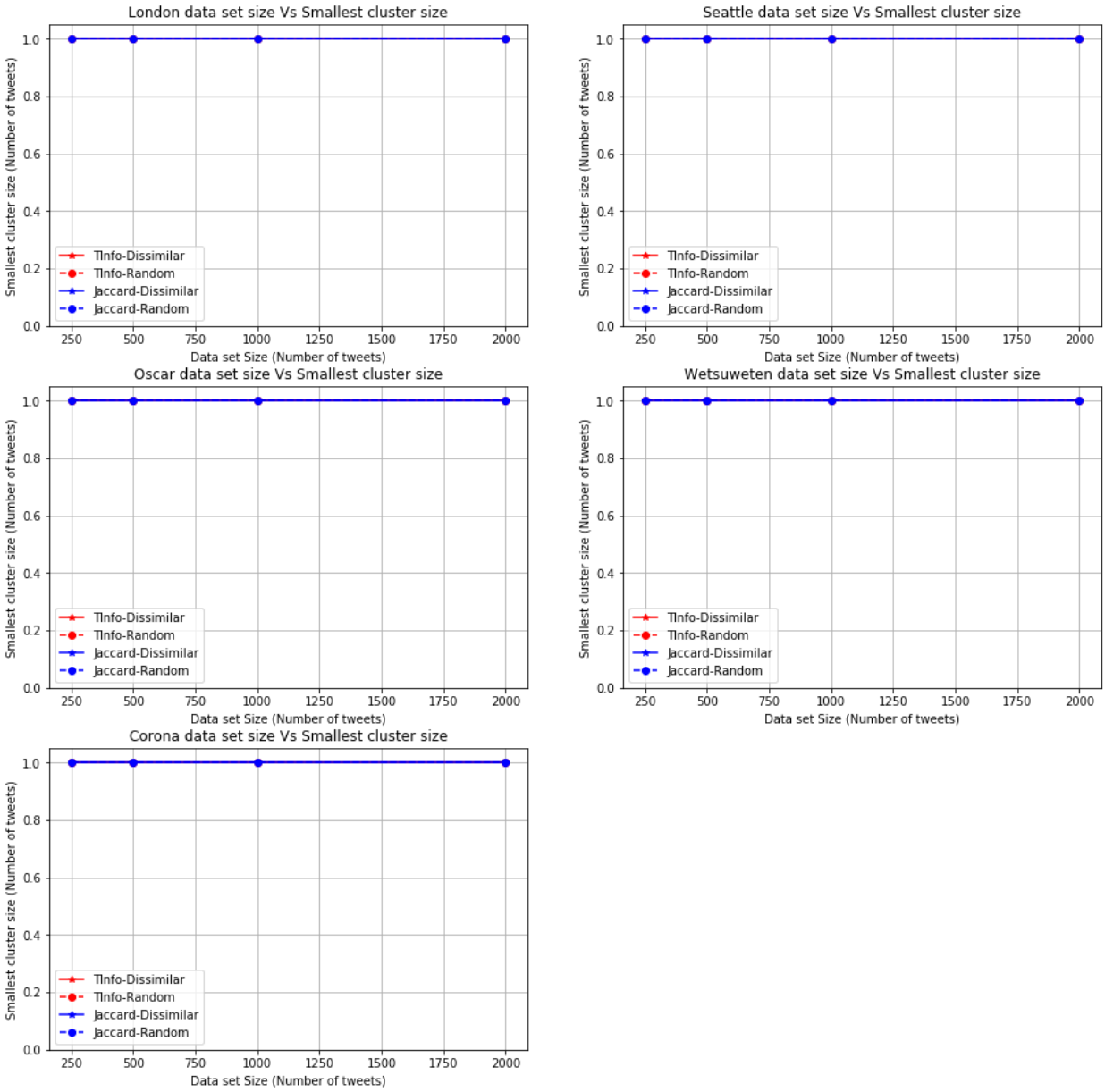

Figure 4.5 Data set size vs Size of smallest cluster

For a good pick of data in general, a high value of the largest cluster size would indicate a dominant sub-topic. By visual observation, from the Figure 4.4, the dissimilar method of centre initializations across both the distance models and all the topics returned larger clusters, and this is because these initial centres are more informed and are spread out wider than the ones in random initialization. For the size of the smallest clusters in Figure 4.5, all the models have at least one lone cluster. The number of lone clusters in a model is usually data-driven and highly affected by the pick of data.

4.1.5 Intra-Cluster Distance (MSE)

Figure 4.6 shows the plot of mean SSE distance at various data set sizes for all the data sets. The four variations of the model are depicted in each subplot and indicated on the legend. The sum of squares error is the objective function in the K-means algorithm and the centres are generated and adjusted to minimize the objective function. This is one of the most important measures to assess the model's performance. The mean SSE of a cluster is defined as the mean of sum of the squares of the distance of every tweet to the cluster centre. This measure gives the reader an estimate of how spread out the cluster is, lower the value of SSE indicates a dense(tight) cluster and a higher value of mean SSE indicates a spread-out cluster.

From the observations of Figure 4.6, we see that the SSE value is the least for the Jaccard-distance based models and the values for T-Information-distance based models are almost twice that of the Jaccard models. Although both values are low, the reason for the T-Information-distance based models to have higher SSE values is related to the number of clusters. Recalling the observations from the Section 3.6.5, as the number of clusters increases to the number of tweets, the SSE values keeps decreasing and finally reaches 0 at the instance where the number of clusters is equal to the number of tweets, resulting all the tweets forming lone clusters themselves.

From Figure 4.3, we observe that the number of clusters formed in the Jaccard-distance based models is much higher than the ones in T-Information models, thus resulting in lower mean SSE values.

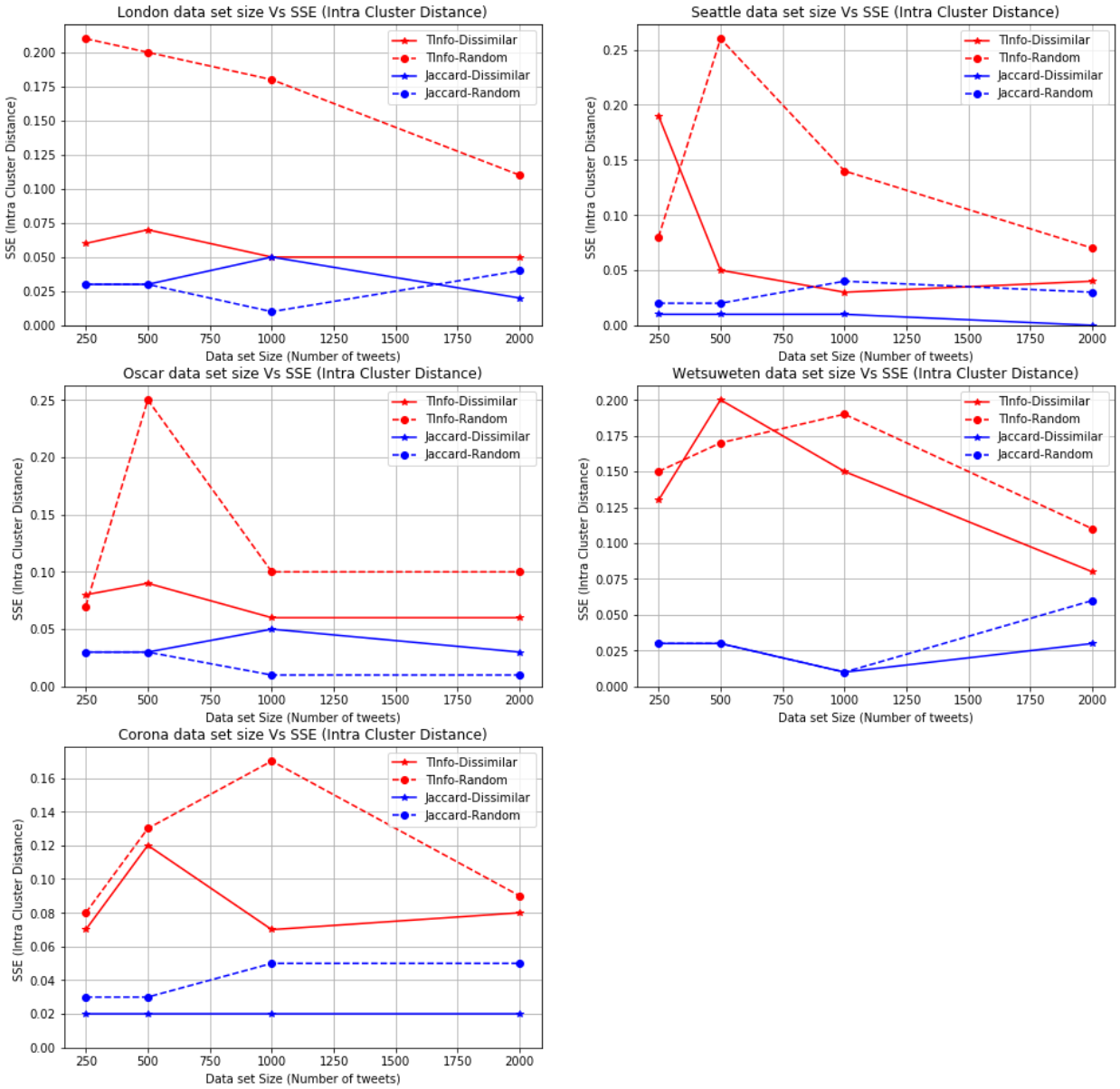

Figure 4.6 Data set size vs Intra Cluster distance

4.1.6 Inter-Cluster Distance

The inter-cluster distance of two clusters is the distance between two cluster exemplars or the cluster centres. Figure 4.7 shows the plot of inter-cluster distance for the size of the data sets. Each subplot shows the plots for the topic. The inter-cluster distance measure represents how different the evaluated subtopics are within a given topic. The value ranges from [0,1] as the distance between the centres is always in the range [0,1]. For a good clustering method, the inter-cluster distance is equal to 1.

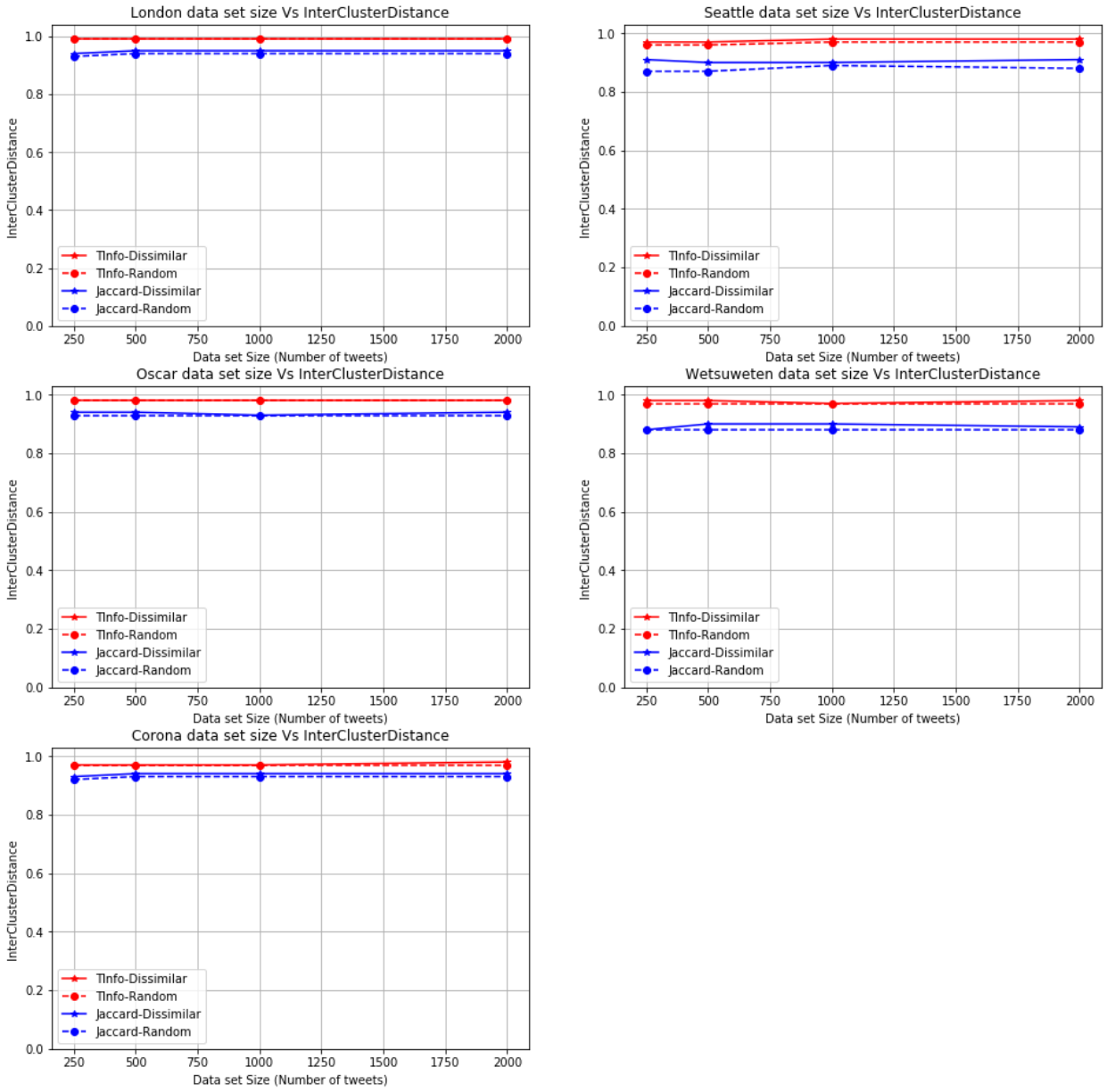

Figure 4.7 Data set size vs Inter-Cluster Distance

As discussed in Section 3.6.6, the goal of the clustering algorithm is to generate clusters in a way to minimize the intra-cluster distance and to maximize the inter-cluster distance. This is also the function being used to generate new cluster centres after the initialization as indicated in Equation 3.8. Figure 4.7 shows, both the distance measures have achieved an inter-cluster distance of greater than or equal to 0.9. While it shows the T-Information-distance based models have an inter-cluster distance value higher than the Jaccard models. Although the difference is small, It is interesting to note that the type of centre

initialization has indeed affected this value and shows that dissimilar initialization has achieved superior ICD measure than the random initialization.

4.1.7 Silhouette coefficient

Figure 4.8 shows the plot of the Silhouette coefficient to the size of the data set. Silhouette index of a cluster is defined as the ratio of the cluster's tightness(spread) to the inter-cluster distance of the closest cluster. The silhouette coefficient is the average silhouette index of all the clusters and is expressed mathematically in Equation 3.17.

This measure is one of the important performance measures as this evaluates the clustering quality and shows how good the tweets are clustered. Higher silhouette coefficient value indicates a higher quality of the clusters generated by the model. Good quality clusters are the ones with low spread (intra-cluster distance) and higher inter-cluster distances. It is interesting to note that the lone clusters contribute a value of +1 in calculating the overall silhouette coefficient. However, this is not acceptable, and hence to reduce the bias of lone clusters, they are not considered in the silhouette calculations.

Observations from Figure 4.8 shows the T-Information-distance based models have higher silhouette coefficient values and are close to +0.7. It is interesting to note that the silhouette scores of Jaccard-distances based models are also relatively high while the dissimilar initialization had the highest average silhouette coefficient values. Seattle data set have comparable Silhouette coefficient scores of around 0.7 for the Jaccard based models, implying that the Jaccard based models form high quality clusters when there are dominant sub-topics which can be seen in the Figure 3.9 and Figure 4.4.

As we recall from the discussion about the parameter from Section 3.6.7. that if silhouette index of a sample is close to 1, the sample is well-clustered and is assigned to a very appropriate cluster. This indicates the quality of clusters formed by the T-Information-distance dissimilar initialization is superior to the other three models.

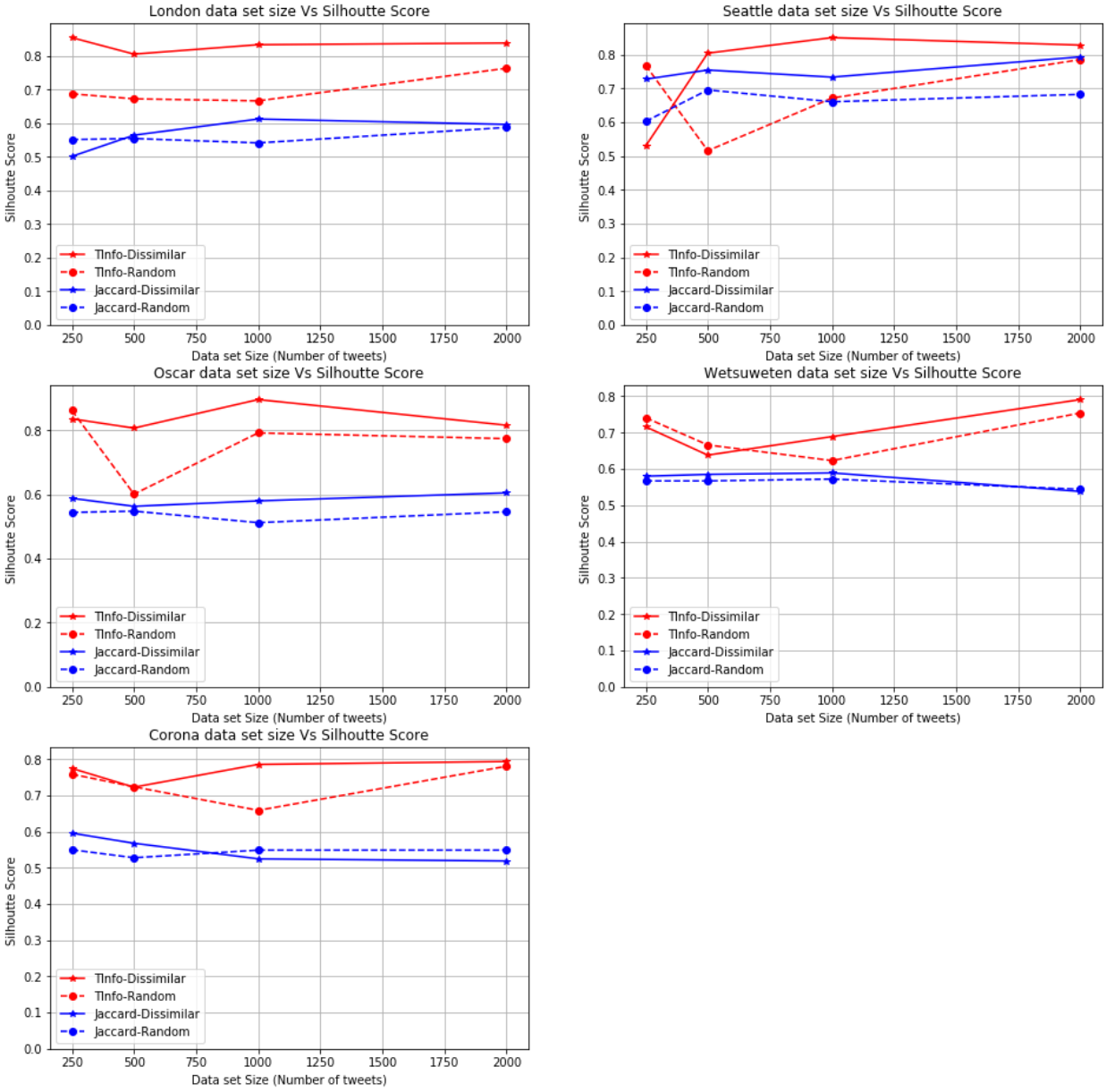

Figure 4.8 Data set size vs Silhouette Coefficient

4.1.8 Davies–Bouldin index

The Davies-Bouldin index (DB Index) is the ratio of intra-cluster distance to inter-cluster distance of each cluster to its most similar neighbour, averaged over all the clusters. The plot of the DB index to the size of the data set is shown in Figure 4.9.

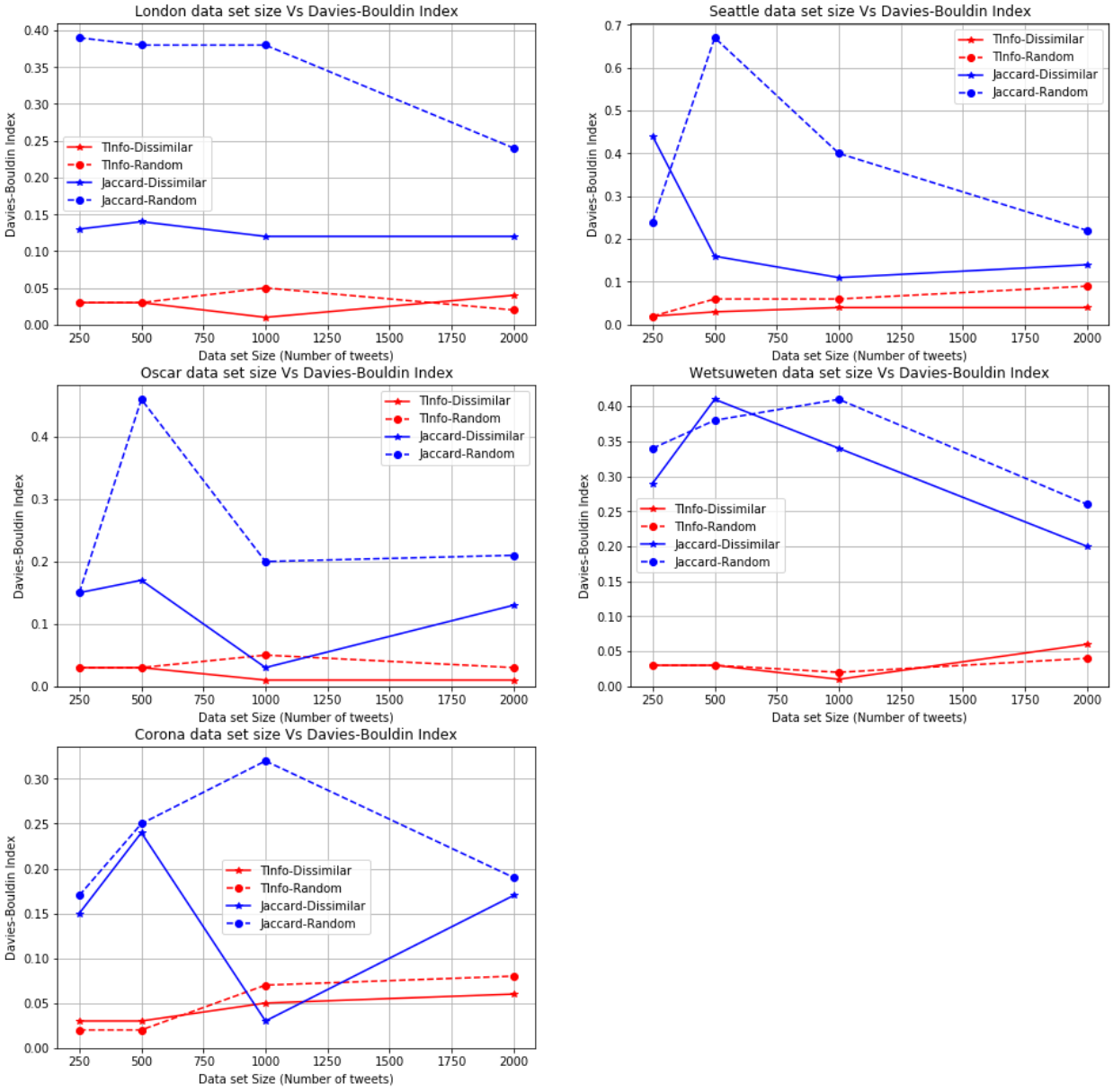

Figure 4.9 Data set size vs DB Index

Figure 4.9 shows that the T-Information-distance based models have the lowest DB index. This is one of the important measures in assessing the model's performance. It assumes the worst possible clustering of the two similar clusters. It is like the silhouette coefficient but also takes into consideration the spread of both the clusters and their inter-cluster distance. It is mathematically represented by Equation 3.19 and by analysis, the DB Index value of a good clustering algorithm must be as low as possible.

The plot in Figure 4.9 shows that the neighboring clusters formed by the T-Information distance models have a higher inter-cluster distance and a lower intra-cluster distance. This trend is observed, because of how these distance measure work. Jaccard word tokens are highly sensitive to grammatical errors and small omissions and hence assumes the tweet as a different sub-topic. The T-information for two tweets, X and Y can be computed by effectively compressing string X using the basis strings generated by a T-code decomposition of string Y. The information distance then becomes a function of how well one string's base strings compress the second string.

4.2 Chapter Summary

T-Information distance-based models achieved a relatively higher rate of the data compression around 25% on average more than the Jaccard based models, while also retaining a higher percentage of valid clusters. The results from Section 4.1.2 and Section 4.1.3 have shown that the number of lone clusters are higher in Jaccard based models than the T-Information models implying a low quality of clustering. This trend is observed due to the way these similarity measures calculate the distance, Jaccard measure focuses on exact word tokens to consider the tweets as similar. A minor omission or presence of unique URL could lead to a higher value of Jaccard distance and this further increase if the tweets being considered have lower number of word tokens as seen in Example 3.1 and Example 3.2.

Results from Section 4.1.4 support that using the dissimilar tweets as initial centres has resulted in large clusters. This trend was observed across both the models indicating that by using informed initial centres, the models were able to cluster better. This is evident by Figure 4.4 where the Jaccard dissimilar model was able to cluster the dominant sub-topic on par the T-Information based model. The quality of the clusters generated by the models is evaluated for inter, intra-cluster distances as well as the model's ability to correctly classify tweets. All the models achieved a good inter-cluster distance of at least 0.9. The T-Information dissimilar initialization has achieved higher values of the Silhouette coefficient of above 0.7 in most of the cases surpassing the other models and a lower value of DB Index indicating highest quality clusters. Analyzing the effects of initialization methods of T-Information models, the dissimilar initialization had better VCR measures and was able to better find trends in the topics, as evident in the Figure 4.4. Although the T-Information random initialization model has performed better than the both the Jaccard-based models, the mean SSE value was on average two times higher than the dissimilar model. Although the number of clusters in both the T-Information initializations were considerably close, the dissimilar model has shown superior clustering and had relatively lower number of lone clusters and a better Silhouette score.

Chapter 5 Conclusion and Future Work

With semi-automation in the social media industry along with the presence of a large user base, voluminous amounts of data are being generated every day thus warranting a need for effective filtering tools to filter out the redundant data. Most of the academic research has focussed on inter-topic clustering. However, this approach is less useful in industry which desires tools that can form clusters (sub-topics) within a topic.

The four common types of social media noise are discussed and in summary, Type II noise – Different contexts can be addressed by using more detailed keywords or using advanced custom search strings. Some of the sample advanced search strings supported by Twitter are presented in the Table 1.1. Type I noise - Non Informative results noise is difficult to filter out as it is highly subjective on the user's intentions, while Type III - Non-relevant content and Type IV – Bot generated posts tend to follow a pattern and the proposed algorithm can filter these redundant tweets.

We have proposed an effective way of forming clusters by using a modified K-means approach. Building on the conclusions of [17], [29] T-Information distance and Jaccard word token distance are used as similarity measures and we have proposed four models of the algorithm shown in the Table 5.1. These models include a traditional approach of a random selection of tweets as initial centres as well as using dissimilar tweets as the initial centres. The proposed algorithm generates new cluster centres by minimizing the sum of square error between the clustered tweets and their cluster centres, while also maximizing the distance between the cluster centres.

Table 5.1 Algorithm models

Model Name	Initialization vectors	Distance measure used
T-Information – Random	Randomly chosen.	T-Information distance
T-Information – Dissimilar	Dissimilar tweets based on T-Information distance measure.	
Jaccard – Random	Randomly chosen.	Jaccard distance
Jaccard – Dissimilar	Dissimilar tweets based on Jaccard distance measure.	

The performance and effectiveness of these models are evaluated by comparing the eight different performance measures, discussed in Section 3.6 on clustering the tweets from five different topics. The effect of data set sizes was explored to determine the impact on the model performance. The model is run on the topics, each varying from 250 to 2000 tweets to mimic the real-world application of clustering tweets obtained from a free tier Twitter endpoint.

Our results bolster the conclusions of Shameen et al.[29], that a K-means approach that uses a similarity distance and selects the most dissimilar tweets as the initial centres is superior and yields quality clusters than a traditional K-means clustering algorithm that randomly selects the initial centres. This holds true for our case of clustering tweets. We propose an alternative approach to the algorithm

presented in [17] which does not require a threshold value and produces high-quality clusters. Of the models, T-Information distance measure with the dissimilar initialization proved to be superior to the other models. This is supported by the results discussed in Chapter 4, as this model showed an average data compression rate of about 25% higher than the Jaccard distance-based models. This model also achieved an average Silhouette coefficient of about 0.7 across all the data sets, surpassing all the other models and had the least DB index value indicating the presence of high-quality clusters.

The T-Information distance-based models required a smaller number of computations and took less time to converge than the Jaccard models. These models achieved higher data compression rates while also achieving good VCR measures of around 20% to 50% on average across the topics indicating a relatively lower number of lone clusters than the Jaccard-distance based models. These values infer that the T-Information distance measure is superior and more robust than the Jaccard distance as a similarity measure on the tweets which often contains omissions and grammatical errors as they are usually based on human emotions and sentiments.

The future work include publishing this model where users can upload their own data to perform intra-set clustering or to feed the data directly from the Twitter API based on the user's API keys and returns the clustered data set along with the performance measures and analytics about the topic. Also expanding the data retrieval from other social media platforms like Reddit or Facebook. To further improve the data compression rate, we plan to implement a multi-layered clustering approach using both Jaccard and T-Information distances and testing it out on large scale data sets and the topics of high throughput.

Bibliography

[1] “Average Time Spent Daily on Social Media (Latest 2020 Data)”, https://cdn.broadbandsearch.net/blog/average-daily-time-on-social-media (accessed Jan. 25, 2020).

[2] “Demographics of Social Media Users and Adoption in the United States.”, https://www.pewresearch.org/internet/fact-sheet/social-media (accessed Jan. 25, 2020).

[3] “Introduction to Activism | Permanent Culture Now”, https://www.permanentculturenow.com/what-is-activism (accessed Feb. 28, 2020).

[4] “What is Hashtag Activism? - Definition from Techopedia”, https://www.techopedia.com/definition/29047/hashtag-activism (accessed Feb. 28, 2020).

[5] “Twitter”, https://twitter.com (accessed Apr. 02, 2020).

[6] "Ice Bucket Challenge dramatically accelerated the fight against ALS", https://www.als.org/stories-news/ice-bucket-challenge-dramatically-accelerated-fight-against-als (accessed Feb.28, 2020).

[7] “2016 presidential candidates differ in their use of social media to connect with the public”, https://www.journalism.org/2016/07/18/candidates-differ-in-their-use-of-social-media-to-connect-with-the-public (accessed Feb. 28, 2020).

[8] Phoenix7777, "*English: Twitter activity of Donald Trump from his first tweet in May 2009 to May 2018". Retweets are not included.* 2017., https://commons.wikimedia.org/wiki/File:Twitter_activity_of_Donald_Trump.png.

[9] Phoenix7777, *English: Twitter activity of Hillary Clinton from her first tweet in June 2013 to September 2017. Retweets are not included.* 2017, https://commons.wikimedia.org/wiki/File:Twitter_activity_of_Hillary_Clinton.png.

[10] “Social media in the United States presidential election, 2016 - Infogalactic: the planetary knowledge core” https://infogalactic.com/info/Social_media_in_the_United_States_presidential_election,_2016#cite_note-LangSocialMediaCircus-4 (accessed Feb. 28, 2020).

[11] L. Alexander, “Blame it on the Zodiac killer: did social media ruin Ted Cruz’s campaign?” *The Guardian*, May 04, 2016.

[12] Echosec, “Social Media Monitoring ” https://www.echosec.net/social-media-monitoring (accessed Feb. 28, 2020).

[13] "Dataminr Products Page" , https://www.dataminr.com/products (accessed May 05, 2020).

[14] “Using standard search", https://developer.twitter.com/en/docs/tweets/search/guides/standard-operators (accessed Jun. 25, 2020).

[15] “Sentiment Analysis”, https://monkeylearn.com/sentiment-analysis (accessed Feb. 06, 2020).

[16] “Recommender Systems in Practice” , https://towardsdatascience.com/recommender- systems-in-practice-cef9033bb23a (accessed Jun. 13, 2020).

[17] Jubinville," Evaluation of Intra-set Clustering Techniques for Redundant Social Media Content ", https://dspace.library.uvic.ca/bitstream/handle/1828/10438/Jubinville_Jason_MASc_2018.pdf (accessed, Feb. 28, 2020).

[18] “Counting characters”, https://developer.twitter.com/en/docs/basics/counting-characters (accessed May 06, 2020).

[19] “Character limits for each social network - Buffer FAQ”, https://faq.buffer.com/article/491-publish-character-limits (accessed Jan. 25, 2020).

[20] “Pricing – Twitter Developers” https://developer.twitter.com/en/pricing (accessed May 06, 2020).
[21] “Twitter Rate limiting” , https://developer.twitter.com/en/docs/basics/rate-limiting (accessed May 06, 2020).
[22] “Search Overview”, https://developer.twitter.com/en/docs/tweets/search/overview (accessed May 06, 2020).
[23] “Enterprise search APIs”, https://developer.twitter.com/en/docs/tweets/search/api-reference/enterprise-search (accessed May 06, 2020).
[24] “Introduction to Tweet JSON”, https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/intro-to-tweet-json (accessed May 06, 2020).
[25] P. Jaccard, “Étude comparative de la distribution florale dans une portion des Alpes et du Jura,” 1901, doi: 10.5169/SEALS-266450.
[26] R. W. Hamming, “Error Detecting and Error Correcting Codes,” *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, Apr. 1950, doi: 10.1002/j.1538-7305.1950.tb00463.x.
[27] E. S. Ristad and P. N. Yianilos, “Learning string-edit distance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 522–532, May 1998, doi: 10.1109/34.682181.
[28] A. Bookstein, V. A. Kulyukin, and T. Raita, “Generalized hamming distance,” *Information Retrieval*, vol. 5, no. 4, pp. 353–375, 2002, doi: 10.1023/A:1020499411651.
[29] 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*, Nov. 2009, pp. 1–6, doi: 10.1109/AHICI.2009.5340335.
[30] J. Brownlee, “A Gentle Introduction to the Bag-of-Words Model,” Machine Learning Mastery, Oct. 08, 2017. https://machinelearningmastery.com/gentle-introduction-bag-words-model/ (accessed Aug. 19, 2020).
[31] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, “Short text classification in twitter to improve information filtering,” in *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval - SIGIR ’10*, Geneva, Switzerland, 2010, p. 841, doi: 10.1145/1835449.1835643.
[32] Grossi, Roberto, and Giuseppe F. Italiano. "Suffix trees and their applications in string algorithms." Proceedings of the 1st south American workshop on string processing. 1993.
[33] 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*, May 2012, pp. 1–4, doi: 10.1109/ECTICon.2012.6254293.
[34] İ. Arın, M. K. Erpam, and Y. Saygın, “I-TWEC: Interactive clustering tool for Twitter,” *Expert Systems with Applications*, vol. 96, pp. 1–13, Apr. 2018, doi: 10.1016/j.eswa.2017.11.055.
[35] M. R. Titchener, “Digital encoding by means of new T-codes to provide improved data synchronisation and message integrity,” *IEE Proceedings E - Computers and Digital Techniques*, vol. 131, no. 4, pp. 151–153, Jul. 1984, doi: 10.1049/ip-e.1984.0028.
[36] N. Rebenich, “Fast Low Memory T-Transform: String Complexity in Linear Time and Space with Applications to Android App Store Security,” p. 96, 2007.
[37] Echosec Systems, “Terms Of Service”, https://www.echosec.net/terms-of-service (accessed May 04, 2020).
[38] “Tweepy” , Available: https://www.tweepy.org/ (accessed May 04, 2020).
[39] N. Rebenich, "Libflott", Available: https://github.com/*ardeego/libflott*. 2019.
[40] M. Anderson, "Libflott-Python", Available: https://github.com/*mike-anderson/libflott-python*. 2018.

[41] “Clustering - K-means Tutorial”, https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html (accessed Jun. 25, 2020).
[42] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987, doi: 10.1016/0377-0427(87)90125-7.
[43] D. L. Davies and D. W. Bouldin, “A Cluster Separation Measure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979, doi: 10.1109/TPAMI.1979.4766909.
[44] Rendón, E., Abundez, I., Arizmendi, A., Quiroz, E.M.: Internal versus external cluster validation indexes. Int. J. Comp. Comm. 5(1), 27–34 (2011)
[45] M. Hassani and T. Seidl, “Using internal evaluation measures to validate the quality of diverse stream clustering algorithms,” *Vietnam J Comput Sci*, vol. 4, no. 3, pp. 171–183, Aug. 2017, doi: 10.1007/s40595-016-0086-9.

Appendix A

Normalized T-Information distance function

This section discusses the evaluation of the sigmoid function used in the T-Information distance measure.

The sigmoid function, $f(x) = \frac{1}{1+e^{-(x+a)}}$ is used to smoothen out the tails of the T-Information distance measure and restrict the values to less than or equal to 1.

Let *x* be the T-Information distance measure and let *f(x)* be the normalized T-Information distance function and *a* be the tuning parameter, then based on the discussion from the Section 3.4.2, we know that this value should always be in the ranges [0.95,0.999]. The upper limit of 1 is approximated to 0.999 since, $\lim_{x\to 0} \ln(x) = \infty$

$$0.95 \leq \frac{1}{1+e^{-(x+a)}} \leq 0.999$$

$1.05 \leq 1 + e^{-(x+a)} \leq 1.001$

$0.05 \leq e^{-(x+a)} \leq 0.001$

$2.99 \leq (x+a) \leq 6.90$

$2.99 - x \leq a \leq 6.90 - x$

But we know the value of x ranges from [0.95,1.4], the value of 1.4 is the maximum observed value of T-Information distance during this research.

$$0.95 \leq x \leq 1.4$$

Lower Limit: Upper Limit:

$2.99 \leq a + 0.95$ => $a \leq 6.90 - 1.4$

$a_{min} \geq 2.04$ => $a_{max} \leq 5.5$

And thus, the variable parameter a in the sigmoid function, takes values from $2.04 \leq a \leq 5.5$

However, to account for instances where the T-Information value may exceed 1.4, the variable parameter '*a*' is set to the average of both the extremes, this the value of a is set to 3.77.

$$TInfo_{Norm} = \frac{1}{1+e^{-(TInfo_{dist}+3.77)}}$$

Where $TInfo_{dist}$ is the T-Information distance and $TInfo_{Norm}$ is the normalized T-Information distance. The following table shows the table of the normalized T-Information values for varied values of T-Information distance values.

Table A.1 Range of Normalized T-Information function

$TInfo_{dist}$	$TInfo_{Norm}$
1	0.9915
1.4	0.9943
14	0.9999
144	1