MICROBLOG SUMMARIZATION BASED ON SENTIMENT AND ASPECT ANALYSIS

by

Syed Muhammad Ali

B.Math, University of Waterloo, Waterloo, ON., 2012
B.B.A, Wilfrid Laurier University, Waterloo, ON., 2012

A thesis

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Science

in the Program of

Computer Science

Toronto, Ontario, Canada, 2016

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Microblog Summarization based on Sentiment and Aspect Analysis

Master of Science (MSc.), 2016

Syed Muhammad Ali

Computer Science

Ryerson University

ABSTRACT

On Twitter, the short nature of the post forces users to remain concise while conveying the main ideas to other users. Hence, the challenge is on how to use the unstructured texts to extract information that can be valuable for organizations. We investigate the best methodology to perform microblog summarization of topics discussed on Twitter. First, we classify the microblogs related to the topic into positive, negative, or neutral sentiments, and then we extract sub-topics (i.e., topic aspects), and pick the top *N* ranked aspects by sentiment temperature for final summarization. We utilize known algorithms for annotation, sentiment analysis, and clustering to determine which combination yields the best results. This paper attempts to address how sentiment analysis in conjunction with aspect extraction of topics can yield more effective summarization. Evaluation results show that sentiment analysis and aspect extraction improve the overall summarization of topics compared to baseline technique.

Acknowledgements

All Praise is to *Allah* (God) who guided me, taught me, and helped me in life. Without His help, it would not have been possible.

I want to thank my co-supervisors, Dr. Ebrahim Bagheri and Dr. Cherie Ding, for their guidance, patience, and support throughout my time at Ryerson. I am deeply grateful.

I also want to thank my family for supporting me in life. I am one of the most fortunate ones in this world. My parents and my sisters spent countless hours, volunteering to evaluate the tweets, clustering them, and then providing feedback. The evaluation was a huge undertaking.

I also want to thank Khuram for assisting me during the first semester of my Master's, when I needed his help while I navigated through the first term.

I also want to thank my lab partners, Hossein and Fattane. I have learned from your expertise and improved my technical skills greatly over the past year. Also, special thanks to Dante for being an amazing TA partner.

Finally, I am glad to have come to Ryerson and meet the incredible people here.

Read: In the Name of your Lord Who created.
Created man, out of a (mere) clot of congealed blood:
Proclaim! And thy Lord is Most Bountiful.
He Who taught (the use of) the pen.
Taught man that which he knew not.

- First Verses Revealed in The Holy Qur'an
(96:1-5)

Table of Contents

List of Tables

List of Figures

List of Appendices

Chapter 1. Introduction

The idea of microblogging occurred to Jack Dorsey of Odeo, Inc., when he and his team wanted to use the concept of Short Messaging Service (SMS) online, where a user can broadcast a message to anyone or a specific group of followers [Sagolla, 2009] . This idea led to the development of Twitter.com, which is now a publicly traded company. During the last three months of 2014, Twitter reported monthly active users of 288 million [Corporate Annual Report, 2014] . On Twitter, users can post a message online as long as it is within 140 characters, and the message can be shown to any other users. Some of the tweets (i.e., microblogs) posted by an influential or common user are *re-tweeted*, which increases the spread of a message. Usually, users re-tweet when they agree with an idea or find it interesting. Users could also "favourite" a tweet to show their interest towards it, which is a feature in Twitter marked with a "star" icon. Re-broadcasting or marking a tweet as favourite are signs that a tweet is structurally sound and semantically coherent. Even if users do not re-tweet a particular post, they may post something on their own that may contain semantically similar content.

Using the re-tweet and favourite attributes of a tweet, we could find out what users *think* about a particular topic. Moreover, we could also determine sentiment of a tweet and group it with other similar sentiments (e.g., positive, negative, or neutral) of the topic, and then extract knowledge on the entire sentiment to determine *why* users think that

way. In order to do this, automated tools that streamline opinion mining tasks and perform natural language processing are required.

In this thesis, we attempt to address the issue of automatically summarizing microblog posts based on two major stages: sentiment analysis and aspect analysis. Sentiment analysis refers to classification of tweets as positive, negative, or neutral. Aspect analysis refers to understanding why the topic has the associated sentiments. For example, a topic about *Toyota Lexus* could have positive, negative, and neutral tweets associated to it. In each of the sentiment buckets, one or more of these three aspects could exist: fuel-efficiency, comfort, and price. Concretely, aspect analysis means breaking a group of tweets related to a sentiment of a topic into subgroups. These subgroups are the aspects, which could refer to themes, point of views, or identifiable and distinct features of the main topic. A good summary will allow organizations to discover trends, opinions, and further their organizational goals. We believe that not only will sentiment combined with aspect analysis provide more effective summaries, but it will also provide insight on controversies that could exist about a particular topic.

1.1. **Motivation**

We want to know if there is a better way to extract summaries from microblogs associated a particular topic. To the best of our knowledge, no one has used sentiment analysis on tweets as a part of their summarization process. Furthermore, we also do not know of any work that uses Word Graphs, which is a graph of word co-occurrences based

on [Ohsawa et. al., 1998], to extract aspects of topics before running their summary algorithms.

Much research has been conducted in the summarization of microblogs. Microblog summarization is useful when we need to know a snapshot of trending topics on Twitter. These trending topics can either be represented by a hash-tag (#) or a phrase. In both instances, the hash-tag or phrase would exist in the tweet. If these topics can be summarized effectively, it will save invaluable time for the reader. Furthermore, a running timeline can be made, with a summary of the topic for different periods. This could be useful for analysis of topic evolution. For instance, sentiment-based aspect analysis will show if certain aspects about a topic fade away, or new ones emerge, and if the overall sentiment about the topic is both positive and negative--generating controversy--or largely neutral.

1.2. **Approach Overview**

Our proposed methodology (Figure 1 and Appendix C) addresses the lack in literature on whether incorporating sentiment analysis and aspect extraction to group tweets *first* will help the actual summarization algorithms. Our methodology requires knowing the topics to search for. We can search for topics through a phrase or a hash-tag search on Twitter. Alternatively, we could use an annotation system like TagMe [Ferragina et al., 2010] to determine all semantically related tweets to a particular topic. In step 2, we utilize the annotation approach in finding related tweets to a topic.

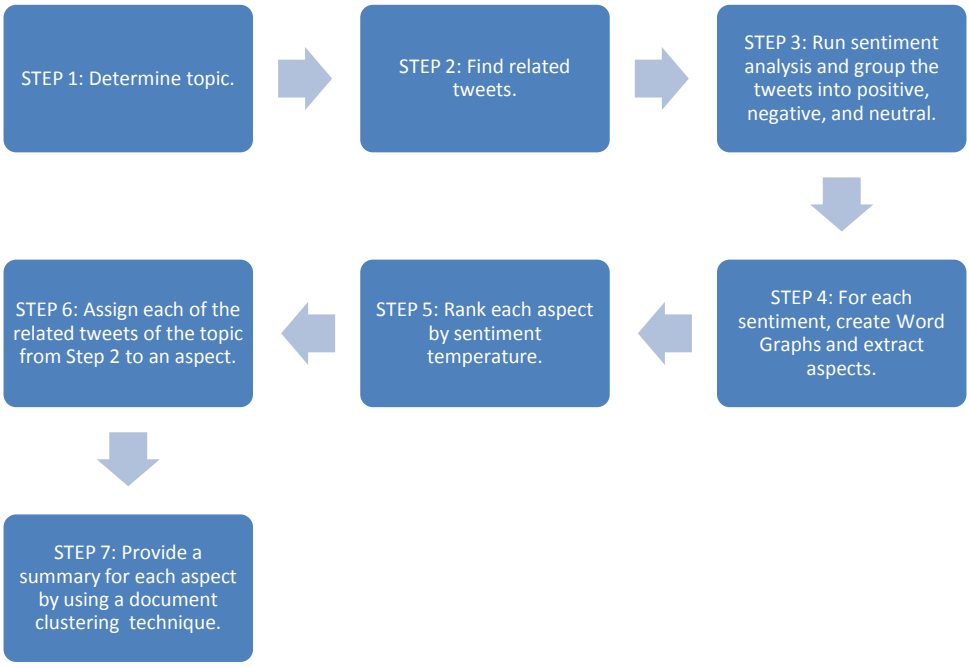

Figure 1: Our proposed methodology

The third step is to perform sentiment analysis on the topics so that we can group the tweets into positive, negative, and neutral buckets. By grouping the tweets according to sentiment, we can ensure that each sentiment is represented in the final summary, which may not have occurred without performing sentiment analysis first. For example, consider a topic that is dominated by highly positive tweets and few negative tweets. A straight summarization technique may only use representative documents that are positive, and negative documents will most likely not appear in the summary. Thus, sentiment analysis helps us develop summaries that consider all sentiments.

In order to perform the sentiment analysis phase, we developed an ensemble of sentiment classifiers to perform the task, which would take an average sentiment score based on two separate classifiers. The reason why we used an ensemble is because we observed that when the classifiers are used independently, they do not always provide the correct

sentiment. One of the classifiers uses a Multinomial Naive-Bayes (MNB) method to classify tweets as positive or negative. The second classifier utilizes recursive neural networks to analyze the structure and parts-of-speech of the sentence in order to provide a positive, negative, or neutral classification.

Once tweets are grouped into positive, negative, and neutral tweets, the fourth step is to create a Word Graph (WG) for each sentiment of a particular topic. The graph models co-occurrences of words occurring in the tweets. The WG is used to create aspects of the topic using an appropriate graph clustering technique. An aspect could represent theme, argument, or characteristic of the topic sentiment. The graph clustering algorithm that we used was Multi-level Clustering (MLC) [Blondel et. al, 2008]. We compared MLC with other graph clustering algorithms using variance of information (VI) [Meila, 2007] and split-join-distances [van Dongen, 2000] to conclude that MLC is the best graph clustering technique to extract aspects.

The fifth step is to rank each aspect according to some criteria. We chose to rank aspects according to the sentiment "temperatures". The overall sentiment temperature of aspect *p* is the average temperature of the sentiment class that was chosen in Step 3 for all tweets of the Word Graph. More details are provided in Section 4.2. Now there could be other ways to rank aspects, such as by the number of tweets associated, but we chose to rank by sentiment temperature instead, which will order the aspects by their dominant temperatures.

The sixth step is to re-assign each of the related tweets of the topic from Step 2 to one or more of the aspects that we extracted. This was done by finding top 1/3 of tweets with maximum overlap between the words of the aspect and the words in the tweets.

Finally, the seventh step entailed using document clustering algorithms for the summarization of each aspect. Once tweets are assigned to each aspect, a document clustering algorithm is used to group similar tweets together before selecting a representative tweet for each group. The three primary document clustering techniques that we investigated were Agglomerative, Bisect K-Means++, and Hybrid TF-IDF [Sharifi et al., 2014].

1.3. **Contributions**

In this thesis, we investigated whether aspect extraction of topics improves the topic summaries. Aspect extraction is done through the clustering of Word Graphs, which are graphs of word co-occurrences in all microblogs associated to a particular topic. Once the aspects are extracted, document summarization algorithms are used to obtain a summary for each aspect.

Our main contribution in this work is that grouping tweets of a topic by sentiments first, and then applying Word Graph construction to extract aspects of the topic improves overall document summarization. Therefore, we can learn about the topic by focussing only on the summaries of its most important aspects for each sentiment category.

1.4. Thesis Organization

Chapter 2 of the thesis talks about related work in the area of Semantic Web and microblog summarization. The ideas presented in the related work are foundation to natural language processing in social media contexts.

Chapter 3 discusses the background of our work. It talks in detail about sentiment analysis, graph clustering, and some document summarization techniques.

Chapter 4 discusses the experimental setup and evaluation. We discuss on how we planned to evaluate summaries using human judgement and automated means.

Finally, Chapter 5 shows the evaluation results and Chapter 6 provides concluding remarks to the thesis.

Chapter 2. Related Work

In this chapter, we present some of the related work in the field of microblogs summarization and semantic extraction from social media. The related work sets up a foundation for our own work described in Chapter 3.

2.1. Named Entity Recognition and Annotation

Entity recognition in microblogs has been researched extensively for better understanding the semantics of a post; they are helpful in summarizing tweets, generating user profiles, or inferring user interests.

For example, Twitter's Lists feature can be used to obtain topical interests of a user [Bhattacharya, 2014]. The description inside Twitter Lists is used to extract topics through named entity recognition. In another paper [Michelson, 2010], Wikipedia was used as a knowledge-base to find entities and high-level categories for each tweet. The idea is to create a set key-value pairs for each tweet. The key will be a "discovered entity" that is a proper noun or non-stop word and the rest of the tweet is considered to be the value (except the discovered entity). Thereafter, the Wikipedia database is queried for related articles for each key/value pair. The "best" candidate article for each entity is selected and it is the one that has the highest overlap of tweet context (value) with that of the Wikipedia article. Once the article is mapped for each entity, the Wikipedia

folksonomy (i.e., a user-generated tagging system to online items) is used to generate a tree of all categories up to five levels deep. Then a ranking function is used to obtain the best top categories across all trees. The solution did not use hash-tags nor embedded URLs.

User interests can be obtained from the Wikipedia Category Graph (WCG) [Kapanipathi, 2014] by linking entities from tweets to a category in WCG algorithmically. Those categories that are linked to multiple entities receive a higher weight, and this indeed proved to be the best method. This is called the "Priority Intersect" method, and the terminology of this common category may be referred to as "Lowest Common Subsumer". This was similar to Twopics algorithm [Michelson, 2010], and it did not take into account any temporal aspects.

Similarly, a graph based framework called KAURI creates a dictionary *D* of key/value pairs by leveraging the four structures of Wikipedia: Entity page, Redirect page, Disambiguation page and Hyperlink in Wikipedia article [Shen, 2013]. Each key is the entity in the tweet and the values are possible mapping entities in Wikipedia. Then a graph G, for each Twitter user, is created to represent all the interdependence information between each of the mapped entities. Each mapped entity node has user interest score and each edge between the nodes shows the topical relatedness. The user interest is initialized by taking a weighted average prior probability, context similarity score, and topical coherence score. The final interest score is computed by iterating through a weighted average formula that traverses through the graph matrix and updates the values until a

certain cut-off point. Once the final interest score is computed for each value, the one with the highest score is mapped to the key.

In another paper, entity meanings are resolved by leveraging edits that the user has made on Wikipedia articles [Murnane, 2013]. The idea is that a user will be interested in those topics for which he makes Wikipedia edits, hence, the tweets will be also about the same set of topics. The usernames of both sites are matched by making a simple string matching, so the method does not always work. It's a graph-based algorithm that computes cosine similarities between direct categories from edited articles and indirect categories from inferred articles from tweets. Furthermore, a content similarity is computed by measuring the TF-IDF of article descriptions and entity meanings (properties of candidate articles).

TagMe [Ferragina et al., 2010] is a novel solution that uses Wikipedia anchor texts to disambiguate very short text, which could be poorly composed. It's a very fast system that can provide on-the-fly hyperlinks to Wikipedia articles for disambiguated entities. It also assigns a relatedness score, which is a measure of an entity being disambiguated correctly based on overlap between the Wikipedia anchor texts and the tweet itself, as well as other entities that are found in the tweet. The authors reported that TagMe yielded an F-Measure of about 78%, with possibility to balance precision (up to 90%) vs. recall (up to 80%).

2.2. **Determining Semantics in Tweets**

How different modelling strategies affect personalization and how temporal patterns affect recommendation quality has been explored [Abel, 2011] using profiling methods, enrichment methods, and temporal effects on personalization. The results showed that entity-based profiling, with enrichment from news articles had a higher mean reciprocal rank (MRR) than topic-based profiling.. In another paper [Abel, 2011], it is discussed how the actual link between tweets and news articles are made. When a URL is provided in the tweet, there are two linking strategies. Strict URL-based strategy is "linking" an embedded hyperlink in a tweet to the tweet itself if the hyperlink refers to a select news publisher. A Lenient URL-based strategy is "linking" a *reply* of an original tweet to an online news article of a select publisher if the original tweet contained the hyperlink of the same news article.

When a URL is not in a tweet, the tweet is linked to a news article from the web if the TF-IDF score is the maximum among all possible tweet / news article pairs. This is accomplished by using one of the following three strategies. A bag-of-words strategy is a comparison between a vector of words in a tweet and vector of words in a title of the article. A hash-tag based strategy is a comparison between a vector of hash-tags in a tweet and vector of words in an article. An entity-based strategy is a comparison between a vector of words in a tweet and vector of entities in a news article. The results showed that the entity-based in combination with lenient-URL based strategy had high coverage of tweets and precision.

The *Root-Path-Degree algorithm* developed by [Al-Kouz, 2012] finds the most representative sub-graph that reflects the implicit interests of the user. The idea is to create two types of graphs; the first graph is based on the entities extracted from user posts and the second graph is based on entities extracted from replies to the user's posts. The edges in each of the two graphs show semantic relationship between nodes, weighted by the frequency of semantic occurrences. Then, for each node in the graph, synonyms are extracted from WordNet and for each synonym topics are extracted from Freebase. The semantic relationships to other topics are generated and for each new relation the edge is added to the appropriate graph. Once the graphs are updated, the root node is set as the one with the highest out-degree. The weight of other nodes is the product of out-degree of the node and number of paths to root from the node. Each node of the final sub-graph that has coherently related topics will have a path to every other node in the graph, where the path does not include the root node. The total weight of all nodes in the selected sub-graph will also need to be the highest among all candidate sub-graphs.

A topic can be described as a set of weighted concepts where a concept, *c*, may be represented via named-entity or hash-tag [Abel, 2011]. It appeared that if a user is an *early adopter* then he would likely be interested in the topic in the long term. The recommendation algorithm used a cosine similarity measure between the Web resource (URL) and the profile vector. Using a ground truth set containing re-tweets with URLs, 1619 sampled profiles were created with weighted vector of interest for each user. Then a candidate set of URLs was created that were re-tweeted by various test users. It appeared

that those users that are active on Twitter have better profiles created based on hash-tag information, whereas sporadic users have better profiles created using entity recognition.

2.3. Microblog Summarization

For general text summarization, we found two foundational approaches. The first approach takes a snapshot of all data within a specific time period and provides a summary. The second approach provides an evolving summary based on change in data overtime. The two approaches are discussed below, although our thesis utilizes the snapshot approach.

2.3.1. Summarization based on Corpus Snapshot

Redundancy can be removed from summarization using three methods. Maximal Marginal Relevant (MMR) [Goldstein et. al, 2000], clustering [McKeown et. al., 1999], and Maximum Coverage (MC) [Filatova et. al., 2004]. In MMR, the redundancy is based on overlap between a candidate sentence to be added to the summary, and those sentences that are already in the summary. In clustering, the sentences are grouped together according to their similarity before a representative tweet is selected from each group. In MC, the sentences that cover the most concepts (i.e., information) are selected, and then a greedy algorithm is used to determine which sentences should be selected for final summary.

The baseline for us will be [Sharifi et al., 2014] who determined that Bisect K-Means++ with Hybrid TF-IDF is the best clustering method for summaries. Sharifi compared his

method with various other summarizers, including MEAD [Radev et. al, 2004], LexRank [Erkan et. al, 2004], TextRank [Mihalcea et. al, 2004], SumBasic [Nenkova et. al, 2005], random summary, and summary based on most recent microblogs. In Section 3.4.3, we will discuss in detail the theory behind Hybrid TF-IDF.

SumBasic [Nenkova et. al, 2005] uses simple probability distribution of words in the dataset. Each sentence is then assigned a weight equal to the average probability of the words in that sentence. In the end, the sentence with the highest probability is picked and every word in that sentence is updated to a reduced probability, to ensure that subsequent sentences with similar words are not picked again. One of the features of SumBasic is that the algorithm tends to favour longer sentences, as they likely contain higher average probabilities. In turn, the longer sentences also increase the overall recall, as noted by [Sharifi et al., 2014].

TextRank [Mihalcea et. al, 2004] uses the PageRank algorithm to rank important keywords of n-grams in a corpus. Thus, their summaries appear to be short snippets of a corpus. In LexRank [Erkan et. al, 2004], a modified cosine-similarity equation is used to construct an adjacency matrix with values of the said similarity between two sentences. This matrix is treated as a Markov chain, and a simple iterative algorithm is used to compute the stationary distribution. Each value of the stationary distribution represents a weight for the corresponding document, and the one with the highest weight is chosen to represent the summary.

For all the methods mentioned above, none of them used sentiment analysis to see whether their final summaries can be further improved. The methods also did not investigate whether aspect analysis will also refine the summaries. Our methodology addresses this void and we chose Sharifi's Bisect K-Means++ and Hybrid TF-IDF as the baselines, because Sharifi et al. used all of the methods above for comparison and determined that Hybrid TF-IDF is the best summarizer, and Bisect K-Means++ is the best clusterer.

2.3.2. Summarization based on Topic Evolution

TwitInfo is a system by [Marcus et. al, 2011] developed that produces summaries on only peaks of high tweet activity. Their streaming algorithm is applicable in journalism, as high tweet activity for a short period of time occurs when news breaks out about an event. The tweets are based on similarity to searched keywords and sorted from most to least similar. The main contribution of the authors was to display the summaries in a timeline.

Another example of summaries based on topic evolution is described by [Chakrabarti et. al., 2011]. The authors discuss how events can be sub-divided into smaller events using Hidden Markov Models (HMMs). They assert that HMMs are useful in detecting "bursty" events by looking at a peak in tweets in a small timeframe, and are also able to learn differences in language models of sub-events automatically. But these are useful when a training set can describe change in events; otherwise, it is difficult to use this method effectively.

Similarly, real-time "bursty" events can be modelled and used with the Phrase Reinforcement algorithm [Sharifi et al., 2014], [Nichols et. al, 2012] to rank sentences and provide real-time event summaries.

Chapter 3. Background

In this chapter, we will discuss the background of our work. Our work utilizes sentiment analysis, graphs, and clustering techniques. We will explain each of these components in detail and accompanying reasons for usage.

3.1. Sentiment Analysis

To the best of our knowledge, sentiment analysis has not been done on tweets for summarization purposes. Interest in finding sentiments on short-text has been researched for many years now. A 2013 survey [Saif et. al, 2013] noted that researchers have approached this topic in a wide variety of ways. There are two main issues with sentiment analysis: the subjectivity factor and a lack of a common gold standard. The survey noted that some datasets for sentiment evaluation are based on highly specific political issues [Diakopoulos et. al, 2010], [Asiaee et. al, 2012], [Speriosu et. al, 2011].

Researchers have attempted to analyze sentiments by providing a simple polarity of positive or negative, mixed polarity with the addition of neutral or irrelevant, and also a specific score within a range, typically between -5 (very negative) and 5 (very positive) [Saif et. al, 2013]. The classification methods included Naive Bayes, maximum entropy, and support vector machines by using n-grams of 1 to 3 words. A more recent and advanced method used recursive neural networks (RNNs) [Socher et. al, 2013]. We will

discuss the theory of Naive Bayes and Recursive Neural Networks because they are used in the sentiment analysis of our research.

3.2. **Multinomial Naive Bayes**

Naive Bayes is one of the most basic and simplest method to perform sentiment analysis, and it performs competitively against other classification tasks [Huang et. al, 2003]. Because of its low computational cost and small training data requirement, it is a favourable methodology for implementation.

Multinomial Naive Bayes (MNB) is used when it is important to take into account dependency between words that exist in the document. Given a document *d*, the probability that it is assigned a sentiment class *c*, is computed as follows:

$$\begin{aligned} c_{map} &= \arg\max_{c \in C} P(c \mid d) \\ &= \arg\max_{c \in C} P(d \mid c) P(c) \end{aligned} \tag{1}$$

where c_{map} is the *maximum a posteriori (MAP)* class. Concretely,

$$\begin{aligned} c_{map} &= \arg\max_{c \in C} P(c \mid d) \\ &= \arg\max_{c \in C} \hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(t_k \mid c) \end{aligned} \tag{2}$$

where t_k is the k^{th} term in a document *d* of length n_d, without stop words. In order to estimate probability *P* for a certain class *c*, we can estimate $P(t_k|c)$ and $P(c)$ by taking the

maximum likelihood estimate (MLE), which is a simple ratio of number of documents in class *c*, N_c, and total number of documents, *N*:

$$\hat{P}(c) = \frac{N_c}{N} \tag{3}$$

The conditional probability of *P* can be estimated as the relative frequency of term t in documents belonging to class c:

$$\hat{P}(t \mid c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}} \tag{4}$$

where *V* is the vocabulary of words from the training documents, T_{ct} is the number of occurrences of *t* in training documents from class *c*, including multiple occurrences of a term in a document.

One of the issues with MNB is that if in the test data there are instances of documents (or n-grams) assigned to a particular class, for which no such assignment exists in the training data, then c_{map} will be 0 for that class. This is the case even if in the test data, there is strong evidence that a particular n-gram should belong to a particular class. There is no workaround for this and this is a limitation of MNB, since the training set is sparse and cannot take into account of every (unlikely) event that could possibly occur.

In order to avoid computational problems, we can use Laplace smoothing. Hence, $\hat{P}(c)$ can be interpreted as:

$$\hat{P}(t \mid c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'}) + B} \tag{5}$$

where $B = |V|$ is the number of terms in the vocabulary. The Laplace smoothing ensures that each term is given a probability for each class. The algorithm in Figure 2 shows how to train a MNB classifier.

3.2.1. Feature Selection for Multinomial Naive Bayes

A popular feature selection method is the Chi-Squared test, which tests for independence of two events A and B. In our context, this means if a term and class occur together frequently, then they could be dependent.

The formula for testing the null hypothesis, which is a term and class are independent, is:

$$X^2(D,t,c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \quad (6)$$

where e_t is a binary value to show if a document contains term *t*, and e_c is a binary value to show if a document is in class *c*, *N* is observed frequency, and *E* is the expected frequency in *D*, which is the training set of labelled documents [Manning et al., 2008].

TrainMultiNomialNB(C,D)
Input: Set of classes, *C*; set of documents, *D*
Output: Set of vocabulary *V*; prior probabilities *prior*; conditional probabilites *cond*

```
1   V <- ExtractVocabulary(D)
2   N <- CountDocs(D)
3   for each c ∈ C
4   do N_c <- CountDocsInClass(D,c)
5   prior[c] <- N_c/N
6   textc <- ConcatenateTextOfAllDocsInClass(D,c)
7   for each t ∈ V
8       do T_ct <- CountTokensOfTerm(text_c,t)
9   for each t ∈ V
10      do condprob[t][c] <- (T_ct + 1) / Σ_{t'∈V}(T_ct' + 1)
11  return V, prior, condprob
```

ApplyMultiNomialNB(C,V, prior, condprob, d)
Input: Set of classes *C*; vocabulary of words *V*; prior probabilities *prior*; conditional probabilities *condprob*; and document *d*
Output: class with highest score

```
1   W <- ExtractTokensFromDoc(V,d)
2   for each c ∈ C
3   do score[c] <- log(prior[c])
4   for each t ∈ W
5         do score[c] += log(condprob[t][c])
6   return argmax_{c∈C} score[c]
```

Figure 2: The algorithm for Multinomial Naive Bayes. [Manning et al, 2008]

The formula can be re-written as:

$$X^2(D,t,c) = \frac{N(N_{11}N_{00} - N_{10}N_{01})^2}{(N_{11}+N_{01})(N_{11}+N_{10})(N_{10}+N_{00})(N_{01}+N_{00})} \quad (7)$$

A high score of X^2 indicates that the null hypothesis should be rejected. In other words, the occurrence of a term and class are dependent on each other. Similarly, for a common stop-word we can expect a low score for X^2, indicating independence between the stop-word term and the class. Therefore, if the word is dependent on a class, then it is selected as a feature for text classification.

3.2.2. Recursive Neural Networks

MNB is a competitive methodology for classifying documents with a sentiment class. However, it comes with some inherent limitations. For example, it does not take into account the position of each word in the *sentence*, does not know how to classify a sentence that is seen for the first time, and also does not take into account the sentence structure. For example, the following sentence contains positive words but is negative overall: *"My computer is extremely fast yet it is neither cheap nor light."*

Stanford's Sentiment Treebank (ST) [Socher et. al, 2013] utilizes recursive neural networks, and includes fine grained sentiment labels for 215,154 phrases in the parse trees of 11,855 sentences. It accurately captures the effects of negation, and is able to determine sentiments of phrases within a sentence.

The main idea of ST is the usage of trees. A sentence is broken down into a binary tree that is constructed based on parts-of-speech tagging. For example, if a sentence contained the word *not*, then the word *not* will be a left node of the tree and all the other phrases will be broken down in the right node of the tree because *not* negates subsequent words. The sentiments at each level are propagated up to come up with an overall sentiment score. In order to develop a powerful function that can aggregate meaning from child nodes more accurately than separate input specific functions, the authors proposed *Recursive Neural Tensor Networks (RNTNs).* The idea is to use tensor-based composition for all nodes and extend the idea of Recursive Neural Networks. The reader is referred to the paper for the technical details, which are arduous to be mentioned here.

The authors stated that the RNTNs proved to be an effective means of classifying sentences with sentiments, and were excellent performers over most *n*-grams, (where *n* is the size of a sentence).. RNTNs also cover the problem of negating positive or negative sentences. However, the system relies on the fact that the sentence is well-structured. Because tweets are not always well-structured and sound, we used an ensemble that combined both MNB and RNTNs. Further discussion is found in Section 4.2.

3.3. **Word Graphs and Graph Clustering Techniques**

Microblogging on Twitter forces users to broadcast a message in 140 characters or less. This forces the users to be concise and to the point. In order to cluster similar tweets together, a number of approaches have been researched. These include using hierarchical clustering methods, term-frequency analysis, and tweet attributes analysis such as favourite and re-tweet counts.

To the best of our knowledge, no one has applied the idea of constructing a *Word Graph* that creates a graph of word co-occurrences and then clustering the graph for summarization. Our goal in this research is to determine if creating Word Graphs to induce aspects of a topic will improve the overall summarization process. This idea is the child of the famous KeyGraph paper by [Ohsawa et. al., 1998]. We believed that making a graph of word co-occurrences will greatly enhance the final summarization of a topic. The reason for this is because a topic could have multiple aspects or themes associated to it. These aspects or themes could be discovered by direct clustering mechanisms (see Figure 4). However, by directly applying the clustering methods we may lose information pertaining to tweet attributes such as favourite counts. What we propose is constructing a Word Graph where each edge has a weight as a sum of total number of times the two words have co-occurred in various tweets and total number of times those same tweets have been "favourited".

In other words, for words *a* and *b* existing in a corpus of tweets:

$$edge\_weight_{ab} = \sum_{\{a,b\} \in t} (favourited_t + 1) \quad (8)$$

where *t* is a tweet from the tweet set *T*. For example, consider the following two tweets for topic *Costco*:

"All the retailers are closed now, except Costco." [Favourited: 0]
"Costco reports 2% profit after stock market closed" [Favourtied: 1]

The words in red are non-stopwords. If we only consider the words in red, a Word Graph would look like this:

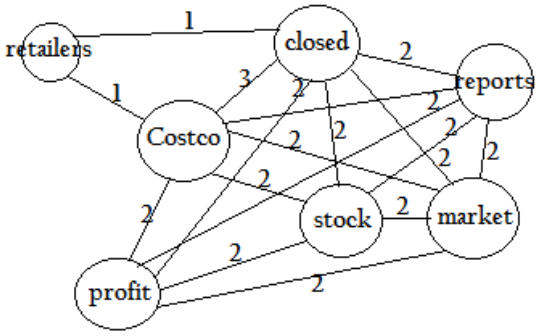

Figure 3: An example of a Word Graph

Once the Word Graph is constructed, we applied a clustering algorithm to it to extract aspects (Table 1). The graph clustering algorithm that we chose was developed by [Blondel et. al., 2008]. In simple terms, the algorithm initializes by placing each vertex in a separate community, and then iteratively moving the vertices around different communities until the highest modularity score is achieved. During the second stage, the algorithm also optimizes the modularity score at the community level to form larger communities.

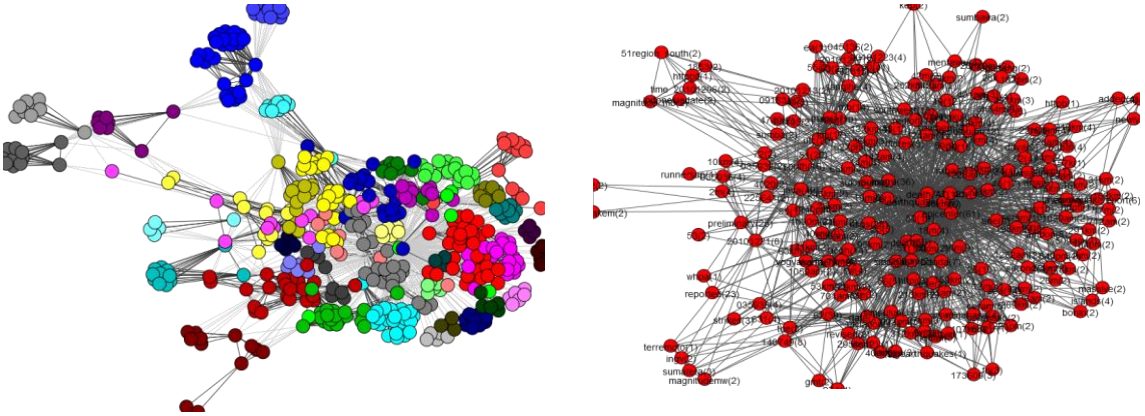

Figure 4: An example of a word graph clustering (left). Each colour represents a cluster (topic aspect) after applying a clustering algorithm. On the right, an example cluster representing an aspect is shown.

Table 1: Example of Aspects for a topic about tsunami

Aspect 1: Total number of words in aspect: 223 Nouns • 196,['earthquake'] • 178,['depth'] • 155,['epicenter'] • 101,['dec'] • 99,['nov'] Adjectives • 13,['southern']	Aspect 2: Total number of words in aspect: 119 Nouns • 88,['volcano'] • 69,['mount'] • 55,['eruption'] • 52,['bromo'] • 47,['indonesias'] • 40,['alert'] • 38,['eruptions', 'merapi', 'news'] • 28,['ash'] • 25,['red', 'issues'] • 20,['toll'] • 19,['death'] Adjectives • 34,['volcanic'] • 21,['beautiful'] • 18,['hot'] • 9,['safer'] • 8,['highest']

The modularity score has been used to measure clustering effectiveness because it measures the density of links inside communities as opposed to between communities [Blondel et. al, 2008]. It is an objective function to maximize, and is stated as:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i c_j) \quad (9)$$

where *A(i,j)* represents the weight of edge between *i* and *j*, k^i or k^j is the sum of edge weights of vertex *i* and *j*, c^i is the community to which vertex *i* is assigned, δ(u,v) = 1 if *u* = *v* and 0 otherwise, and *m* equals to the sum of all edge weights.

During the second stage, when communities are compared to each other for modularity optimization, the weights "edge weights" between different communities equal to the sum of all the edge weights that connect the two communities. This allows possible merging of the two communities to make one larger community should that lead to a higher modularity score. Once aggregation is done at the community level, we re-iterate by going back and checking the communities at the local, vertex level (i.e., first stage). The process continues until modularity cannot be further increased or if the modularity does not improve more than a threshold.

For informational purposes, we calculated the split-join-distance [van Dongen, 2000] and the variance of information (VI) [Meila, 2007] metrics against two other clustering algorithms: Newman's eigenvector method [Newman, 2006] and InfoMap [Rosvall et. al, 2008].

The split-join-distance measures the overlap between two different clusters, and is not commutative. Hence, if a clustering C_a produces a high split-join-distance against C_b, and C_b produces a low distance against C_a, then that means that clusters in C_a have, on

average, little overlap to C_b, but C_b has a higher overlap to C_a. One reason this could be is the higher number of clusters that exist in C_b.

The variance of information (VI) metric shows how much information is lost when clustering is changed to another technique. We can see that Blondel's method is a good compromise between our three clustering choices, as shown in Section 4.6 and Table 2.

Table 2: Summary of graph clustering algorithms explored

Clustering Technique	*Action*	*Output*	*Considers Edge Weights*
Edge Betweenness	Removes the most commonly used edges that connect shortest path between every vertex pair in the graph	Remaining Clusters	No
Bicomponent	Runs a depth-first search to find the biconnected components of the undirected graph	All components of a graph that have a property that at least two vertices must be removed in order to disconnect the graph	No
Weak Component	Runs a breadth-first search to find maximal subgraph in which all pairs of vertices in the subgraph are reachable from one another in an undirected graph, which are called weak components	A set of weak components (subgraphs)	No
Voltage Clustering	Algorithm by [Wu et. al, 2004][36] combined with k-means for determining cluster membership	Maximum number of clusters as per user request	No
InfoMap [Rosvall et. al, 2008]	Random walks to reveal community structure	Arbitrary amount of clusters. Returns an unusually high amount for our use.	Yes
Newman's Eigenvector Method [Newman, 2006]	Uses eigenvectors of matrices to find community structures	Produced similar amount of clusters as Blondel's algorithm, and was competitive in results as well.	Yes
Blondel's Multi-level Clustering [Blondel et. al., 2008]	Optimize modularity at the local level and at the community level.	Arbitrary number of clusters, usually lower than Newman's method.	Yes

3.4. Document Summarization Techniques

As already mentioned in Section 2.3.1 and 2.3.2, there are multiple ways to summarize microblogs. They include: random summarization by selecting random sentences from a corpus; most recent summarization by selecting the most recent microblogs; computing TF-IDF scores for each microblog an selecting arbitrary *k* tweets that have the highest scores *and* are dissimilar to each other based on cosine similarity; and clustering of microblogs to group highly similar tweets and selecting one representative tweet from each cluster for the final summary. Clustering has three main steps: (1) using a clustering algorithm to group the tweets; (2) selecting a representative tweet; and (3) concatenating representative tweets to form a final summary. Since our work uses clustering techniques, we will provide the background for each clustering technique that we investigated. We will also provide the background for two non-clustering techniques: Hybrid TF-IDF and Phrase Reinforcement algorithms.

3.4.1. Agglomerative Clustering

One of the earliest and most widely used clustering techniques is agglomerative clustering [Ackermann et al., 2014]. This method uses a bottom-up approach as follows. Each document will be in its own cluster at first, and will be merged together to the closest cluster (i.e., document) according to a distance measure. If only one cluster is required, then the algorithm repeats itself until all of the documents are clustered into one mega-cluster (i.e., an amalgamation of all documents). In order for the algorithm to work correctly, we need to specify how many clusters we desire so that it stops at a certain cut-off point. Once the algorithm terminates, it returns the set of clusters where each cluster

has the most similar documents. The most representative document is selected from each of the clusters, and combined together to form an overall summary for the topic.

The two most common distance measures to cluster documents are Euclidean (L2) and Manhattan (L1). The Euclidean distance is the shortest line between two document vectors. The Manhattan distance is the sum of absolute value of differences between each element of the document vectors, and it can be thought of as a car travelling in a city block. According to scikit learn[1], the Manhattan measure is better for a sparse matrix. Since we will be making a matrix where each element will correspond to the cosine similarity between two documents, many of the elements will be 0 due to no similarity between the two corresponding documents. Therefore, we shall also use the Manhattan measure for our purpose.

The problem still remains to *which* two documents, one from each of the two clusters, should the distance measure be applied when the two clusters contain more than one document each. There are a number of ways to do this. The complete-linkage strategy proposes finding the distance between two clusters by measuring the distance of the furthest documents. The average-linkage strategy is to group two clusters together for which the average distance between the documents of each cluster is minimized. The single-linkage strategy computes the distance between the closest documents of the two candidate clusters. Finally, the Ward strategy minimizes the sum of squared distances in the Euclidean space *within* all clusters; it sees whether joining two candidate clusters

[1] http://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering

would yield the minimum sum of squared distances between all documents in the merged cluster. We will be using the Ward strategy since it appeared to be balanced like the average-linkage method, but rigorous at the same time. In fact, it is computationally inexpensive to run the Agglomerative clustering using Ward [Murtagh et. al., 2014], and then obtain the representative document of each cluster by using 1-means algorithm. We will adopt the same approach in a couple of our procedures as well; more details are in Section 4.7.

3.4.2. Bisect K-Means++ Clustering

The bisect k-means++ algorithm is a top-down variant of hierarchical clustering techniques, where all documents are merged to form one mega-cluster, and subsequently sub-clusters are formed according to documents' similarities. Although it adopts a top-down approach, whereas Agglomerative clustering adopts a bottom-up approach, when the algorithm terminates at a specified cut-off point (i.e., desired number of clusters are formed), the subsequent steps to form a final summary are exactly the same as Agglomerative clustering. Namely, for each cluster that is returned upon algorithm termination, a representative document is selected. The representative documents from all clusters are merged together to form a final summary of the topic.

The algorithm starts off by merging all the documents together to form a mega-cluster. Next, it selects a random document, and marks it as the first centroid. Then it computes the probabilities of the remaining documents being chosen as a centroid. It does this by marking a document as the next centroid which is furthest away from the first centroid.

Hence, we need to select a document, *v'*, which maximizes the following formula:

$$maximize\ \frac{D(v')^2}{\sum_{v \in V} D(v)^2} \tag{10}$$

where *D(v)* is the Euclidean distance of the vertex to the closest centroid. In plain terms, the further a document is away from the current centroid, the more chance it has to be selected as the second centroid. The process continues until two centroids are chosen. Then we assign the remaining documents to the centroid that most closely align to its features and re-calculate the centroids of both clusters. Then we repeat the entire bisecting process again on the largest formed cluster, and keep repeating until a total of *k* clusters are formed.

3.4.3. Hybrid Term-Frequency Inverse Document Frequency (Hybrid TF-IDF)

A hybrid approach to the classical TF-IDF technique was proposed by [Sharifi et al., 2014]. The traditional TF-IDF technique assigns a weight to each sentence in a document that reflects its saliency within the document. The weight of a sentence is the sum of individual term weights within the sentence. This allows common stop words or rare words to be given less weight and more weight to those words that occur frequently. A term could be any lexical feature, including n-grams [Sharifi et al., 2014]. To determine the weight of a single term, we use the following formula:

$$TF - IDF\ =\ tf_{ij} * \log_2 \frac{N}{df_j} \tag{11}$$

where tf_{ij} is the frequency of term T_j within document D_i, N is the total number of documents, and df_j is the number of documents where term T_j occurs [Sharifi et al., 2014].

One issue with the TF-IDF formula is that it does not work well with microblogs. We do not have a conventional document, but rather text of at most 140 characters. In the construction of a document-term matrix, the many terms may not occur very often, yielding a sparse matrix. To compensate for this problem, the authors developed a hybrid version of the formula, where the term frequencies are calculated across all microblogs but the IDF component treats each document as a separate microblog. Each document weight is divided by a normalization factor, which is the maximum of minimum threshold or the number of words in the sentence. The authors determined the minimum threshold of 11 after many tests. The threshold of 11 ensures that posts containing 11 terms are given priority selection. A sentence longer than 11 terms will be penalized because there is less weight assigned to it, and a sentence that is shorter than 11 terms will also be penalized because 11 is larger than the number of terms that exist in the post. The algorithm is summarized below:

$$W(S) = \frac{\sum_{i=0}^{\#WordsinSentence} W(w_i)}{nf(S)} \tag{12}$$

$$W(w_i) = tf(w) * \log_2(idf(w_i)) \tag{13}$$

$$tf(w_i) = \frac{\#OccurrencesOfWordInAllPosts}{\#WordsInAllPosts} \tag{14}$$

$$idf(w_i) = \frac{\#SentencesInAllPosts}{\#SentencesContainingWord} \quad (15)$$

$$nf(S) = max[MinimumThreshold, \#WordsInSentence] \quad (16)$$

where W is the weight assigned to a sentence or a word, nf is the normalization factor, w_i is the i^{th} word and S is the sentence (i.e., microblog).

3.4.4. Phrase Reinforcement (PR) Algorithm

A summary in exactly one sentence or phrase for a given a set of tweets was proposed by [Sharifi et al., 2014]. The authors proposed an algorithm that takes advantage of position of words around the topic phrase (Figure 5). The idea is that users will mention keywords about the topic closer to the position of the topic phrase within a sentence.

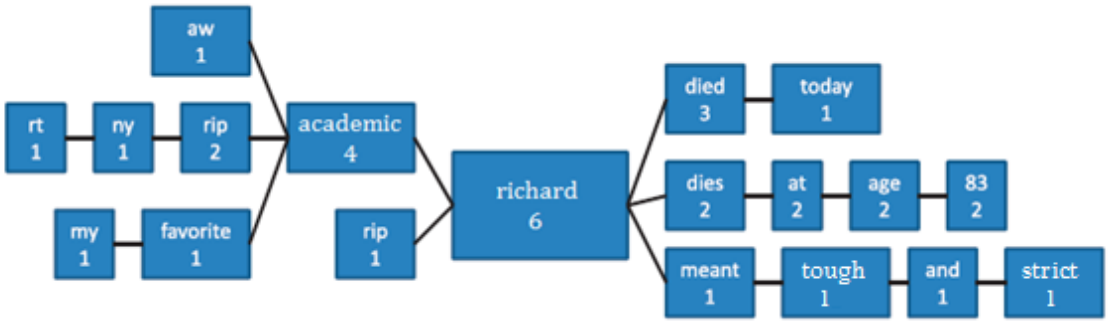

Figure 5: Fully constructed PR algorithm with unique word positions

The algorithm develops a directed graph where each vertex represents a word and unique position after or before from the topic phrase. In other words, there could be repetitive words in the graph because there could be sentences that mention those words at different

positions in the sentence relative to the topic phrase. Each vertex is given a weight as follows:

$$\text{weight}(node) = \text{count}(node) - \text{distance}(node) * \log_b \text{count}(\text{node}) \quad (17)$$

where the count of a node equals the number of occurrences of the respective word in the tweets and the distance of the node is the number of positions a word is away from the topic phrase. The log*b* is a penalty given to the weight, so the further a word is from the topic phrase the higher the penalty it will get. The final summary is the one that gives the highest total weight of a directed path that will obviously include the topic phrase. This is achieved by first finding the most weighted phrase going *forward* from the topic phrase, and most weighted phrase going *backwards* from the topic phrase and combining the two together.

There are some inherent limitations with this algorithm. Although the directed and position specific nature ensures that the graph is acyclic, the final summary sometimes may not be semantically sound. For example, consider the following two sentences for topic *Costco:*

"All the retailers are closed now, except Costco."
"costco, best buy, target all on sale"

If the two sentences were weighted according to the PR algorithm, then the final summary would be: *"All the retailers are closed now, except Costco best buy, target all on sale"*.

Although the PR algorithm provides a one document summary, we can use it if we expect that aspect extraction from a topic will yield highly similar documents grouped together. Hence, we tested the PR method in our evaluation to extract one sentence summary for each aspect.

3.5. **Summary**

In this chapter, we provided the background of our work related to microblog summarization. In particular, we looked at how we will use sentiment analysis, Word Graphs, and document summarization techniques to summarize topics discussed on Twitter. For sentiment analysis, we will use an ensemble of Naïve Bayes and RNNs. For Word Graph clustering, we will use Multi-Level clustering by Blondel. Finally, for document summarization techniques, we will look at multiple methods and compare them to determine which one is best: Agglomerative clustering, Bisect K-Means++ clustering, Phrase Reinforcement, and Hybrid TF-IDF.

Chapter 4. Experimental Setup and Evaluation

4.1. Data collection

We used a dataset of 2.8M tweets between November and December, 2010 from our research lab. The dataset did not consist of all tweets during these two months. We did not know which topics trended during this time period. Hence, we utilized TagMe to annotate the tweets with Wikipedia concepts where each concept had a relatedness score. Topics consisted of a single concept as well as multiple concepts which meant that the topic was very focused. We used the topics generated by [Fani et al., 2015] since they used the same dataset.

4.2. Sentiment Analysis

Our first goal in this research is to determine if first grouping the tweets by their sentiments will improve the overall summarization of the topics. We used the Datumbox Framework and Stanford CoreNLP Sentiment package [Manning et. al, 2014] to develop an ensemble. As mentioned earlier, each have their drawbacks which was proven by our empirical tests as well. Since the Datumbox Framework uses multinomial Naive Bayes, preprocessed tweets on it were not necessary. However, the Stanford CoreNLP toolset was sometimes providing correct sentiments on fully pre-processed tweets and, on other occasions, correct sentiment on original tweet text. Therefore, our ensemble took an

average of three sentiments: original text on Stanford, preprocessed text on Stanford CoreNLP, and original text on Datumbox. The sentiments provided by using both tools were used to come up with a final sentiment of positive, negative, or neutral.

We used the default trained model provided by the Stanford CoreNLP package to use its sentiment analysis feature. However, for Datumbox Framework, not only did we use the training data that came with the package, but we also concatenated the user reviews from the SFU Review Corpus[2] into the training files. The reviews were marked positive or negative based on "recommended" and "not recommended" tag, respectively, by the reviewer. The corpus included reviews from books, cars, computers, cookware, hotels, and phones.

After the aspects have been made, we also use SentiWordNet to obtain crude sentiment *temperatures*. Recall that each aspect is extracted using clustering techniques on Word Graphs, so they are a collection of words. Any known nouns, adjectives, adverbs, and verbs were used to get the positive, negative, and objective (neutral) temperatures from SentiWordNet [Esuli et. al, 2006]. These temperatures were averaged together to obtain the overall temperature of the aspect. The calculation is as follows:

[2] https://www.sfu.ca/~mtaboada/research/SFU_Review_Corpus.html

$$sentiment\_temperature_p = \max\left(\sum_{w \in p}(positive\_score_w * weight_w),\right.$$
$$\sum_{w \in p}(negative\_score_w * weight_w),$$
$$\left.\sum_{w \in p}(neutral\_score_w * weight_w)\right) \quad (18)$$

where,

$$weight_w = degree_w * .30 + (favourited + 1)_w * .70 \quad (19)$$

and $score_w$ is a sentiment percentage for word *w according to SentiWordNet [Esuli et al., 2006]*, and $degree_w$ is the degree for word *w*. The degree of *w* is given a 30% weight because some of the edges of *w* may be connected to words of another aspect. The second component of $weight_w$ is given a weight of 70% because the number of times all tweets are favourited where *w* appears show that *w* is likely an important word overall. As an example, consider an aspect with just three words. Each of these three words have a positivity, negativity, and neutrality percentage. We can calculate the overall sentiment temperature of aspect *p* by finding out the dominant sentiment percentage as a weighted average of word occurrences and tweets favourited that contained one of the words.

This is a naive method to obtain sentiment temperatures, we recognized that our temperature result will have its limitations. To circumvent this limitation, if a graph was uniquely identified as one of the three sentiments based on the ensemble of Stanford CoreNLP and Datumbox (i.e., since all the tweets in the graph were all originally in the

same sentiment class), then only the temperatures of that same sentiment will be considered for every aspect of that graph. Hence, for aspect *p*, (19) will change to:

$$sentiment\_temperature_p = \sum_{w \in p} (ensemble\_class\_score * weight_w) \qquad (20)$$

where *ensemble_class_score* is the single sentiment class (i.e., positive, negative, or neutral) of all tweets from the ensemble of our two sentiment classifiers. The aspects will be ordered by sentiment temperatures from highest to lowest, and the top four aspects will be selected for evaluation. We asked the evaluators: *"Based on your manual summary, from scale of 0-5, please rate how accurate you feel are the (sentiment) percentages for each rank (out of 100)",* where the word *sentiment* was replaced with "positivity", "negativity", and "objectivity/neutrality". This question was based off the *Content* metric from DUC 2002, which asks a human judge to measure how complete an automated summary expresses the meaning of a human summary.

4.3. **Assigning Tweets to Topics**

A topic is one that has at least one TagMe concept, which is based on Wikipedia. Hence, “Apple” could be a topic but has multiple meanings associated to it. A topic that is made of two concepts: “Apple” and “iPhone” will require that a tweet has annotations for both of these concepts. Conversely, TagMe will also be able to annotate those tweets by looking at overlap between the tweet and Wikipedia. We chose those topics for evaluation that had at least 2-3 concepts.

Next, we needed to determine how many tweets corresponded to each topic. This was done by developing a scoring function, which looked at the annotation of each tweet given by TagMe, the relatedness score of each annotation, and the number of concepts in the topic itself. This was a modification of a scoring function of Fani et al. Our scoring function is based on empirical tests that would provide sufficient amount of highly related tweets, thus we tried different parameters and conditions until we felt that our results are good for analysis. A tweet is assigned:

1. If the topic size is one concept and the tweet also contains an entity for the same concept and the relatedness score of the single concept meets a threshold,

2. If the topic size is two concepts, and the tweet also contains entities relating to each of the two concepts, and the total relatedness score of the both concept meets a threshold,

3. If the topic size is three or more concepts, and the tweet contains at least two of the three concepts, and number of concepts in the tweet is at least 3 times the size of the topic size, and the average relatedness score of each concept in the tweet is at least a certain threshold.

After trying different values and using human judgement, we observed that a total relatedness score of 0.30 for Steps 1 and 2 and threshold of 0.07 for Step 3 was good for our purposes.

4.4. Preprocessing the Tweets

An issue with microblogs is their informal structure and syntax. They contain a lot of noise that could negatively affect our process, especially sentiment analysis. Moreover, it was also a good idea to preprocess the tweets and save them in a database for quick extraction later on. All tweets were preprocessed as follows which closely followed the steps of [Sharifi et al., 2014]:

1. Convert any HTML-4 and HTML-3 encoded characters into ASCII.
2. Remove any Unicode characters (e.g., '\x000').
3. Remove any embedded URLs (e.g., http://), HTML tags (e.g., <body>), other tags (e.g., <>), tokenize any smileys, remove any accents, and user mentions (e.g., @Muhammad).
4. Discard the document if it is not English. We used Language Detection tool by Shuyo[3].
5. Remove duplicate posts by same user.
6. Remove any terms that are equal or larger than 20 characters. This is to ensure that any long hash-tags or other obscure terms are removed.
7. Remove any consecutive question marks that are 6 characters or longer (e.g., ??????). This is to ensure that any tweet is removed that was detected as English incorrectly, because certain non-English tweets may have lost their original Unicode formatting already and were replaced with question marks instead.

[3] https://shuyo.wordpress.com/

8. Remove the stop-words.

9. *For Phrase Reinforcement algorithm:* Break the documents into sentences. Most tweets have only one sentence.

10. *For Phrase Reinforcement algorithm:* Detect the longest sentence that contains the topic phrase and use it to represent the tweet.

11. For Word Graph construction only: Remove any punctuation marks.

Tweet preprocessing was completed using the Datumbox Framework.[4].

4.5. **Word Graph Construction**

In order to construct our Word Graphs, we used the Jung (Java Universal Network/Graph Framework) API[5] and extended it for our own purposes. The graphs were ported to Pajek format and used in Python's igraph[6] library for clustering. Each Word Graph was uniquely identified by the topic and the overall sentiment of positive, negative, or neutral. Thus, each topic had a Word Graph for each sentiment.

Furthermore, we also investigated if tweets had any replies associated to it. On Twitter, users can reply to tweets, favourite them, or re-tweet them. In many instances, replies may not have the same annotation as the original tweet and, hence, will not be assigned to the same topic as the original tweet. In order to ensure that replies are also assigned to the

[4] http://www.datumbox.com/machine-learning-framework/
[5] http://jung.sourceforge.net/doc/api/
[6] http://igraph.org/

same topic, we looked at the *replyToId* attribute of the tweet and linked the original tweet to it. So if the original tweet is assigned to a topic, automatically all its replies would be assigned to the topic as well for the same sentiment. Each edge weight between the original tweet and the reply would equal to the amount of times the reply is favoured plus 1, consistent with the methodology of weights provided to co-occurrence of words within a tweet itself.

It is possible that certain replies may contain an opposite sentiment, however, we will take that into account by changing the sentiment *temperature* of the original tweet using any replies associated to it. All in all, only 1.4% of all tweets in the dataset were affected by this.

4.6. Clustering Techniques for Word Graphs for Aspects Extraction

Our second goal in this research is to determine if having Word Graphs to induce aspects of the topic, before summarization, will improve the overall summaries. We considered many clustering techniques to cluster our Word Graphs, as shown in Table 2 in Section 3.3.

InfoMap was the selected method for clustering Word Graphs. As described earlier in our work, this method consistently returned the lowest amount of clusters which was reasonable for us to work with. In order to justify the use of InfoMap over other graph clustering techniques, we found variance of information (VI) and split-join measures to compare InfoMap to other clustering techniques. Furthermore, the VI metric as shown in

Table 3 measures the amount of information loss when changing from one type of clustering to another. Smaller values for Blondel against Newman and InfoMap indicate that Blondel has good clustering because less information is lost . The split-join-distance shown in Table 4 was high for Blondel against Newman (221) and InfoMap (377), which meant that the amount of overlap was low if we change our clustering from Blondel to one of the other two. Newman also showed a comparable high score, meaning smaller overlap. However, since the VI metric was more stable for Blondel, it was proper to choose it to cluster our Word Graphs.

Table 3: Variance of information between three clustering algorithms

	Variance of Information (VI)
Blondel-Newman	1.72
Blondel-InfoMap	1.43
Newman-InfoMap	2.09

Table 4: Split-join-distance between three clustering algorithms

From/to	Blondel	Newman	InfoMap
Blondel	-	221	377
Newman	261	-	423
InfoMap	42	112	-

4.7. Clustering Techniques for Documents (Microblogs)

As described earlier in our work, we used Agglomerative, Bisect K-Means++, Hybrid TF-IDF, and Phrase Reinforcement for clustering documents. There are eleven different ways that we used these techniques.

The first two methods involved directly applying Agglomerative and Bisect K-Means++ to documents without making any Word Graphs, in order to draw comparisons of the effectiveness of the Word Graphs. The Agglomerative implementation was in scikit learn, whereas Bisect K-Means++ was implemented with the help of Pyclust package of Python as well as scikit learn. Each of these techniques returned four clusters, so that we can use [Sharifi et al., 2014] as the baseline as shown in Figure 6. These four clusters were passed to a 1-means algorithm to determine the centroid which was used as the representative tweet for that cluster. Therefore, each graph had four tweet summaries.

Instead of using 1-means on each of the four clusters of Agglomerative and Bisect K-Means++, the third and fourth methods involved the use of Hybrid TF-IDF to select the top *k* documents that represent each of the clusters. Sharifi et al. determined that Bisect K-Means++ combined with Hybrid TF-IDF was the best methodology for multiple document summarization using clustering algorithms. Hence, we did the same and picked top weighted tweet from each of the four clusters using Hybrid TF-IDF and combined them together to produce a four sentence summary for each graph. In order to avoid picking posts that are highly similar to each other, the cosine similarity measure was used to determine the similarity between each post. The cosine similarity is defined as follows:

$$sim(v_i, v_j) = cos(v_i, v_j) = \frac{v_i^t \, v_j}{\|v_i\| \|v_j\|} \quad (21)$$

where v_i and v_j are sentence *i* and *j* that need to be compared. The similarity threshold of 0.77 was used. In other words, if sentences *i* and *j* have similarity of 0.77 or greater, then only *i* will be selected in the final four sentence summary. We keep reiterating over all tweets until at most four are selected for presentation. We chose 0.77 as the threshold because this is what Sharifi et al. determined was optimal. Similarly, the fifth method involved using just Hybrid TF-IDF as that was the best overall summarizer with respect to F-Measure according to Sharifi et al.

The next five methods are exactly as the ones described above, with the difference that they are applied on tweets of *each aspect of the Word Graphs.* The eleventh method is to use PR algorithm to get a 1 sentence/phrase summary of each aspect, without any clustering of that aspect. We chose base 100 since that had a stable ROUGE-1 performance in paper of Sharifi et al. In order to determine which topic phrase to set as root node, we used the TagMe concept titles in the topic and broke them down to unigrams, and considered each unigram as a topic phrase. We did not consider stop words as possible topic phrases. The PR algorithm looked for the most weighted phrases from the root towards right and left separately, and then combined the two phrases from both directions together that had the same root node.

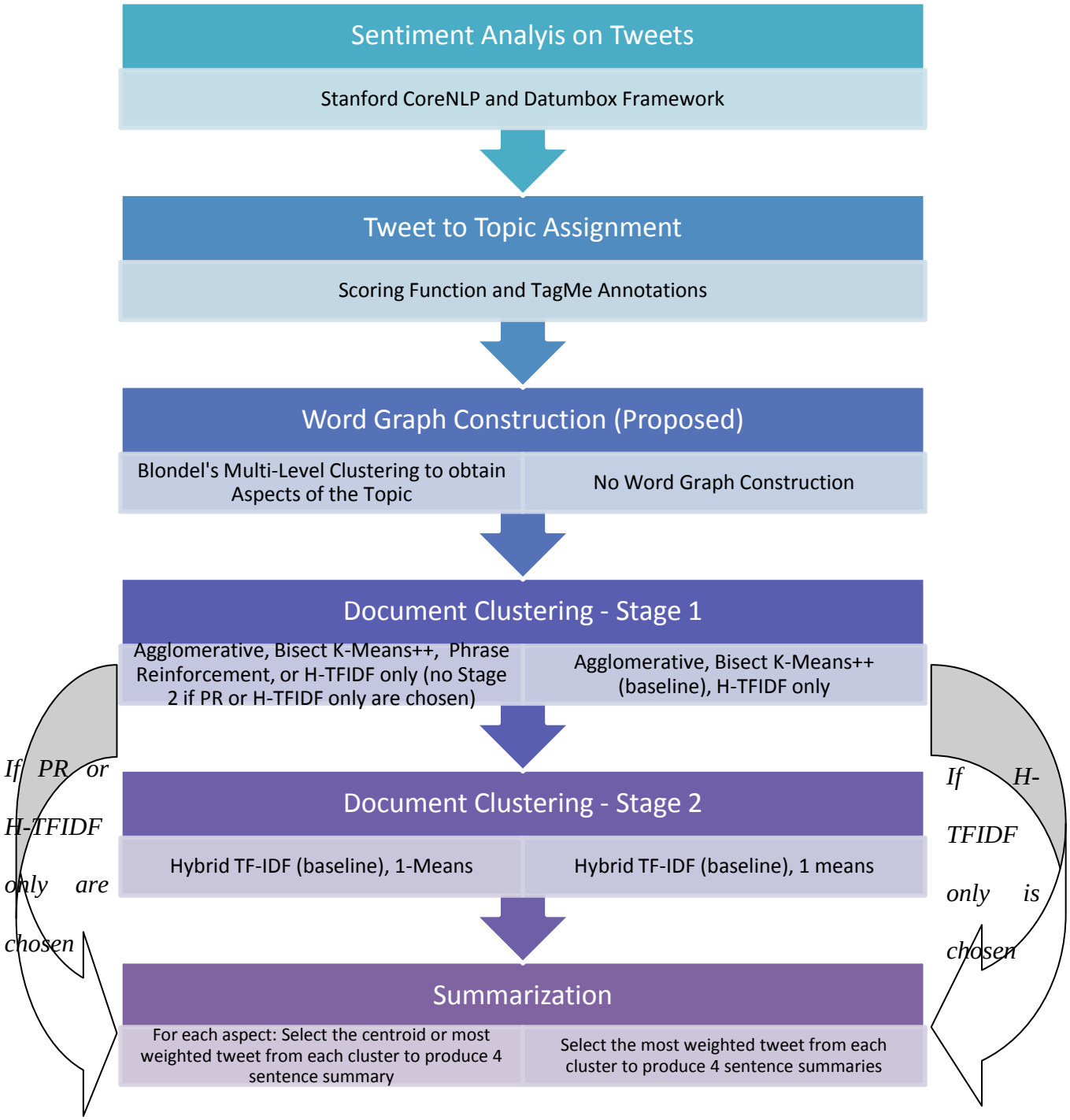

Figure 6: Shows the various steps to come to the final four sentence summaries for each aspect (left columns) or for the whole topic (right columns)

4.8. **Conclusion**

In this chapter, we looked at the evaluation setup of our research. In the next chapter, we will present the results of our evaluation and discuss how the creation of Word Graphs helps in overall summaries of the topics.

Chapter 5. Evaluation Results

5.1. Introduction

Our goal is to determine whether having Word Graphs to induce aspects improves the overall summarization process and if sentiment temperatures rank aspects correctly as most positive, most negative, or most neutral.

Six topics were evaluated by four individuals. Each topic had at least 80 positive tweets and 80 negative tweets. Two of the topics were analyzed twice by two individuals; two topics were evaluated two times each. Hence, in total we had eight analysis runs. All volunteers had to perform summarization tasks prior to Word Graph construction and post-Word Graph construction for the topics that they were assigned, respectively. As mentioned in Figure 6 of Section 4.7, there are two types of workflows for evaluation. In the first workflow (right column of Figure 6), we directly cluster the documents in four groups and find a representative tweet from each cluster to come up with a four sentence summary. In the second workflow (left column of Figure 6), we first create Word Graphs to induce aspects of the topic and then cluster each aspect into four groups, and pick a representative tweet from each cluster to come up with an *aspect* summary.

For evaluating summaries prior to Word Graph construction (first workflow), each volunteer was given three sets of tweets for their assigned topic: positive, negative, and

neutral. For each set of tweets, they were required to group the tweets into four clusters and then pick a representative tweet from each cluster to obtain a four sentence summary for that sentiment. Afterwards, the volunteers were asked to provide Content scores (see Section 5.2 for details) for different algorithms by comparing their summary to the summary of the algorithm.

For evaluating summaries after Word Graph construction (second workflow), volunteers were given three sets of tweets for each topic: positive, negative, and neutral. Each of these sets contained four more sets of tweets. These subsets corresponded to only the top four ranked aspects by sentiment temperature as determined by SentiWordNet and aspect information. For each aspect, they were required to group the tweets into four clusters and then pick a representative tweet from each cluster to obtain a four sentence summary for that aspect. Hence, each aspect (or sentiment rank) had four clusters in the end amounting to a four-sentence summary. For example, when SentiWordNet identified the top four positive *aspects* of the topic by positive temperatures, then those four aspects were clustered into four groups each before summarizing each aspect by picking a representative tweet from each cluster. We maintained the choice of 4 clusters at every step in order to compare to our baseline. Afterwards, the volunteers were asked to provide Content scores for different algorithms by comparing their summary to the summary of the algorithms.

Our baseline was Sharifi's Bisect K-Means++ with Hybrid TF-IDF and Hybrid TF-IDF by itself, obviously without any Word Graphs. We also used Sharifi's Phrase Reinforcement algorithm for comparison.

5.2. **Content Scores**

We wanted to measure how the volunteers feel that their manual summaries express the meaning of the automatic summaries, from a scale of 0 (no similarity in meaning) to 5 (same in meaning). We obtained the average content scores over all clustering techniques. Lower scores are given by volunteers because they did not like preprocessed versions of tweets with stop words removed, or they felt sentiments were incorrect, or that they were tough markers.

Table 5: Content Scores by Sentiment

Sentiment	Average Score	Standard Deviation
Positive	2.9	0.60
Negative	2.3	0.86
Neutral	3.0	1.12

If we analyze the content score for the different algorithms in Table 6, we can see that the agglomerative technique produced the best results. The PR algorithm was the worst performer because the topic phrases, which were an extraction of the concept title, were not always there. Since our tweets were based on TagMe annotation, that does not always mean that a word from the concept title would also exist in the tweet. This was an expected result. We also noticed that overall, content scores increased when volunteers were analyzing summaries of aspects, which was also expected because they were looking at a more focused subset of tweets (Figure 7).

Table 6: Content Scores by Algorithm before and after Word Graph construction

	Score (before WG)		*Score (after WG)*			
Technique	*Score*	*Std. Dev*	*Score*	*Std. Dev*	*Our Proposed Work*	*Comments*
Agglomerative	1.87	1.56	3.15 (best)	0.39	Yes	Improvement from baseline by 0.23
Agglomerative + Hybrid TF-IDF	1.60	1.38	2.91	0.32	Yes (partial)	Hybrid TF-IDF was from Sharifi et al.
Bisect K-Means++	1.85	1.39	2.75	0.62	Yes	Sharifi et al. did not use Bisect K-Means++ with 1-means pass
Bisect K-Means++ / Hybrid TF-IDF	1.62	1.20	2.92	0.39		Sharifi et al.
Phrase Reinforcement	N/A	1.12	1.08 (worst)	0.26		Sharifi et al. Before WG, PR was not used as it only produces a one-sentence summary for the entire corpus of tweets. After WG, PR was applied to each cluster
Hybrid TF-IDF only	1.51		2.72	0.51		Sharifi et al.

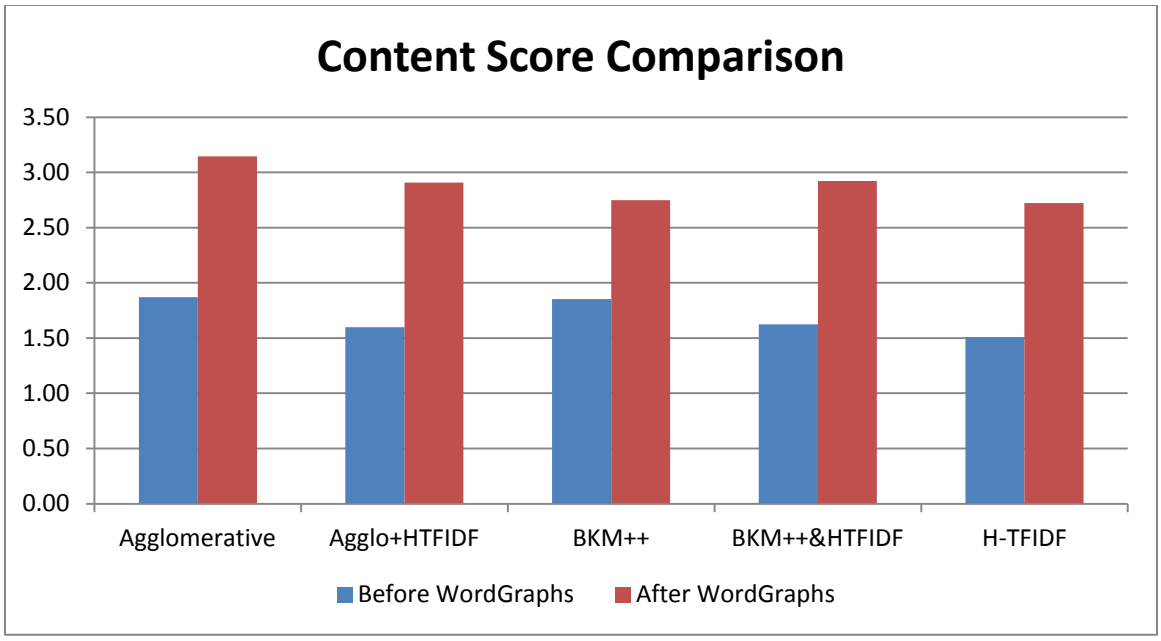

Figure 7: Content score comparison after making Word Graphs

5.3. Evaluation Methods

There is no clearly defined standard for evaluating automatic summaries. However, as suggested in [Lin et. al, 2003] we can perform intrinsic evaluation by comparing the summary to a manual summary (i.e., gold standard). One popular automatic evaluation metric that has been adopted by the Document Understanding Conference (DUC) is ROUGE. ROUGE is a suite of metrics that automatically measure the similarity between an automated summary and a set of manual summaries [Lin et. al, 2003]. The calculation of ROUGE-N, based on n-gram, is as follows:

$$\text{ROUGE} - \text{n} = \frac{\sum_{s \in MS} \sum_{n-gram \in S} \text{match}(n-gram)}{\sum_{s \in MS} \sum_{n-gram \in S} \text{count}(n-gram)} \quad (22)$$

where *MS* is the set of manual summaries, *n* is the length of n-grams, and match(*n*-grams) is the number of co-occurrences that an *n*-gram was found in both the manual summary

and automated summary. Since [Sharifi et al., 2014] used ROUGE-1 metric, we will adopt the same for better comparison to their results.

The ROUGE-1 metric can be modified to obtain precision of auto summaries as follows:

$$p = ROUGE - 1' = \frac{\sum_{m \in MS} \sum_{u \in m} \text{match}(u)}{|MS| * \sum_{u \in a} \text{count}(u)} \left(= \frac{matched}{retrieved} \right) \quad (23)$$

where $|MS|$ is the number of manual summaries and a is the automatic summary. Finally, the F-measure is computed as:

$$F - \text{measure} = \frac{2pr}{p + r} \quad (24)$$

where, in our case, r is the ROUGE-n calculation for 1-gram. Various complex methods to evaluate automatic summaries have been considered [Saggion et. al, 2010] and [Louis et. al., 2009] , but no one model seemed to be conclusively working according to Sharifi et al. However, ROUGE is still the most widely used summarization evaluation framework.

In order to test for semantic coherence, we also used a manual metric used during DUC 2002: the *Content* metric which asks a human judge to measure how complete an automated summary expresses the meaning of a human summary. We asked volunteers to provide their rating between 0 (no similarity in meaning) and 5 (same in meaning).

5.4. **ROUGE-1 Scores**

In order to calculate the ROUGE-1 scores, we used ROUGE 2.0 [Ganesan, 2015] for Java. ROUGE uses unigrams in each of the manual summaries and compares it to the automatic summaries. It also depends on the size of the manual summaries, which we controlled by asking for four sentence summaries from our volunteers.

Figure 8 shows the results of ROUGE based on clustering techniques prior to Word Graphs. We can immediately see that Sharifi et al.'s baseline of Bisect K-Means++ with Hybrid TF-IDF was outperformed by Agglomerative clustering with 1-means pass. The standalone Bisect K-Means++ method refers to picking of the centroids from each cluster with a 1-means pass, which was not attempted by Sharifi et al.'s, and performed competitively as well. Agglomerative clustering technique with 1-means pass was also something that was not seen in previous papers for microblog clustering, and it seems that it outshines all the other algorithms.

One of the goals of the research was to see whether making Word Graphs prior to document clustering helps in the clustering process. Figure 9 shows the performance of the methods, with the addition of Phrase Reinforcement. We wanted to bring in the PR method at this stage to see the effectiveness of choosing the highest ranked phrases from both directions of the topic phrase. In many instances, our tweets did not contain the topic phrases that existed in the concept titles of the topic. Hence, the performance of the PR algorithm was low.

We can see that all algorithms performed better after Word Graph construction. The best overall summarizer was the Bisect K-Means++ with Hybrid TF-IDF. The Hybrid TF-IDF only algorithm, whether with or without Word Graphs, did not perform as well as we thought. We believe that this is because that the Hybrid TF-IDF algorithm picks those tweets that have the most salient features in the set. Our manual summaries consisted of those tweets that the volunteers felt were more representative of the group, because they were re-tweeted or the content was often repeated, which could potentially cause ROUGE-1 scores of Hybrid TF-IDF algorithm to decrease.

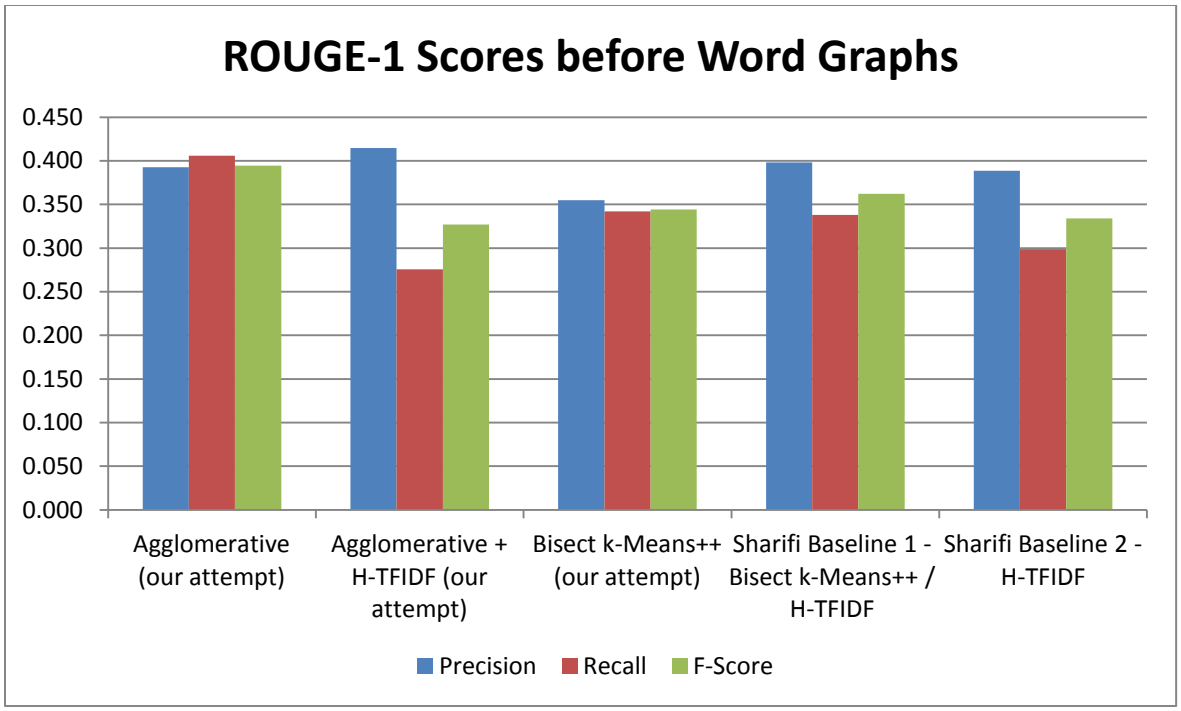

Figure 8: Performance of document clustering techniques prior to Word Graphs.

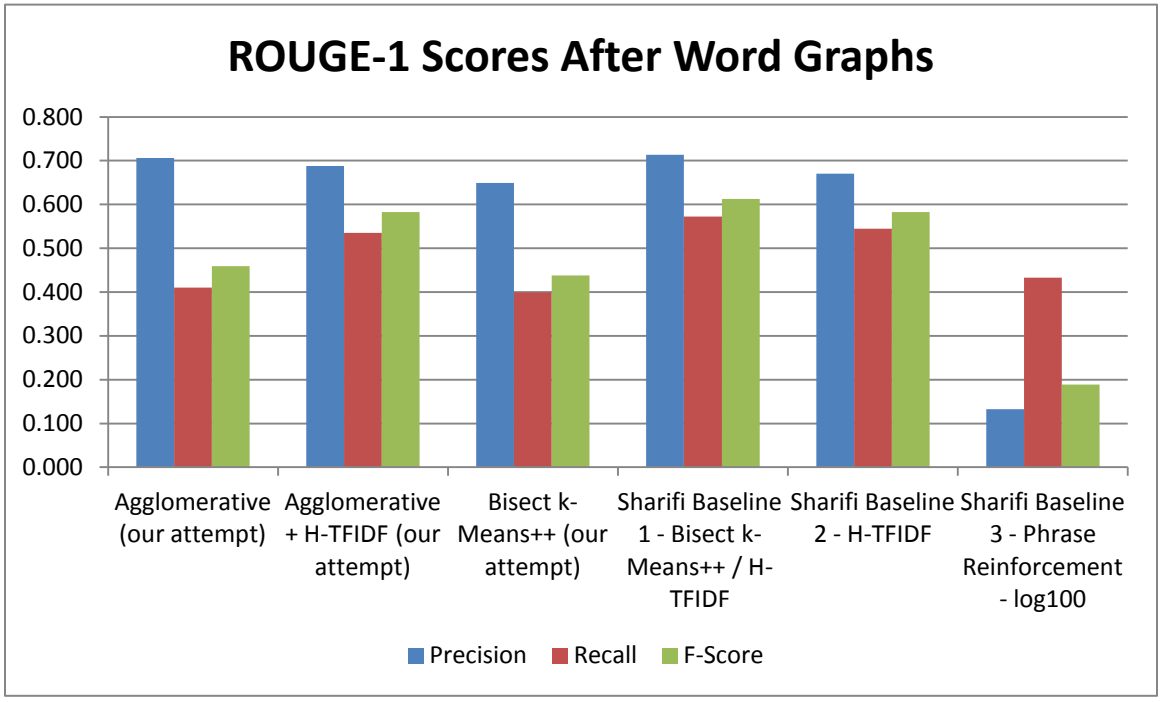

Figure 9: Performance of document clustering techniques after Word Graphs.

We also wanted to see which method benefits the most by Word Graphs. Figure 1 shows the results of the performance deltas. We can see that Agglomerative with Hybrid TF-IDF (F-Measure delta: 0.255+) benefits the most, followed by Bisect K-Means++ with Hybrid TF-IDF (F-Measure delta: 0.251+), and then Hybrid TF-IDF only (F-Measure delta: 0.249+) We also noted that all techniques had positive deltas, which was expected because the algorithms were given a more focused set of tweets to provide summaries for.

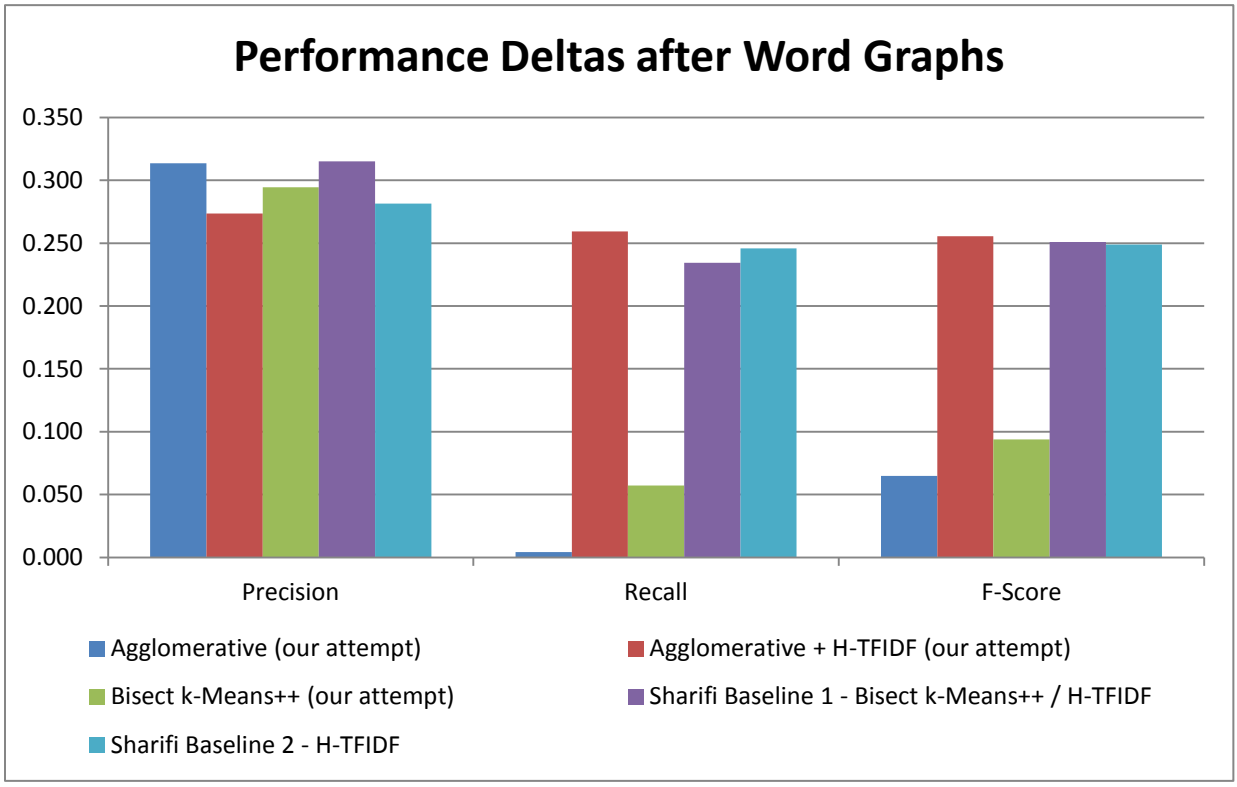

Figure 10: Delta performance of document clustering techniques

5.5. Aspect Ranking by Sentiment Temperatures

When we asked our volunteers to cluster different aspects, we had provided them only four aspects for each topic that were returned by Agglomerative or Bisect K-Means++. These four aspects were the top rated aspects by sentiment temperature. The temperature was calculated using SentiWordNet as already described in Section 4.2. We asked our volunteers how accurate they felt were the temperatures for each of the top four rankings of an aspect based on their own manual summary. They provided the rankings between 0 (not similar in meaning) and 5 (same in meaning). Table 7 shows that the average score was satisfactory, except for positive aspects. Since we calculated the temperatures in a naive way, we had no strong expectations on the validity of the rankings. In future, a non-naive method could improve these scores.

Table 7: Average score of volunteers on the accuracy of aspect temperatures

Positive	Negative	Neutral
2.6	3.0	3.3

5.6. Summary

In this section, we observed that the Agglomerative clustering technique with 1-means pass was better summarizer without construction of Word Graphs. With the construction of Word Graphs, all summarization algorithms that we experimented had better F-Measures.

We also noted that users were overall relatively content when comparing their manual summaries to the automated summaries. The Agglomerative clustering technique with 1-means pass received the highest content score.

Finally, computing sentiment temperatures with a naive method has average results. We feel this could be improved with a non-naive method.

In the next chapter, we will provide concluding remarks to our thesis.

Chapter 6. Summary and Conclusions

In this thesis, we wanted to use various techniques and tools relating to natural language processing and the Semantic Web to determine if there is a better methodology to summarize topics in microblogs. We performed sentiment analysis on tweets to determine if they were positive, negative, or neutral; used named-entity recognition and annotation to match tweets with topics; generated aspects of topics; and clustered the aspects to come up with final summaries for the aspect and the topic.

In the first part of the thesis, we provided related work in the area of microblogs and how unstructured content is being handled for various tasks such as inferring user interests, extracting semantics from tweets, and summarization. Next, we presented our methodology and theory behind some of the algorithms being used. In Chapter 4 and 5, we discussed the evaluation setup and results.

We found that the Agglomerative clustering technique with 1-means pass, which to the best of our knowledge has not been used for microblog summarization, performed better than all other algorithms before Word Graphs were constructed, including Sharifi's baselines. After Word Graph construction, Sharifi's Bisect K-Means++ with Hybrid TF-IDF was a better performer. However, our objective was to measure if constructing Word Graphs improves overall summarization process. The F-Measures obtained after constructing Word Graphs improved scores for all algorithms, but Agglomerative

clustering with Hybrid TF-IDF had the largest improvement with the construction of Word Graphs, with a F-Measure delta of +0.255.

Some of our findings in this thesis are:

- Word Graph construction to extract aspects improves overall F-Scores regardless of the document summarization algorithm used afterwards. The topic is summarized by summarizing only the most important aspects, instead of summarizing the entire topic.

- Agglomerative clustering has better Content scores prior and post- Word Graph construction.

- Agglomerative clustering with 1-means pass is better than the state of the art, albeit marginally, for microblog summarization without Word Graph construction.

- Sentiment analysis helps improve the overall summarization tasks by grouping tweets by sentiments first.

- Word Graphs can be used to generate aspects of the topic in order to understand what are the most positive and negative aspects of the topic. For some topics, this can be used to understand controversy in a topic.

- Users like to see non-processed tweets in their summary.

Future work could entail investigating how to better rank the aspects of a topic. Currently, we ranked aspects by sentiment temperature however there could be other ways to rank as well. Another ranking method could involve ranking by the importance of the aspects, which could be a function of the number of tweets for that aspect and how recent the tweets are in that aspect. Our volunteers also expressed that sentiment classification of tweets need to be improved. This could be mitigated by creating an

ensemble classifier that is well-trained on a variety of different topics in social media. Obviously, this requires good amount of training data which we already know is a problem (see Section 3.1) as most sentiment training banks are highly focused on political issues.

Another opportunity for research is the timing of sentiment analysis. In our thesis, we adopted a sentiment to aspect approach: (*) we classified the tweets into positive, negative, or neutral; assigned these tweets to topics; obtained aspects for each sentiment class; ranked the aspects according to sentiment temperature which was based off the sentiment class. In future, we could explore the results of Content scores by adopting an aspect to sentiment approach, by first removing the first step (marked by *), clustering the entire corpus of tweets for the topic, and completely relying on SentiWordNet to classify each aspect as positive, negative, or neutral according to the most dominant temperatures found in the aspect.

Appendices

A. Topic Selection

Topic 1 - Foot (unit) - Vehicle - Accident - Anchorage, Alaska - Snow Positive tweets 80 Negative tweets 80 Neutral tweets 80 Internal cohesion of this topic is: 1.0	Topic 2 - Upgrade U - IPhone 3G - Apple Inc. - Lawsuit Positive tweets 80 Negative tweets 80 Neutral tweets 80 Internal cohesion of this topic is: 1.0
Topic 3* - Privacy - Facebook Positive tweets 172 Negative tweets 162 Neutral tweets 387 Internal cohesion of this topic is: 1.0	Topic 4 - China - Inflation Positive tweets 80 Negative tweets 80 Neutral tweets 80 Internal cohesion of this topic is: 1.0
Topic 5* - HIV/AIDS - Malaria - World Pneumonia Day Positive tweets 89 Negative tweets 237 Neutral tweets 266 Internal cohesion of this topic is: 1.0	Topic 6* - Bloomberg Businessweek - Economy of the United States - Retail Positive tweets 95 Negative tweets 247 Neutral tweets 262 Internal cohesion of this topic is: 2.0

Topics marked with (*) were evaluated with stopwords removed.

Each topic is made of at least two to a maximum of five TagMe concepts.

B. Clustering Examples

Topic and Sentiment	Agglomerative	Bisect K-Means++ w/ H-TFIDF	Hybrid TF-IDF	Manual
1 (Negative) *Unprocessed Tweets*	RT @wxchannel: NASA Modis satellite imagery showing snow cover and the storm over the Northeast Monday: http://bit.ly/hfBdLa #eastsnow rt sharp gradient in snow on western edge contrast forecast snowfall from washington dc to boston rt colder air is filtering inblue canyon is 29 degreessnow at tahoe should all be snow above 5000 feet now rt it now appears heavy snow is not likely in indiana christmas eve tilt a few snow showers are possible north east	Brrrr...Today in 1947, over 26 inches of snow fell on New York City; it was the city's heaviest snowfall on record. RT @13News: #Snow for the record books: NORFOLK -- Sunday's snowfall was the 3rd heaviest on record for Norfolk.... http://bit.ly/fevHrz At least 46 states had snow this #Christmas. More than 50% of Lower 48 had #snow cover Christmas morning: http://ow.ly/3uWAU 9:00am Snow Update: LATEST: Slow and steady would describe the snowfall here in the viewing area. Here is the l... http://bit.ly/fhLvqv	Brrrr...Today in 1947, over 26 inches of snow fell on New York City; it was the city's heaviest snowfall on record. 9:00am Snow Update: LATEST: Slow and steady would describe the snowfall here in the viewing area. Here is the l... http://bit.ly/fhLvqv RT @13News: #Snow for the record books: NORFOLK -- Sunday's snowfall was the 3rd heaviest on record for Norfolk.... http://bit.ly/fevHrz 'Tis the season to be... snowy! Snowfall has begun in the northern areas of the U.S. Share your experiences here: http://on.cnn.com/fuxyVc	RT @colbertema: Winter Weather Advisory until 6pm today. Moisture is bringing moderate snow showers. New snow accumulation of 1" likely. 'Tis the season to be... snowy! Snowfall has begun in the northern areas of the U.S. Share your experiences here: http://on.cnn.com/fuxyVc I hate snow.. It's so.. Snowy Crap!! Looking at forecast for next 10 days ... Rain & Snow showers are in for 12/11 #SantaCon! PLEASE, PLEASE, PLEASE DON'T MESS UP MY DAY!
2 (Positive) *Processed Tweets*	apple sued over privacy apples black	rt apples latest ipad ad is magically amazing	rt obama praises the success of apples steve jobs	rt apples new energy efficient devices arent so great for the environment

	friday shopping event starts in the us apples ipad helps israeli hospital treat patients reuters rt apples new energy efficient devices arent so great for the environment	rt mashable news apples black friday shopping event starts in the us apples new energy efficient devices arent so great for the environment rt apples ipad helps israeli hospital treat patients	rt apples black friday shopping event starts in the us rt apples latest ipad ad is magically amazing apples new energy efficient devices arent so great for the environment	apples patent may unlock 3d technology googles new android music player how did the apple logo come to be
3 (Positive) *Processed* *Tweets*	privacy ?? facebook follow! pls www.facebook.com preprocessdoc_em1 review at&t facebook, sort at&t announced feature facebook page follow! pls preprocessdoc_em6	change! #pray follow! pls preprocessdoc_em6 greatest surveillance history: #facebook #privacy #vrm today's chance entered win dallas stars tickets! >>> ...	pretty cool vicks searching dedicated nfl fan sending sb. check facebook.com/nyq ... "celebrate shelter pets day" facebook -- post great shelter pets! tsa giddy thanksgiving holiday! biggest invasion personal privacy adolf ... privacy facebook ? top stories today brodkin storace sharma freytes	follow! pls preprocessdoc_em1 facebook - tsa giddy thanksgiving holiday! biggest invasion personal privacy adolf hitler! facebooks profiles impact privacy -
4 (Positive) *Processed* *Tweets*	china can cap inflation next year regulator china again hikes interest rate in inflation fight they must be crazy in china raising prices helps inflation wen confident china can contain inflation rt china moves to cool its inflation	china raises rates to fight inflation rt wen says china confident of keeping inflation in check rt china moves to cool its inflation canadas inflation rate eases to 2	rt expect blistering inflation and two more chinese rate hikes by the end of the year by canadas inflation rate eases to 2 rt life is easier for western expatriates in china than it is for chinese expatriates in the west rt china rate move prompts mixed reaction a mixed reaction to chinas christmasday rate interest rate	rt china raises interest rates again to cool inflation china moves to cool its inflation cables blame chinese for google hacking china censorship technology worldnews inflation heading in different directions in china and the us

			increase le httpbitl	
5 (Positive) *Processed* *Tweets*	unicef executive director anthony lake desmond tutu discuss generation born free hiv aids reach hope move closer day eliminate hiv/aids face earth. http:// ... #worldaidsday. best defense hiv/aids people, esp youth, fight hiv/aids	aids breakthrough: daily pill helps risk hiv/aids church plans hiv/aids forum: "thou shalt love lord thy god thy heart, thy soul, ... #worldaidsday. best defense hiv/aids people, esp youth, hope move closer day eliminate hiv/aids face earth. http:// ...	elton john people's hero - music - work hiv/aids heroes? read ... unicef executive director anthony lake desmond tutu discuss generation born free hiv aids reac ... church plans hiv/aids forum: "thou shalt love lord thy god thy heart, thy soul, ... http://tinyu ... hope move closer day eliminate hiv/aids face earth. http:// ...	govt concerned cost hiv/aids hope move closer day eliminate hiv/aids face earth. http:// ... salute big homie helpin' raise hiv/aids awareness #thinkred #redalbum elton john people's hero - music - work hiv/aids heroes? read ...
6 (Positive) *Processed* *Tweets*	robust retail sales lift merchants' holiday spirits retail sales boost growth prospects #topnews november retail sales signal strong holiday shopping season promo: 2-for-1 retail shop perfect last-minute gift sweet	sick shopping? : retail experiments! indie #design goods! available text! http:// ... november retail sales top forecasts (international herald tribune): share frien... #ec ... retail special: big sale!! enjoy 50% sale items noon dec 26 shops.	shopping japanese dept. store biggest journal/stationery section hea ... verdict retailers holiday sales jump 5.5%, best year terms growth 2005. #retail november retail sales signal strong holiday shopping season retail sales posted strong gain november holiday shopping season solid start:	retail sales boost growth prospects november retail sales signal strong holiday shopping season retail special: big sale!! enjoy 50% sale items noon dec 26 shops. good vintage store online! wheee! #vintageshopping

C. Process Overview

The following is a demonstration of the proposed methodology. A small example is used, with a few tweets.

Step 1 and 2: Obtain corpus of tweets for a topic. Suppose the topic is "Snowfall" and the tweets are:

- wow crazy weather around the world high elevations of ca could get 15 ft of snow
- mountains news epic storm could drop 8 feet of snow on colorado high country
- hey tweeties hope you had a blessed day we had snow here today just a little but still saw tons if accidents
- rt powerful western storm dumps inches of rainfeet of snow for ca today 12ft wasatch and rockies too storm warms u
- rt mornin all woke up to snow falling outside my house today first snowfall of the season for us beautiful
- wow 13 ft of fresh snow in our mountainsguess there is an upside to 7 days of rain have fun so cal skiers
- even though its snowing outside today is a very nice day
- rt my goodness its snowing really hard here and its only 500 ft elevation
- how to keep airports open even at 2 ft of snow in helsinki which hasnt been closed since cont
- rt the uk continues to reel from a few inches of snowbut im trying to think of a way to get to the 2 ft of powder that hit
- i know the snow is bad but an ice storm is really bad i wondered if it would be heavy wet snow instead of the powder kind
- powerful western storm dumps inches of rainfeet of snow for ca today 12ft wasatch and rockies too storm warms up the east late wk
- bkken i have the best tree ever its like 20 tall the snow just made it amazing preprocessdocem1 smh its 10 ft
- snow showers continue today in indy area
- snow showers and squalls will increase today some will be heavy at times leading to quick accumulations and snow covered roads

Step 3: Classify each tweet as positive or negative. For simiplicity, we will not classify for neutral tweets.

Positive Tweets:

- rt powerful western storm dumps inches of rainfeet of snow for ca today 12ft wasatch and rockies too storm warms u
- rt mornin all woke up to snow falling outside my house today first snowfall of the season for us beautiful
- wow 13 ft of fresh snow in our mountainsguess there is an upside to 7 days of rain have fun so cal skiers
- hey tweeties hope you had a blessed day we had snow here today just a little but still saw tons if accidents

Negative Tweets:

- wow crazy weather around the world high elevations of ca could get 15 ft of snow
- mountains news epic storm could drop 8 feet of snow on colorado high country
- rt powerful western storm dumps inches of rainfeet of snow for ca today 12ft wasatch and rockies too storm warms u
- rt my goodness its snowing really hard here and its only 500 ft elevation
- how to keep airports open even at 2 ft of snow in helsinki which hasnt been closed since cont
- rt the uk continues to reel from a few inches of snowbut im trying to think of a way to get to the 2 ft of powder that hit
- i know the snow is bad but an ice storm is really bad i wondered if it would be heavy wet snow instead of the powder kind
- powerful western storm dumps inches of rainfeet of snow for ca today 12ft wasatch and rockies too storm warms up the east late wk
- bkken i have the best tree ever its like 20 tall the snow just made it amazing preprocessdocem1 smh its 10 ft
- snow showers continue today in indy area
- snow showers and squalls will increase today some will be heavy at times leading to quick accumulations and snow covered roads

Step 4: Create Word Graphs and extract aspects. Suppose after creating Word Graphs and clustering, the following aspects are formed:

Positive	Negative
Aspect 1: • snow • ft • beautiful	Aspect 1: • ft • hard • elevation
Aspect 2: • accidents	Aspect 2: • snow • showers

Step 5: Rank each aspect by sentiment temperature.

Positive	Negative
Aspect 1 - Rank 1 - 15% positive	Aspect 2: - Rank 1 - 30% negative
Aspect 2 - Rank 2 - 5% positive	Aspect 1 - Rank 2 - 10% negative

Step 6: Assign each tweet to an aspect.

Positive	Negative
Aspect 1 - Rank 1 - 15% positive • rt powerful western storm dumps inches of rainfeet of snow for ca today 12ft wasatch and rockies too storm warms u	Aspect 2: - Rank 1 - 30% negative • snow showers continue today in indy area • snow showers and squalls will

- rt mornin all woke up to snow falling outside my house today first snowfall of the season for us beautiful
- wow 13 ft of fresh snow in our mountainsguess there is an upside to 7 days of rain have fun so cal skiers

Aspect 2 - Rank 2 - 5% positive

- hey tweeties hope you had a blessed day we had snow here today just a little but still saw tons if accidents

increase today some will be heavy at times leading to quick accumulations and snow covered roads

- mountains news epic storm could drop 8 feet of snow on colorado high country
- i know the snow is bad but an ice storm is really bad i wondered if it would be heavy wet snow instead of the powder kind

Aspect 1 - Rank 2 - 10% negative

- wow crazy weather around the world high elevations of ca could get 15 ft of snow
- rt powerful western storm dumps inches of rainfeet of snow for ca today 12ft wasatch and rockies too storm warms u
- rt my goodness its snowing really hard here and its only 500 ft elevation
- how to keep airports open even at 2 ft of snow in helsinki which hasnt been closed since cont
- rt the uk continues to reel from a few inches of snowbut im trying to think of a way to get to the 2 ft of powder that hit
- powerful western storm dumps inches of rainfeet of snow for ca today 12ft wasatch and rockies too storm warms up the east late wk
- bkken i have the best tree ever its like 20 tall the snow just made it amazing preprocessdocem1 smh its 10 ft

Step 7: Provide a summary for each aspect. Since we are using a small example, we shall pick just one representative document for each sentiment/aspect pair.

Positive	Negative
Aspect 1 - Rank 1 - 15% positive • rt mornin all woke up to snow falling outside my house today first snowfall of the season for us beautiful	Aspect 2: - Rank 1 - 30% negative • snow showers and squalls will increase today some will be heavy at times leading to quick accumulations and snow covered roads
Aspect 2 - Rank 2 - 5% positive • hey tweeties hope you had a blessed day we had snow here today just a little but still saw tons if accidents	Aspect 1 - Rank 2 - 10% negative • powerful western storm dumps inches of rainfeet of snow for ca today 12ft wasatch and rockies too storm warms up the east late wk

References

Abel, F., Gao, Q., Houben, G. J., & Tao, K. (2011). Analyzing user modeling on twitter for personalized news recommendations. In *User Modeling, Adaption and Personalization* (pp. 1-12). Springer Berlin Heidelberg.

Abel, F., Gao, Q., Houben, G. J., & Tao, K. (2011). Semantic enrichment of twitter posts for user profile construction on the social web. In *The Semanic Web: Research and Applications* (pp. 375-389). Springer Berlin Heidelberg.

Abel, F., Gao, Q., Houben, G. J., & Tao, K. (2011, June). Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In *Proceedings of the 3rd International Web Science Conference* (p. 2). ACM.

Ackermann, M. R., Blömer, J., Kuntze, D., & Sohler, C. (2014). Analysis of agglomerative clustering. *Algorithmica*, *69*(1), 184-215.

Al-Kouz, A., & Albayrak, S. (2012, August). An Interests Discovery Approach in Social Networks Based on Semantically Enriched Graphs. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on* (pp. 1272-1277). IEEE.

Asiaee T, A., Tepper, M., Banerjee, A., & Sapiro, G. (2012, October). If you are happy and you know it... tweet. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (pp. 1602-1606). ACM.

Bhattacharya, P., Zafar, M. B., Ganguly, N., Ghosh, S., & Gummadi, K. P. (2014, October). Inferring user interests in the twitter social network. In *Proceedings of the 8th ACM Conference on Recommender systems* (pp. 357-360). ACM.

Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, *2008*(10), P10008.

Chakrabarti, D., & Punera, K. (2011). Event Summarization Using Tweets. *ICWSM*, *11*, 66-73.

Diakopoulos, N. A., & Shamma, D. A. (2010, April). Characterizing debate performance via aggregated twitter sentiment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1195-1198). ACM.

Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457-479.

Esuli, A., & Sebastiani, F. (2006, May). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC* (Vol. 6, pp. 417-422).

Fani, H., Zarrinkalam, F., Zhao, X., Feng, Y., Bagheri, E., & Du, W. (2015). Temporal Identification of Latent Communities on Twitter. *arXiv preprint arXiv:1509.04227*.

Ferragina, P., & Scaiella, U. (2010, October). Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management* (pp. 1625-1628). ACM.

Filatova, E., & Hatzivassiloglou, V. (2004, July). Event-based extractive summarization. In the *Proceedings of ACL Workshop on Summarization*, Barcelona, Spain. Association of Computational Linguistics.

Goldstein, J., Mittal, V., Carbonell, J., & Callan, J. (2000, November). Creating and evaluating multi-document sentence extract summaries. In *Proceedings of the ninth international conference on Information and knowledge management* (pp. 165-172). ACM.

Huang, J., Lu, J., & Ling, C. X. (2003, November). Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on* (pp. 553-556). IEEE.

Kapanipathi, P., Jain, P., Venkataramani, C., & Sheth, A. (2014). User interests identification on twitter using a hierarchical knowledge base. In *The Semantic Web: Trends and Challenges* (pp. 99-113). Springer International Publishing.

Lin, C.-Y. and Hovy, E. (2003) Automatic Evaluation of Summaries Using N-gram Co-occurrence Statistics. *NAACL '03: Proc. 2003 Conf. North American Chapter of the Association for Computational Linguistics on Human Language Technology*, Edmonton, Alberta, Canada, pp. 71–78. Association for Computational Linguistics, Stroudsburg, PA, USA

Louis, A., & Nenkova, A. (2009, August). Automatically evaluating content selection in summarization without human models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1* (pp. 306-314). Association for Computational Linguistics.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval* (Vol. 1, No. 1, p. 496). Cambridge: Cambridge university press.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.

Marcus, A., Bernstein, M. S., Badar, O., Karger, D. R., Madden, S., & Miller, R. C. (2011, May). Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 227-236). ACM.

McKeown, K., Klavans, J., Hatzivassiloglou, V., Barzilay, R., & Eskin, E. (1999, July). Towards multidocument summarization by reformulation: Progress and prospects. In *AAAI/IAAI* (pp. 453-460).

Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of multivariate analysis*, *98*(5), 873-895.

Michelson, M., & Macskassy, S. A. (2010, October). Discovering users' topics of interest on twitter: a first look. In *Proceedings of the fourth workshop on Analytics for noisy unstructured text data* (pp. 73-80). ACM.

Mihalcea, R., & Tarau, P. (2004, July). TextRank: Bringing order into texts. In Lin, D., & Wu, D. (Eds.), *Proceedings of EMNLP 2004*, pp.404-411 Barcelona, Spain. Association for Computational Linguistics.

Murnane, E. L., Haslhofer, B., & Lagoze, C. (2013, May). Reslve: leveraging user interest to improve entity disambiguation on short text. In *Proceedings of the 22nd international conference on World Wide Web companion* (pp. 1275-1284). International World Wide Web Conferences Steering Committee.

Murtagh, F., & Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion?. *Journal of Classification*, *31*(3), 274-295.

Nenkova, A., & Vanderwende, L. (2005). The impact of frequency on summarization. *Microsoft Research, Redmond, Washington, Tech. Rep. MSR-TR-2005-101*.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, *74*(3), 036104.

Nichols, J., Mahmud, J., & Drews, C. (2012, February). Summarizing sporting events using twitter. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces* (pp. 189-198). ACM.

Ohsawa, Y., Benson, N. E., & Yachida, M. (1998, April). KeyGraph: Automatic indexing by co-occurrence based on building construction metaphor. In *Research and Technology Advances in Digital Libraries, 1998. ADL 98. Proceedings. IEEE International Forum on* (pp. 12-18). IEEE.

Radev, D. R., Jing, H., Styś, M., & Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management*, *40*(6), 919-938.

Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, *105*(4), 1118-1123.

ROUGE 2.0 - Java Package for Evaluation of Summarization Tasks with Updated ROUGE Measures. (n.d.). Retrieved April 13, 2016, from http://kavita-ganesan.com/content/rouge-2.0

S. van Dongen, Performance criteria for graph clustering and Markov cluster experiments, Technical Report INS-R0012, Centrum voor Wiskunde en Informatica, 2000.

Saggion, H., Torres-Moreno, J. M., Cunha, I. D., & SanJuan, E. (2010, August). Multilingual summarization evaluation without human models. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (pp. 1059-1067). Association for Computational Linguistics.

Sagolla, D. (2009). How Twitter Was Born. Retrieved from http://www.140characters.com/2009/01/30/how-twitter-was-born/

Saif, H., Fernandez, M., He, Y., Alani, H.: Evaluation datasets for twitter sentiment analysis. In: *Proceedings, 1st Workshop on Emotion and Sentiment in Social and Expressive Media (ESSEM)* in conjunction with *AI*IA Conference*. Turin, Italy (2013)

Sharifi, B. P., Inouye, D. I., & Kalita, J. K. (2014). Summarization of twitter microblogs. *The Computer Journal*, *57*(3), 378.

Shen, W., Wang, J., Luo, P., & Wang, M. (2013, August). Linking named entities in tweets with knowledge base via user interest modeling. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 68-76). ACM.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)* (Vol. 1631, p. 1642).

Speriosu, M., Sudan, N., Upadhyay, S., Baldridge, J.: Twitter polarity classification with label propagation over lexical links and the follower graph. In: Proceedings of the EMNLP First workshop on Unsupervised Learning in NLP. Edinburgh, Scotland (2011)

Twitter Annual Report 2014. (n.d.). Retrieved April 12, 2016, from http://www.viewproxy.com/twitter/2015/1/index.html#/44

Wu, F., & Huberman, B. A. (2004). Finding communities in linear time: a physics approach. *The European Physical Journal B-Condensed Matter and Complex Systems*, *38*(2), 331-338.