THE UNIVERSITY
of EDINBURGH

A Novel Platform for Topic Group Mining, Crowd Opinion Analysis and Opinion Leader Identification in On-line Social Network Platforms

Cheng-Lin Yang

Doctor of Philosophy

Centre for Intelligent Systems and their Applications
School of Informatics
University of Edinburgh

2020

Abstract

In recent years, topic group mining and massive crowd opinion analysis from on-line social network platforms have become some of the most important tasks not only in research area but also in industry. Systems of this sort can identify similar topics from a very large dataset, group them together based on the topic, and analyse the inclination of the content's owner. To solve this problem, which involves research from a number of different areas, an integrated platform needs to be proposed.

Most community mining techniques treat the network as a graph where nodes represent users and edges reflect user relationship between two users. One obvious drawback of these approaches is that it can only utilise the explicit user relationships provided by on-line social network platforms. All other possible relationships will be ignored. Some on-line social network platforms restrict the length of content a user can publish. This causes traditional document clustering methods to perform poorly. Meanwhile, the restriction of content length also affects opinion mining performance since most content lacks contextual features. Hence, other context features that are not immediately or obviously related need to be investigated to improve performance in user inclination classification.

This research proposes a novel three layered platform. Two core technologies of the platform are topic group mining and user inclination analysis. The integrated approach was evaluated by a series of experiments to examine each core technology. The results indicate that the proposed integrated platform is able to produce the following results. 1) Scores up to 0.82 by V-measure evaluation function in topic group mining. 2) High accuracy rate in inclination mining. 3) A flexible and adaptable platform design which can accommodate different on-line social networks easily.

Acknowledgements

First and foremost, I owe my deepest gratitude to my supervisors, Dr. Jessica Chen-Burger and Dr. David Robertson for their guidance and continuous support throughout this exciting and challenging journey. Jessica for introducing me this interesting research topic and the weekly meetings with her were inspiring and full of lively discussions. Dave is the most widely knowledgeable I have ever met, this thesis cannot be completed without his insightful advices. To my external examiners, Dr. George Coghill and Dr. Harith Alani, thank you for the critical but encouraging viva, which provided valuable suggestions for me to improve my work greatly. At the meantime, I would like to express sincere gratitude to my internal examiner Dr. Michael Rovatsos. Not only for his brilliant feedback during the viva, but also for his patient to arrange the viva that worked for different time zones.

A big thank you to Dr. Gayathri Nadarajan for bringing me to F4U project (Fish4Knowledge). This gives me a chance to build up a user-facing workflow system, the experience that I also utilised to the proposed framework of this thesis. Meanwhile, I would like to thank Dr. Fang-Pang Lin at NCHC, Taiwan. He kindly provided a large and powerful high performance computing cluster, which allows me to build up the invaluable knowledge of handling massive data in parallel environment.

My deep appreciation goes to my dear friends who have given me many encouragement during the journey; Abby Chen, Yu-Tzu Liu, Yu-Pei Hong, Pei-Yu Hung, Pei-Yun Yu, Jérôme Chassagnard, Shang-I Tsai, Ching-Fan Yang, Hui-Chin Huang and Chang-Fu Tsai.

My parents Po-Kuei Huang and Chen-Shen Yang have always believed in me, even when I was in the worst setback moment of my research. Finally, words cannot express the debt of gratitude and love for my wife Shin-Ting Liu, who has been by my side throughout this PhD journey.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified. The following are the principal publications derived from this thesis:

- C.-L. Yang and Y.-H. Chen-Burger. On-line communities making sense: a hybrid micro-blogging platform community analysis framework. In Agent and Multi-Agent Systems. Technologies and Applications, pages 134–143. Springer, 2012.
- C.-L. Yang, N. Benjamasutin, and Y.-H. Chen-Burger. Mining hidden concepts: Using short text clustering and wikipedia knowledge. In Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on, pages 675–680. IEEE, 2014.
- C.-L. Yang and Y.-H. Chen-Burger. A hybrid on-line topic groups min- ing platform. In Agent and Multi-Agent Systems. Technologies and Ap- plications. Springer, 2015.

(*Cheng-Lin Yang*)

Other Research Publications

The academic results of Fish4Knowledge[1] Project are listed below:

- **Book Chapter:**

  1. G. Nadarajan, C.-L. Yang and Y.-H. Chen-Burger, "Chapter 9 - Intelligent Workflow Management for Fish4Knowledge using the SWELL System" in Springer Big Data for Marine Biology – A F4K Story – An Integrated Inter–disciplinary Computational Approach
  2. G. Nadarajan, C.-L. Yang and Y.-H. Chen-Burger, "Appendix - Database Tables Related to F4K Workflow" in Springer Big Data for Marine Biology – A F4K Story – An Integrated Inter–disciplinary Computational Approach

- **Conference Publications:**

  1. G. Nadarajan, C.-L. Yang, Y.-H. Chen-Burger. "Multiple Ontologies Enhanced with Performance Capabilities to Define Interacting Domains within a Workflow Framework for Analysing Large Undersea Videos", 5th International Conference on Knowledge Engineering and Ontology Development (KEOD 2013), Portugal, Sept 2013.
  2. G. Nadarajan, C.-L. Yang, Y.-J. Cheng, S.-I. Lin, Y.-H. Chen-Burger, F.-P. Lin. "Real-time Data Streaming Architecture and Intelligent Workflow Management for Processing Massive Ecological Videos." International Conference on Social Computing (SocialCom) 2013, pp. 1074-1080, Washington, USA, Sept 2013.

[1] http://fish4knowledge.eu/

Dedicated to my beloved parents and my lovely wife
For their love, encouragement and endless support

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

As a result of the rapid development of the Internet, it is now accessible in all parts of the world. Moreover, because of the drastic functional improvements in computing devices, computing devices have shrunk from the sizes of rooms to sizes that can be held in a person's hand, as seen with smartphones and tablets. With the rapid development of communication technology, mobile Internet has become an indispensable utility in daily life. People exchange information and knowledge on the Internet through various devices, thus forming a large social network. Consequently, new types of communication micro-blogging platforms, such as Twitter[1], Plurk[2] and Yahoo Meme[3], have been gaining popularity since 2007. These platforms quickly attracted the attention of many users, including many celebrities, politicians, and television and sports stars, who use these platforms to share their daily lives and personal opinions. Because the growth rate of registered users is extremely high, enterprises have begun using such platforms as a public relation tool to announce news or to provide the latest promotional information. These platforms also serve as a new type of news source, through which journalists are able to search for upcoming breaking news. For example, the first report of the emergency landing of a US Airways flight on the Hudson River was posted on Twitter [Beaumont, 2011].

When using a traditional blog system, e.g., Blogger and Wordpress, the user has to organize several paragraphs to form a blog post, which

[1] https://twitter.com/
[2] http://plurk.com/
[3] Service closed on 25/05/2014

becomes problematic if the user wishes to share his current feelings in only a few words or sentences or to share interesting photos with a few comments. Such posts would be considered odd by blog viewers because they expect a full article rather than a few sentences or photos without descriptions.

A micro-blogging system successfully addresses the needs of this type of user. This type of system is referred to as "micro" because it typically limits either the word count or the number of characters to 200 words. In addition, unlike the traditional blog system, in which the user has to compose the article on a personal computer, a micro-blogging system makes use of various communication technologies. The user can submit a micro-blog post from his desktop, laptop, tablet, or smartphone or even by SMS. An individual micro-blog post can also contain video, image or audio links with a few comments. Followers of a micro-blog will be automatically notified by the system so that they can immediately respond to the post after it has been posted.

The content of a micro-blog is considered a document. Unlike traditional documents, the majority of the content of micro-blogs is not written by well-trained professionals. The semantics and wording of the content will not be conscientious, careful or subjective compared to content written by professionals such as commentators, writers or journalists. Most micro-blogs are written in a casual style and can thus be blunt and easy to read and understand. Furthermore, users of micro-blogging platforms often express their opinions on and inclinations about topics or products. Therefore, we observe many emotional terms, e.g., like, awesome, and useless, in the user's content. This information will be extremely valuable for experts in various fields for making crucial decisions if such terms can be processed and analysed automatically.

However, the considerable amount of information scattered across micro-blogging platforms makes it impossible to understand which topics people are discussing and their inclinations towards specific topics via manual analyses. Certain proposed intelligent systems are capable of extracting topic information from a set of documents, but most documents are well formed and focus on a specific topic. In contrast, a typical micro-blog post generally has a poorly defined or no explicit topic, which

results in difficulties in extracting information from these proposed systems. This problem is exacerbated by the nature of micro-blogs, which limits the length of the content to a certain number of characters. For this purpose, combinations of large data processing techniques, including topic clustering approaches and user inclination analysis methods are investigated. These are motivated by the requirements listed in Section 1.2. First, an overview of the research questions and the goal of this thesis are presented.

1.1 Research Hypothesis and Claims

The hypothesis underlying the research presented in this thesis is as follows:

> *A flexible and adaptable platform that integrates organised text-based resources of generic world knowledge (such as WikiPedia pages) and sentiment mining clustering techniques is able to assist users in analysing the inclination of the crowd towards a certain topic from a large data source.*

Based on the requirements listed in the next section, a three-layered platform that utilises information extraction, topic clustering and user inclination clustering will be designed and implemented. The main claims made in this thesis are as follows:

1. The entire process, including data collection, topic extraction and clustering and user inclination clustering, is novel that can be and had been automated with minimal user interaction. The use of this approach provides an efficient working platform for performing massive on-line crowd inclination analyses, which are typically time-consuming and human-labour-intensive tasks.

2. Conducting topic extraction and clustering on very short and unstructured documents in accordance with a crowd-sourced knowledge base provides greater accuracy compared with those that had applied traditional information retrieval techniques.

3. A user inclination analysis enhanced with the added user relationships obtained from automatically generated topic groups that are clustered from a large number of very short and unstructured documents can be conducted to label the inclination of users' documents with promising precision. This provides the foundation for future studies.

1.2 Questions and Requirements

Several questions were addressed for this thesis:

1. What are the requirements for a suitable system that can provide automated assistance for non-technical users to conduct user topic inclination analysis tasks? Additionally, what are the relevant/useful technologies that may be useful in this context?

   This question examines the methodologies that should be considered to solve the task at hand. A deeper question is how can the different methodologies and technologies work together to meet these requirements? It is clear that the proposed framework needs to provide a high level of automation to considerably reduce the massive workload involved in traditional manual processing. It is therefore useful to determine which mechanisms are suitable for this task.

2. Knowledge bases are generally helpful as an information querying source for modern expert applications. In this context, what will be a suitable knowledge base for our system? What information needs to be included? What are the suitable representations for this? What are the relevant problem solving/inference knowledge?

   One of the research gaps that we need to address is the extraction of topic(s) from an incomplete paragraph or from micro-blog posts with insufficient information. For this purpose, how the knowledge base will be introduced in accordance with the topic-extraction technique will need to be investigated. The knowledge base must acquire a reliable source that also has comprehensive coverage of different fields and that is freely open to the public. Meanwhile, the topic should be added in a timely manner when a new topic

is emerging. Furthermore, the role of the knowledge base in topic classification should also be investigated. A knowledge base will be considered suitable if it meets all requirements aforementioned.

3. Machine learning technologies have been successful for clustering information found in on-line social network platforms such as Facebook[4] and Google+[5]. What type of clustering methodology would be suitable for generating the topic groups and user inclinations problems that we are attempting to address in this thesis?

   Many studies use machine learning algorithms to cluster users on on-line social network platforms based on their relationships or linkages. However, applying a combination of natural language processing (NLP) techniques, information retrieval and machine learning to an on-line social network is a relatively new approach. There is often a trade-off between the different technologies and between time and accuracy. Therefore, existing methodologies are explored to find a set of suitable algorithms that can not only achieve the expected accuracy but also process the data in a reasonable amount of time.

4. Sentiment classification methodologies are research topics that have been developed for decades. How would these methodologies perform in response to extra social relationships?

   The choice of applying sentiment classification to on-line social network platforms is clearly a challenge. The main concern is considering on-line social relationships to determine whether people are still influenced by their on-line "relatives". To make the proposed framework work, it is necessary to design an algorithm that takes on-line relationships into account.

5. It appears that we need a platform that integrates all techniques together; therefore, what are the requirements for such ideal platform?

   An ideal platform would be one that is efficient and effective for

[4]https://www.facebook.com/
[5]https://plus.google.com/

managing a considerably large amount of data (more than one million tweets). Additionally, an ideal platform needs to be sufficiently flexible to employ multiple classification methods such that users can fine tune the results by modifying some or all of the classification methods, e.g., one may consider the trade-off between an approach that takes less processing time but is less accurate *versus* an approach that requires more processing time but is more accurate. Hence, a successful platform should demonstrate how data can be handled by components within proposed framework without blocking the entire processing flow.

1.3 Problem Domain and Thesis Background

Consider a scenario in which a large set of micro-blog posts are continuously collected within a certain timeframe (e.g., three months) and saved in a database. Each micro-blog post has different lengths, and together, they contain users' inclinations towards specific topics. The posts are available to users who traditionally analyse them manually. To provide a reasonable context, consider that a large set of micro-blog posts is presented to marketing analysts. The marketing analysts perform data sampling and cleaning, topic filtering and user inclination analysis. The data sampling and cleaning into a smaller dataset is essential to the final result. For instance, those datasets that are too small can produce biased results, and datasets that are too large may dilute the results due to an excessive number of unrelated posts. Topic filtering involves identifying and eliminating unrelated posts that are not applicable to further analysis. User inclination analysis requires that analysts determine the inclination of the posts for the topic of interest and examine users' relationships between each other because people tend to be influenced by their close friends. However, conducting the analysis described above is extremely time consuming and laborious. This thesis attempts to provide a form of automation to make these processes less labour intensive. To achieve this goal, we need to investigate a combination of several techniques within a layered framework. This framework is designed to satisfy users' interests with an informed result. Figure 1.1 presents a pictorial

overview of this scenario.

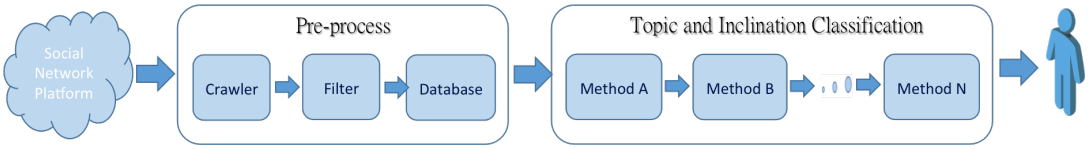

Figure 1.1: Overview of research story

To successfully perform the scenario illustrated in Figure 1.1, a successful methodology must perform automated data collection, transform the data into a machine-processable format, and then find a set of tools that can process the given data and generate results for the user. For this purpose, a three-layered framework is proposed.

First, we review the existing methodologies that provide automated support for clustering large data sets. Their abilities, relevance, contributions and limitations will be discussed. Additionally, we will identify the gaps between these methodologies and our goals, which will be discussed in Chapter 2.

To address the research gaps within existing methodologies, a novel three-layered framework for automatically clustering topic groups and detecting user inclinations within each topic group is devised and presented in Chapter 3. This framework will utilise two major methodologies: short-text topic group clustering and user inclination with user relationship analysis. Moreover, this framework will also utilise basic information retrieval tools and a well-known distributed database system.

One of the major goals of this thesis is to help identify topic groups from a large short-text posting dataset. Due to the natural limitations of the data, a large number of posts may provide little knowledge that will cause the mis-clustering of groups. This thesis also seeks to integrate, within the framework, an established crowd-sourced knowledge database. Chapter 4 describes the topic group clustering method developed to achieve our objectives.

A set of clustered topic groups will be required to detect the inclination of posts in each topic group. Chapter 5 will introduce two different

user inclination detection models so that we can evaluate the accuracy in a later chapter. Furthermore, in recent years, researchers have also focused on user relationships because of the emergence of social network platforms. Some of these researchers believe that the real-world influences among friends will also be reflected on cyber platforms. This is the key difference between traditional websites and social network platforms. We also devise a user relationship graph based on platform-specific user relationships to assist us in analysing a user's inclination.

The three-layered framework will be evaluated on a large test dataset. Chapter 6 presents a set of experiments for evaluating two major layers in terms of accuracy and user satisfaction. Finally, the contributions of this thesis are highlighted in Chapter 7.

1.4 Research Contributions

The main contributions of this thesis are outlined as follows:

- **Novel Mechanism for Automated User Inclination Platform:** The most important accomplishment of this thesis is that it provides an evaluation platform for the growing field of on-line social network user inclination analysis. This new platform enables a flexible pick-and-mix of suitable technologies to work together in a new system; while its generated results can be easier evaluated using our evaluation framework, as reported in this thesis.

- **High Accuracy in Topic Group Mining:** Providing accurately classified topic groups is another major achievement of the work in this thesis. Two different approaches are introduced for document enrichment, which add Wikipedia topics and categories to original tweets, into our platform because we discovered two major problems with the bag-of-words model, which we will discuss in Section 4.7. From the experiments that we conducted in Chapter 6, the resulting topic groups clustered by enriched tweets were significantly better than those that used original tweets.

- **Novel Approach for Inclination Mining for Short-Text Communications:** Analysing the inclination of a short text is consid-

erably more difficult than that of traditional documents based on textual features because the 140-character limitation of a tweet is too short to identify sentiments in many cases. Hence, we attempt to utilise other features that are not immediately or obvious related to content to analyse the opinion of tweets. This thesis proposes a novel inclination mining method that utilises the content of tweets and user relationships. Based on the evaluation result in Chapter 6, the proposed method is able to produce an accurate result.

Chapter 2

Background and Literature Review

Chapter 1 presented a list of research questions that address the gaps in current technology by highlighting the large amount of data that leads to the impossibility of manually clustering topic groups and analysing user inclination. Social network analysis (SNA) is commonly used. Hence, an automated system must be developed to perform such tasks, thereby reducing human workloads. In Section 1.2, the requirements of a suitable system were outlined. In this chapter, social network analysis techniques, text clustering algorithms and sentiment analysis methods will be investigated to identify approaches that represent the state of the art.

2.1 Trends of Communication Patterns

With the rapid development of communication technology, the Internet has become an indispensable utility in daily life. People exchange information and knowledge over the Internet through various devices, thereby forming a large social network and various types of on-line communities. The platforms of these on-line communities have transformed from text-based to multimedia-based Web 1.0 (Section 2.1.2) and into the user-centric Web 2.0 (Section 2.1.3). With these new communication technologies, users use forums and blogging platforms to share their knowledge and experience with multimedia resources. By using various

search engines, such as Yahoo[1] and Google[2], users who are interested in particular topics can quickly and easily obtain relevant information. Such users are able to discuss specific topics with other users from different countries via the Internet.

Unlike communities in the traditional sense, members of an on-line community are not restricted to the same geographical area. An on-line community can be defined as a social phenomenon formed by a group of people who communicate with each other through the Internet. Interactions on the Internet satisfy peoples' interests and fantasies and lead to the development of social relationships [Romm et al., 1997].

2.1.1 Text-based Platforms

At the end of the 1970s, Usenet[3] represented a large fraction of available topic-based newsgroups and was the largest information exchange platform at the time. Users were able to post and retrieve rich information from newsgroups. Members communicated via email or by posting to a distributed Usenet server via a native client. Asynchronous communication made interactions between members difficult because the technology available at the end of the 1970s was not able to notify users when new information was posted. This delayed communication also caused the relationships among members to be looser.

The Bulletin Board Service[4] (BBS) was introduced a few years later to improve the user interaction experience. This system utilised Usenets's protocol as its underlying information-exchange mechanism and provided an internal messaging system that allowed a user to send a short instant message to another on-line user. This system provided users with the opportunity to communicate and share their opinions in real time. This function increased user interaction and resulted in users becoming familiar with other users more quickly. The BBS community could be organised more easily than the Usenet community. Today, BBS is still popular in Taiwan, Hong Kong and China, where users are attracted by

[1] http://yahoo.com/
[2] http://google.com/
[3] http://en.wikipedia.org/wiki/Usenet
[4] http://en.wikipedia.org/wiki/Bulletin_board_system

its fast response time and internal messaging system in addition to games that are provided by some BBSs.

2.1.2 Web 1.0

The World Wide Web (WWW), which combines both text and multimedia sources, was introduced in 1991. In contrast to Usenet and BBS, the WWW allows users to publish their own information or knowledge without asking for permission from publishers such as magazine editors. Mosaic[5], the first web browser, was introduced in 1993. Mosaic was a client installed on users' computers. This browser retrieved data from the Internet and displayed the organized content directly on a screen.

The impact of this innovation was tremendous because it allowed viewers worldwide to browse vast amounts of data from the browser as long as an Internet connection had been established. Unlike traditional publications, which readers need to read sequentially, WWW provides the *hyperlink* function, meaning that readers can easily move from website to website by clicking a link. This attribute enables people to obtain more information at a faster speed than ever before.

A forum system is an information platform that contains different sections, as shown in Figure 2.1. Each section focuses on one topic such as web hosting usage, Internet provider reviews and programming languages. As an increasing number of forums appear on the Internet, people have attempted to share their own experiences on forums. The role of forums is similar to that of the BBS, except that forums are now able to utilise the WWW. People can obtain more relevant information and rich multimedia on a single page. Furthermore, they can navigate to a reference website through a *hyperlink*. Consequently, forums quickly attracted many users, who subsequently formed different communities.

2.1.3 Web 2.0

In 2003, Dale Dougherty and Craig Cline proposed the term "Web 2.0" during a brain storming conference [O'Reilly, 2005]. Their concept was that all web applications should be designed to be interactive and enable

[5]http://en.wikipedia.org/wiki/Mosaic_(web_browser)

Figure 2.1: A snapshot of Forum sections listing

	Text-based platform	Web 1.0	Web 2.0
Multimedia	none	simple	rich
Content Versatility	text	simple	rich
Linkage	none	poor	strong
Notification	email	none	auto notification
Social Behaviour	few	low	strong

Table 2.1: Summary and comparison between text-based platform, Web 1.0 and Web 2.0

sharing and connectivity. For example, a user can import his contacts from Gmail to Facebook and recommend that his Facebook friends use Twitter.

Utilising this concept, different platforms can now create content and exchange information with each other. With the development of Web 2.0, people have developed more Internet-based social behaviours. They use del.icio.us to share websites that interest them, share their photographic works on Flickr, contribute their professional knowledge on Wikipedia, and place videos or interesting clips on YouTube. This has strengthened the relationships between each platform and created a sophisticated social network. Hence, it creates a lot of research opportunities. The growing amount of data and emerging platforms raise great concerns about how to gather massive information swiftly, manage data properly and conduct scientific experiments effectively. This thesis is also inspired by these interesting questions. To summarize, the key aspects of a text-based platform, Web 1.0 and Web 2.0 are listed in Table 2.1.

2.2 Identify Clusters in a Graph

Social network analysis (SNA) considers human communities and their corresponding relationships as a graph, as shown in Figure 2.2. In this type of graph, a node represents an individual, and the edge that connects two nodes indicates the relationship between two individuals [Wasserman and Faust, 1994]. By applying traditional clustering algorithms from graph theory, such as hierarchical clustering [Everitt et al., 2009] and k-means [Hartigan, 1975], we can locate clusters in a given graph. This is particularly useful for us to identify topic clusters in the social platform.

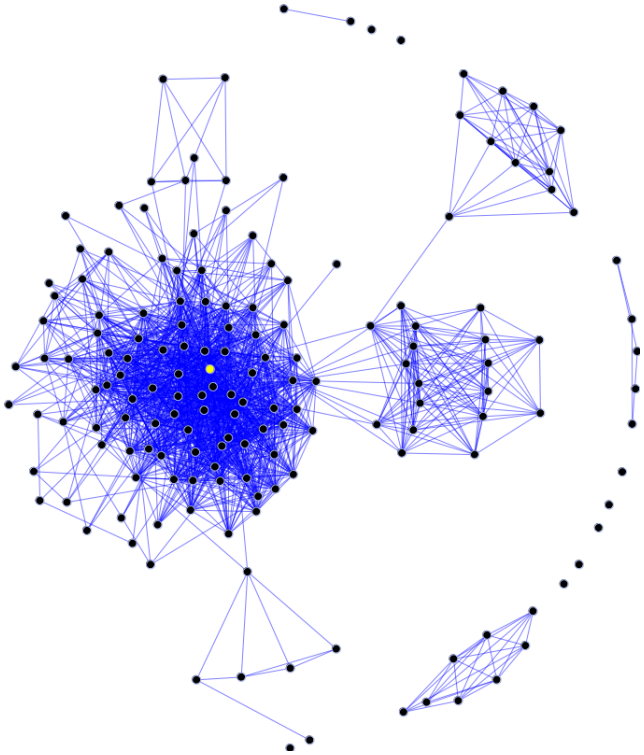

Figure 2.2: Example of a social network diagram

2.2.1 Traditional Approach

Hierarchical clustering treats each data point as a unique node. This approach finds similar nodes, which are determined based on the distance between two nodes calculated using a predefined distance function, and merges them as a new cluster. This procedure continues until all nodes are merged into new clusters. The benefits of hierarchical clustering are

as follows: 1) the algorithm is simple and easy to implement, and 2) it does not require the determination of the centre point among all nodes. As long as we know the distance between the nodes, the algorithm is able to find the clusters in the given data. However, the drawbacks of hierarchical clustering are as follows: 1) we cannot predict how many clusters will be generated, and 2) the computational complexity is $O(Ed\log N)$, where N is the number of nodes and E is the number of edges, which represents substantial computational complexity. When the given dataset is too large, the algorithm cannot generate the results in a reasonable amount of time.

In terms of the need to understand the micro-blogging system, hierarchical clustering is able to generate a maximum number of clusters because it will merge nearby nodes into a cluster, which is beneficial because we cannot predict how many types of communities will emerge in given data [Steinbach et al., 2000]. However, in terms of the data size of a real micro-blogging system, the hierarchical clustering algorithm cannot deliver results in an acceptable timeframe, which renders this algorithm infeasible for our scenario because we need to identify the on-line communities in a real-time micro-blogging system.

In contrast, in the k-means algorithm, k is the number of clusters that we expect to observe. Using this information, the algorithm will randomly choose k data points and consider these points as the centre points of clusters. Then, the algorithm calculates the distance from each data point to all centre points. The centre points then move to their closest data points. The procedure stops when no centre points can be moved. The k-means algorithm provides the following benefits: 1) fast convergence and 2) a computational complexity of $O(NVk)$, where N is the number of nodes and V is the number of vectors, which means that this algorithm requires fewer computational resources.

2.2.2 Local, Global and Score-based Approaches

Researchers have discovered that with traditional methods, a community is not well defined. Hence, newer methods attempt to analyse the network based on density. Researchers believe that the relationships within a

community will be considerably more dense than relationships outside the community. New methods have been proposed based on this assumption by other researchers.

2.2.2.1 Local Methods

Because researchers believe that a community in a social network should consist of denser relationships, we should be able to find a subgraph that is closely related in a given graph. The most well-known method consists of finding a *clique*. A clique is a subset of a graph that consists of three or more nodes, as shown in Figure 2.3. Each node in the clique must have an edge that is connected to all other nodes.

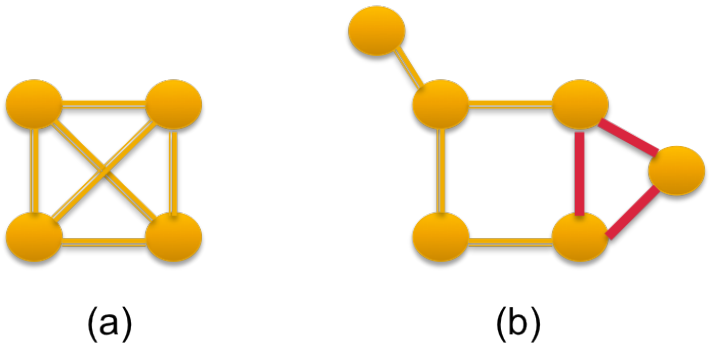

Figure 2.3: (a) Maximal clique and (b) a clique in a graph

It would be meaningless if the discovered *clique* is too small. However, finding the largest clique is an NP-hard problem [Garey and Johnson, 1990]. Moreover, the definition of a clique is restricted. A clique can collapse if some of the edges are lost, which is easy to observe in practical data. To solve this problem, several clique relaxation techniques have been proposed, including k-cliques [Alba, 1973] and k-clubs [Mokken, 1979].

The original definition of a clique restricts the distance from a node to every other node to be equal to one. The k-clique technique eases the distance limitation by requiring that the distances between any two nodes be less than k. Because the k-clique technique does not require that the shortest path between two nodes pass through the k-clique itself, the shortest path may pass through a node that does not belong to the

k-clique. Figure 2.4 demonstrates this scenario. When k=2, there are two 2-cliques in Figure 2.4: {1,2,3,4,5} and {2,3,4,5,6}. The distance between nodes 4 and 5 in the {1,2,3,4,5} clique is 3, which is larger than 2. The shortest path between node 4 and 5 entails passing through node 6. However, node 6 is not included in the {1,2,3,4,5} clique. Situations such as this cause two problems: 1) the distances between k-clique nodes may be longer than k, and 2) the nodes in the k-clique are potentially not connected.

To overcome the k-clique problem, k-clubs were proposed. The definition of k-clubs is that any distance between node pairs in a k-club group must be less than k, and the paths of node pairs must pass through the k-club's group. For example, in Figure 2.4, when k=2, there will be three 2-clubs: {1,2,3,4}, {1,2,3,5} and {2,3,4,5,6}.

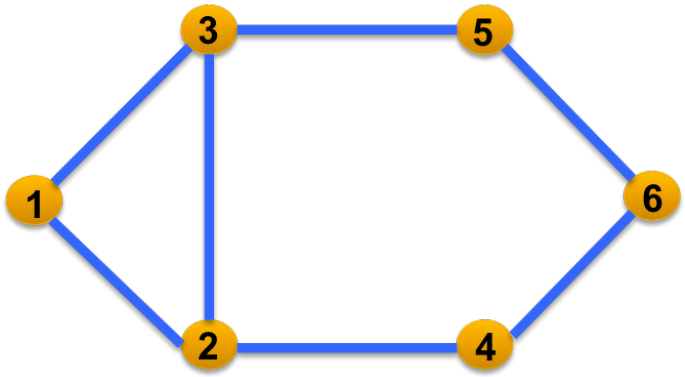

Figure 2.4: An example of a k-clique and k-clubs

2.2.2.2 Global Methods

Rather than finding cliques, global methods such as the G-N algorithm focus on calculating relationship densities [Girvan and Newman, 2002]. The procedure of the G-N algorithm consists of calculating the betweenness of each edge in a given graph. The betweenness is used to measure the importance of a node between any two other nodes. In Figure 2.5, the most important node will be Betty because she is a bridge between Sean's cluster and George. Each member in Sean's cluster who wants to

communicate with George needs to pass through Betty. Therefore, the betweenness of Betty will be the highest of the nodes in Figure 2.5.

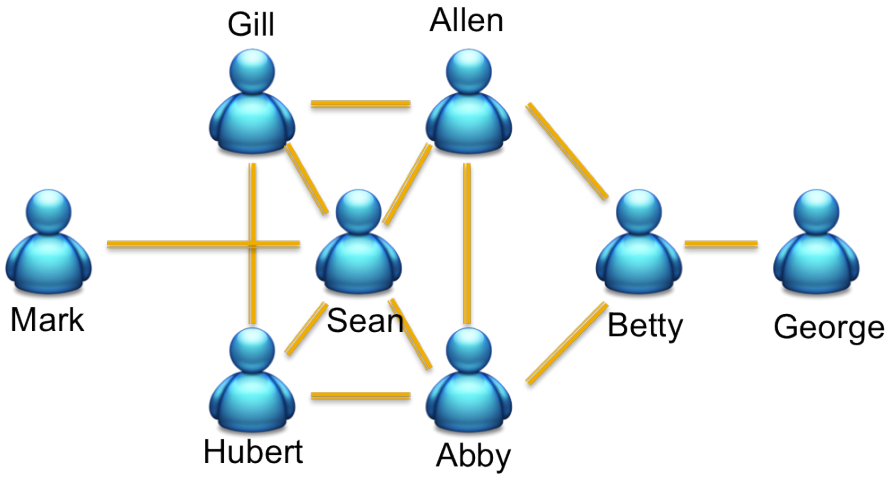

Figure 2.5: A example of betweenness

The G-N algorithm will purge the edge with the largest betweenness each time until the graph is partitioned into several subgraphs. It is clear that when applying the G-N algorithm, the betweenness calculations are performed repeatedly, which results in long computational times. The time complexity of the G-N algorithm is $O(N^E)$, where N is the number of nodes and E is the number of edges. The G-N algorithm is not suitable for large-scale networks.

[Wu and Huberman, 2004] considered the social network as an electrical circuit, with the edge of the graph containing different levels of resistances. The main concept of this approach is that human relationships fade outward from layers of friends, which is identical to voltage becoming increasingly smaller as it flows through a power circuit. The procedure for this approach consists of selecting a node and assigning it a value of 1 volt. Then, 0 volts are assigned to randomly selected nodes from the network. The voltages of all nodes are calculated by applying Kirchoff's laws. The value of each node should between 0 and 1. Subsequently, a threshold is provided, and nodes that have a value higher than the threshold are assigned to one cluster, and the remaining nodes are assigned to another cluster. This approach can also be extended to find multiple clusters by providing a threshold range. The computational

complexity of this method is $O(N+E)$, where N is the number of nodes and E is the number of edges.

2.2.2.3 Score-based Methods

Algorithms such as PageRank [Page et al., 1999], Hyperlink-Induced Topic Search (HITS) [Kleinberg, 1999] and betweenness centrality[Krebs, 2002] are also used by researchers. The concept of these algorithms is to compute an authority score based on the relationships of nodes in the network. Betweenness centrality was used by [Krebs, 2002] to identify terrorist groups. Additionally, [Qin et al., 2005] applied the PageRank algorithm in their search for the global Salafi Jihad network. However, the dataset that they used was relatively small, and their target result had clearly predefined attributes.

2.2.3 Finding by Text

Some researchers have focused their attention on text-mining techniques to find communities in social networks.

[Zhang et al., 2008b] and [Shen et al., 2006] hypothesised that if two individuals share interests, they are likely to be *hidden* friends or in the same community. Based on this assumption, they collected blog posts from the Internet and applied latent semantic indexing (LSI) [Deerwester et al., 1990] to each post. Using keyword filters, the interests of users will be generated. Finally, the users that share interests will be considered to be in the same community.

Using techniques such as LSI helps researchers to observe the features of blogs. Occasionally, the discovered features are dangerous to society, such as if they reveal that the owner of the blog tends to harbour racial sentiments or a hatred of democracy or a particular country. However, the actual level of danger of a blog owner cannot be determined by his posts alone. [Chau and Xu, 2007] hypothesised that if a blog owner who has such dangerous features within his posts also recommends blogs or websites with dangerous features or adds many blog owners who also have dangerous features as friends, he is highly likely to be a member of a criminal group.

2.3 Social Network Analysis Tools

Many open-source and commercial solutions are available for analysing social networks. The majority of these tools provide rich user interfaces and are designed for ease of use. Generally, they are divided into two categories, namely, programming libraries and applications, which are outlined below.

2.3.1 SNA Programming Library

igraph[6] is a widely used C library designed for handling large-scale networks. The built-in functions offer many well-known SNA methods. This library also provides a high-level interface for use with popular scripting languages and applications, such as Ruby, Python and R, which is a feature that is particularly helpful for analysing data programmatically. The Java Universal Network/Graph framework (JUNG[7]) is an open-source SNA library implemented in Java. Similar to igraph, it offers most of the existing SNA methods. JUNG has been shown to be capable to analysing a network with one million nodes. Moreover, it supports various layout algorithms, allowing the user to choose the best one to visualise the network. This perfectly demonstrates how a user-friendly platform can be useful if it provides easy replaceable components. Hence, the proposed framework in Chapter 3 will have similar functionalities.

2.3.2 SNA Application

The SNA Package[8] is a popular library for the well-known statistics software R[9]. The software implements numerous SNA functions such as degree centrality, closeness centrality, betweenness centrality and stress-cent centrality. Using the SNA package can help provide an overview of the given data. However, its major drawback is that you must be familiar with R, which makes the average user unable to easily manipulate the program.

[6] http://igraph.org/redirect.html
[7] http://jung.sourceforge.net/
[8] http://cran.r-project.org/web/packages/sna/index.html
[9] http://www.r-project.org/

CFinder[10] is an open-source tool based on the clique percolation method [Palla et al., 2005], which is written in Java. CFinder allows the user to easily customise the visualisation of overlapping communities and manipulate data. Furthermore, CFinder provides a command line version that provides greater flexibility and enables integration with other systems. However, the core algorithm is hard coded, which limits its extensibility. So, we will try to solve this issue in this thesis.

Cytoscape [Smoot et al., 2011] is another open-source tool written in Java. This tool has an open API, which allows developers to extend its functionality by implementing plugins. Currently, Cytoscape has more than 50 SNA plugins for analysing data. Additionally, Cytoscape has various visual styles, which helps the user generate a clear and visually pleasing network visualisation. We tested Cytoscape with a large dataset and found that its performance is poor. It is also impossible to visualise a large-scale network using Cytoscape.

2.4 Community Mining

If the user can only identify clusters by simple edge connections, usage of the result will be limited. The user can only identify which nodes are closely related by edges of the graph but not be able to group them by context. In this section, we discuss community mining based on social network analysis approaches and other interesting approaches to provide an overview of community mining.

2.4.1 Social Network Analysis Approaches

According to [Kunegis et al., 2009], "Social network analysis studies social networks by means of analyzing structural relationships between people". Mining communities that use traditional social network analysis approaches generally focus on the structure of the social network, which is represented by a direct or indirect graph. Each node in the graph represents an instance in the network, e.g., a person or object, whereas links between nodes represent relations between the instances. The re-

[10] http://www.cfinder.org/

lations between the instances in the network can be defined by explicit information such as friends on Facebook or followers on Twitter.

[Schwartz and Wood, 1993] used the structure of a subgraph in the network to identify groups of people that share interests based on email history. [Kunegis et al., 2009] analysed the structure of a network to identify communities among Slashdot[11] users. The method of [Kunegis et al., 2009] is based on social network analysis approaches with negative-weighted edge graphs. [Yang et al., 2007] used an algorithm based on graph theory to identify signed social networks. [Flake et al., 2000] stated that a minimum cut framework can be used to efficiently identify members in a community. [Newman, 2004, Kumar et al., 1999] detected communities by considering the network structure. [Newman, 2004] took network structure properties such as loops and edges of the network into account. [Kumar et al., 1999] considered bipartite subgraphs to locate communities of websites. Several studies [Gibson et al., 1998, Larson, 1996] used link analysis to identify web communities based on the concept that pages that are on the same topics tend to connect with each other via hyperlinks. For example, a web page on technology often links to other pages focusing on technology.

However, discovering communities that are formed by hidden relations between instances (e.g., people share interests in social network analysis approaches) is limited because such approaches primarily model only explicit relations of the instances in the network [Shen et al., 2006].

2.4.2 Other Approaches

The use of traditional social network analysis approaches is challenged by the limitation of identifying implicit relationships between instances in the network because social network analysis approaches are not able to capture the rich semantics that may exist in relationships or links in a social network. To overcome this problem, [Adamic and Adar, 2001] mined users' communities by not only analysing link structures (inlink and outlink) of users' webpages and mailing lists but also by taking the content of the webpages into account. [Shen et al., 2006] used other

[11] http://slashdot.org/

approaches to find communities of latent friends, i.e., people who share similar interests based on the topics of their blog posts.

They proposed three new methods that are based on the concept of topic modelling and document clustering. The first method is the cosine similarity-based method. This method attempts to find the similarity between people using the cosine similarity between their blog contents. The second method is the topic-based method, which calculates the similarity between topics derived from topic modelling using latent dirichlet allocation (LDA). Finally, the third method is a combination of the first two methods. [Gao et al., 2012] also attempted to solve this problem. They identified the interests of groups of people based on their relationship and content information using a probabilistic factor model to discover their communities. First, they constructed matrices of user interests. Then, they used the probabilistic factor model to identify latent communities based on the matrices of interests.

Similar to [Gao et al., 2012, Yang et al., 2007], our goal is to identify communities of tweets in Twitter, which is more closely related to the document clustering problem rather than the social network analysis problem. Therefore, in later sections, we discuss document clustering, particularly in the case of short text documents.

2.4.3 Bag-of-Words Model

The bag-of-words model is one of the most popular techniques used in text classification and clustering. This model ignores the ordering of terms in a document and focuses only on the number of occurrences of each term. In the bag-of-words model, each document is represented in terms of a $|D|$–dimensional vector *doc*:

$$doc = (term_1 : weight_1, term_2 : weight_2, \ldots, term_n : weight_n)$$

where D is the total number of documents in the corpus and the dimension corresponding to word $term_i$ has a value $weight_i$, which is known as the term weight. There are several methods for computing the weighting values of these terms. In this thesis, we introduce the two most common techniques for calculating the term weighting: 1) term frequency

(TF) weighting and 2) term frequency-inverse document frequency (TF-IDF) weighting.

$$tf_{i,j} = \frac{N_{i,j}}{\sum_k N_{k,j}} \tag{2.1}$$

where $N_{i,j}$ is the total number of terms N_i in the document j and $\sum_k N_{k,j}$ represents the sum of terms in the document j.

$$idf_i = \log \frac{D}{df_i} \tag{2.2}$$

where df_i is the total number of terms n_i in the corpus. Multiplying $df_{i,j}$ and idf_i allows us to filter out most of the common terms and retrieve the important terms, which will have higher TF-IDF weights.

$$tf - idf = tf_{i,j} \times idf_i \tag{2.3}$$

2.4.4 Similarity Measure

Every clustering method needs to calculate the similarity between documents to group similar documents into the same cluster. Therefore, the similarity measure is one of the major factors that affects the performance of clustering methods. In this section, we discuss cosine similarity, which is one of the most popular similarity measures for text documents.

Cosine Similarity is a similarity measure based on the vector space model that measures the difference between two vectors based on the cosine of the angle between them.

$$sim_{cosine}(A, B) = \frac{A \cdot B}{\|A\| \, \|B\|} \tag{2.4}$$

However, in the clustering algorithm, we generally use the distance measure, which is a dissimilarity measure. For this purpose, we can derive the distance measure by subtracting the similarity measure from 1 as follows:

$$dis_{cosine}(A, B) = 1 - sim_{cosine} = 1 - \frac{A \cdot B}{\|A\| \, \|B\|} \tag{2.5}$$

[Rangrej et al., 2011] evaluated the performance of cosine similarity in a short text clustering task in a practical setting. They examine the similarity measure between two clustering algorithms: k-means and affinity propagation. Cosine similarity yielded a 10.25% cluster error with k-means clustering and performed better, as demonstrated by an affinity propagation (AP) of 2.95%.

2.4.5 Identification and Filtering of Wikipedia Topics

Wikipedia is currently used in many fields related to machine learning, such as natural language processing, text classification and text clustering. One of the difficulties in using Wikipedia is accurately matching between the input text and Wikipedia topics (articles) because each word or each phase in the input text can refer to one or more Wikipedia topics. For example, the word "apple" can refer to both "apple (fruit)" and "Apple Inc.".

Therefore, selecting the most appropriate topics is an important problem. This problem is classified as a word sense disambiguation problem. A word sense disambiguation problem is a problem related to selecting the most appropriate meaning of each word in a document. This problem has been of interest for many years, and numerous researchers have proposed various methods for solving this problem. A review of word sense disambiguation methods can be found in [Navigli, 2012].

An alternative approach that performs well for word sense disambiguation is using machine learning methods to learn a labelled training set and classify ambiguous words. This topic is used in text annotation with Wikipedia links. [Cucerzan, 2007, Ferragina and Scaiella, 2010, Mihalcea and Csomai, 2007, Milne and Witten, 2008b] are several studies on text annotation with Wikipedia links. [Mihalcea and Csomai, 2007] was the first paper that discussed Wikipedia as a resource for annotation and was followed by a significant improvement in this field by [Milne and Witten, 2008b].

However, most studies performed experiments in the context of standard-length documents. These experiments did not ensure that the approaches used in the papers would perform well using short text documents such as tweets, news or search snippets. In 2010, [Ferragina and Scaiella, 2010] brought the concept of annotating plain-text with Wikipedia links to the context of short-length documents. They used anchors as an identification resource rather than only the Wikipedia title, as in [Wang et al., 2009], because anchors are appropriately selected by the people who create the pages. Their approach consists of three main steps: anchor parsing, anchor disambiguation and anchor pruning. Performing these steps

provides the ability to address short text for the annotation system. The overall performance of [Ferragina and Scaiella, 2010]'s system was significantly greater compared to that of [Milne and Witten, 2008b]'s system for both short and long text document cases.

2.5 Inclination (Opinion) Mining

The natural language processing (NLP) field includes a branch of research that focuses on inclination mining, which uses NLP techniques such as text analytics to observe and retrieve subjective information from data of interest. The overall goal of inclination mining is to identify the sentiment of a user towards various topics. In this thesis, it is also very important for the proposed system to detect the options expressed in each document so it can help the user to identify the group leaders, which will be useful for real-world cases. In the following subsection, we will investigate works related to inclination mining.

2.5.1 Sentiment Dictionary Approach

This type of inclination analysis method only considers the lexicon of the document. Prior knowledge, such as a dictionary of sentiment terms, is required for this approach to label the sentiment of given documents. [Turney, 2002] proposed a method that utilises semantic orientations (SOs) to label the sentiment of terms extracted from on-line user feedback. It was found that the occurrence rate of negative semantic inclination and negative reference opinion terms is considerably higher than that of positive reference opinion terms. Namely, negative semantic orientation terms are closer to negative reference opinion terms than positive terms. In light of this characteristic, [Turney, 2002] derived a semantic orientation (SO) mining approach based on pointwise mutual information (PMI), which is commonly used to measure the degree of correlation between two terms, as shown in Equation 2.6, for sentiment labelling.

$$PMI(term_1, term_2) = \log_2 \left(\frac{P(term_1 \wedge term_2)}{P(term_1)P(term_2)} \right) \tag{2.6}$$

where $P(term_1 \wedge term_2)$ is the probability that $term_1$ and $term_2$ occur simultaneously and $P(term_1)$ and $P(term_1)$ are the probabilities of $term_1$ and $term_2$, respectively. Then, we can calculate the SO value of the given term $term_{gt}$ using Equation 2.7

$$SO(term_{gt}) = PMI(term_{gt}, \text{“}excellent\text{”}) - PMI(term_{gt}, \text{“}poor\text{”}) \quad (2.7)$$

where “excellent” represents the positive terms and “poor” represents the negative terms.

The objective is to estimate the positive and negative semantic orientation for each term from a given document. Summing the SO values of every term in the given document will reveal the opinion polarity of the document.

A sentiment term dictionary that includes positive and negative terms was used in [O'Connor et al., 2010]. In their study, a method was proposed to analyse a Twitter dataset. A relative sentiment detector (RSD) was defined in this method, as shown in Equation 2.8. RSD considers the ratio of positive and negative terms within a tweet. If the RSD is greater than 1, then it will label the tweet as a positive sentiment, and if the RSD is less than 1, it will label the tweet as a negative sentiment.

$$RSD = \frac{count(positive\ terms)}{count(negative\ terms)} \quad (2.8)$$

Rather than analysing the tweet sentiment from its own context, [Chamlertwat et al., 2012] utilised SentiWordNet[12] as an external sentiment dictionary to label the tweet. Each term in SentiWordNet has a pair of scores, which are assigned to a positive score and a negative score. Then, the algorithm sums all the scores to determine the tweet's sentiment.

2.5.2 Machine-learning-based Approach

Certain researchers have different perspectives on sentiment analysis. They consider this problem as a classification problem and attempt to apply the classic machine learning method to overcome it. [Pang et al.,

[12]http://sentiwordnet.isti.cnr.it/

2002] is one of the earliest studies that attempted to classify the sentiment of documents by applying different types of supervised machine learning algorithms to the documents. The algorithm classified documents into two groups: positive and negative. Studies such as [Agarwal et al., 2011] and [Pak and Paroubek, 2010] followed [Kim and Hovy, 2004] and [Pang et al., 2002]'s steps; they applied similar approaches to the Twitter platform and attempted to classify tweets based on sentiments. [Davidov et al., 2010] attempted to improve the classification results by using 15 emojis and 50 hashtags as external supporting sources.

As mentioned in Chapter 1, a tweet is a special type of document that is subject to a content length limitation. Unlike regular articles or blog posts, a tweet only contains very limited information, which easily misleads the classifier.

People everywhere love Windows & Vista. Bill Gates

Windows 7 is much better than Vista!

If we attempt to classify the two tweets listed above, "Bill Gates" might be misunderstood as having positive sentiment. In addition, "Windows 7" and "Vista" may also be considered as positive. To overcome this problem, [Jiang et al., 2011] introduced various syntax rules to the classifier and improved the generated result.

2.6 Opinion Mining Tools

The inclination mining technique can be applied to many business areas. One possible application is mining review articles on specific products such as movies, tablets and bikes. In general, the goal of these applications is to extract information from the user's content and identify the sentiment of the extracted information. [Liu et al., 2005] proposed a method that extracts product features using a supervised rule discovery. First, the selected important "part-of-speech" is provided for each term. Then, the features of the product are manually selected after the tagging process. A label *feature* will be labelled to the product features. The association mining system CBA (classification based on association [Ma,

1998]) will be used to observe the rules of *feature* occurrences. Then, the rules that are generated from previous observations are used to identify the feature that the user may mean in their texts.

In recent years, many projects have attempted to mine opinions from Twitter, which is the largest micro-blogging platform in the world. Twitter provides researchers real-world user data and can be accessed in real time. For instance, [Chamlertwat et al., 2012] extracted phone features, such as screen size, chipset power and ram size, from collected smartphone-related tweets. [Golder and Macy, 2011] found that the mood of users is affected by the biological clock across different countries and cultures from Twitter. Furthermore, in the studies of [Bollen et al., 2011] and [Tumasjan et al., 2010], the extracted general public opinion from Twitter could be used to predict the results of stock markets and elections.

2.7 Identification of Opinion Leaders in On-line Social Media

Researchers and people in the marketing area all over the world are using surveys as reliable sources of data that provide a good understanding of their area of interest. However, the downsides of taking surveys are quite obvious: surveys need to have a certain number of participants to make them statistically meaningful and also require significant manpower to process and analyse the data to generate credible results.

It used to take a lot of effort, in both time and manpower, to collect and analyse the communications between users. With the growth of on-line social media platforms, people are generating valuable information such as their social connections, identities, habits and interactions. We can utilise modern technologies to crawl and collect millions of user generated data points automatically in a much shorter time. This ability gives researchers a great chance to calculate influence by collecting these data. Additionally, viral marketing (or word of mouth) can drive marketing experts to identify on-line opinion leaders. Hence, in this section, we will discuss studies that utilise different methods to identify on-line

opinion leaders from different social media platforms.

2.7.1 Attributes that interest researchers the most

There are many kinds of user-generated data on a social media platform, and researchers are interested in applying or proposing algorithms to identify on-line leaders by using these attributes. The most commonly seen attributes that have attracted the attention of researchers are:

- Connections between user A and user B
- The relationship that bundles with the connection, e.g., friends and following
- Comments or messages with timeline
- User's information (profile)
- Actions on the comments such as "likes" or "retweets".

Among the listed attributes, researchers frequently choose user profiles, connections and relationships among users to identify opinion leaders. In the other words, they consider opinion leaders as a group of users who have more connections and can attract other users to have on-line interactions with their posts and comments. For example: [Lin et al., 2013] focused on finding opinion leaders on the Sina Weibo platform. The attributes they chose are friends, fans, posts, re-posts (similar to "retweets" on Twitter), timeline and the length of posts. [Li and Du, 2011] conducted research on MySpace using the number of comments, blogs, visits, reviews, properties of authors and viewers to identify the influence of a given topic.

Twitter, as the largest on-line micro-blogging platform, is the most popular choice for researchers such as [Bakshy et al., 2011] [Cha et al., 2010] [Xu et al., 2014] [Zafarani et al., 2014]. It contains fairly simple relationships and interactions compared to other platforms. These attributes include numbers of retweets, mentions, followers and likes. On the other hand, researchers have mainly focused on graph-based relationships between users on Facebook and have tried to identify communities

among them [Shafiq et al., 2013] [Bodendorf and Kaiser, 2009] [Cho et al., 2012]. In general, researchers attempt to identify opinion leaders based on the natural structures of each on-line social platform. Hence, it is understandable that they utilise different methods to approach the problem.

2.7.2 Approaches to identifying on-line opinion leaders

Here, we split approachestoof identifying on-line opinion leaders into two categories. The first are network-based approaches where researchers focus on the connections and relationships among users. Then, different graph theory methods are applied to analyse the network. In contrast, individual-based approach focuses on each user's attributes and the interactions between them.

2.7.2.1 Network-based approach

A on-line social networks are usually considered as networks of relationships between friends. Hence, a social network can be defined as (V, E), where V is a set of user nodes and E is a set of user relationship edges.

Various of parameters and algorithms were used in [Shafiq et al., 2013] to calculate opinion leadership on a user's graph of Facebook. Their method combines the degree of edges, the count of friendship triangles, PageRank, clustering coefficient, shortest path and centrality. Similarly, [Cho et al., 2012] defined a virtual user network that has users as nodes and self-defined *intimacy* relationship between users. Then, the probability of adoption and context propagation were defined by calculating the *intimacy* and the degree of neighbouring nodes of the network. The simulation they conducted is to see how fast the context will be spread and adopted by giving a context to different users and make it spread in the network. The result of their finding is interesting. The people who have higher degrees of connections made the context spread more quickly; the higher total *intimacy* of the users was, the better chance was that they would be influenced (context adoption) by others.

2.7.2.2 Individual-based approach

The concept of this approach is to find on-line opinion leaders based on a set of criteria. For example, how many users are following these leaders, how many interactions they have with his followers, and their expertise level. The comparison can be made by calculating the influence of users based on different criteria mentioned before. Additionally, this comparison can be conducted by combining all criteria together by assigning weights to each criterion based on importance. For the former method, an evaluation function, such as TOPSIS [Hwang et al., 1993] and AHP [Triantaphyllou et al., 1998], will be applied to generate the final scores of each user and rank users by generated scores.

Taking [Cha et al., 2010] as an example. First, they converted each Twitter user's count of retweets, in-degree and mentions into a set. Then, the correlation coefficient between two users were calculated and ranked. The result of their work indicated that mentions and retweets have a significant positive correlation (0.638, among the top ten percent). This result means that a user who was usually mentioned by other users was also usually retweeted and vice versa. It it worth noting that "in-degree" doesn't have strong relation to mentions or retweets, the correlations of which were only 0.286 and 0.122, respectively.

In the later method, [Lin et al., 2013] utilise seven user attributes and classified them into three criteria groups: support, activity and influence. Support group contains user attributes such as *count of forward, comments.*; Influence group contains attributes like the *number of friends, fans and posts*; and *the length of the content and time* were classified in the activity criteria group. Every user's score was averaged by the attributes in each group. A final result for each user is calculated by the AHP function, shown as 2.9:

$$AHP(i) = weight_S \times S(i) + weight_A \times A(i) + weight_I \times I(i) \qquad (2.9)$$

where $weight_S, weight_A and weight_I$ represent the weighting of the support, activity and influence criteria groups, and the weighting of each criteria group was set to 85%, 10%, 5%, respectively. Because [Lin et al., 2013] believed that a user should express his opinion, converse with his

followers actively and provide guidance to his followers as an opinion leader, the weighting of the support criteria group is the greatest.

[Li and Du, 2011] conducted a sophisticated study on identifying opinion leaders on on-line social blog platforms. At the first, they categorized blogs into specific topics by calculating ontological similarity. Then, they labelled all viewers and authors' attributes of each blog, which were mentioned before. Finally, the TOPSIS model was introduced to rank users based on the criteria.

2.8 Summary

Many established approaches for social network analysis, community mining and inclination mining as well as various well-known applications (tools) in this research field are introduced in this chapter. However, these approaches have yet to fill some of the research gaps that were addressed in Chapter 1. This thesis contributes to the aforementioned fields by providing a three-layered platform that combines several technologies. The proposed platform will be described in the next chapter.

Chapter 3

Proposed Framework

Chapter 2 discussed related work in the fields of social network analysis, topic group mining and sentiment analysis. The gaps that on-going research have yet to overcome were addressed. This chapter presents a novel three-layered solution that integrates several machine learning techniques with a flexible platform design. The design principle of the platform is that each layer consists of several key components that are not only responsible for a specific task but also loosely integrated with the other components. The major benefit of this architecture is that each component can be improved or even replaced by similar function modules without modifying the other components. This change is achieved by designing a generic interface and unified document schema for each component. Hence, the adaptability of the platform is significantly higher than that of other proposed systems.

The main concept of the proposed framework is that each component can be replaced easily.

Each component of each layer is briefly described in this chapter to provide an overview of the proposed solution. The details of the major components, such as the **Topic Classifier**, will be presented in Chapter 4, and the **Polarity Classifier, Social Interaction Analysis and Topic Group Inclination Analysis** will be discussed in Chapter 5. In Chapter 6, we will demonstrate the overall performance of the integrated system in solving the described research problems. The architecture diagram for this framework is shown in Figure 3.1. The components of each layer and their functions are presented in the following.

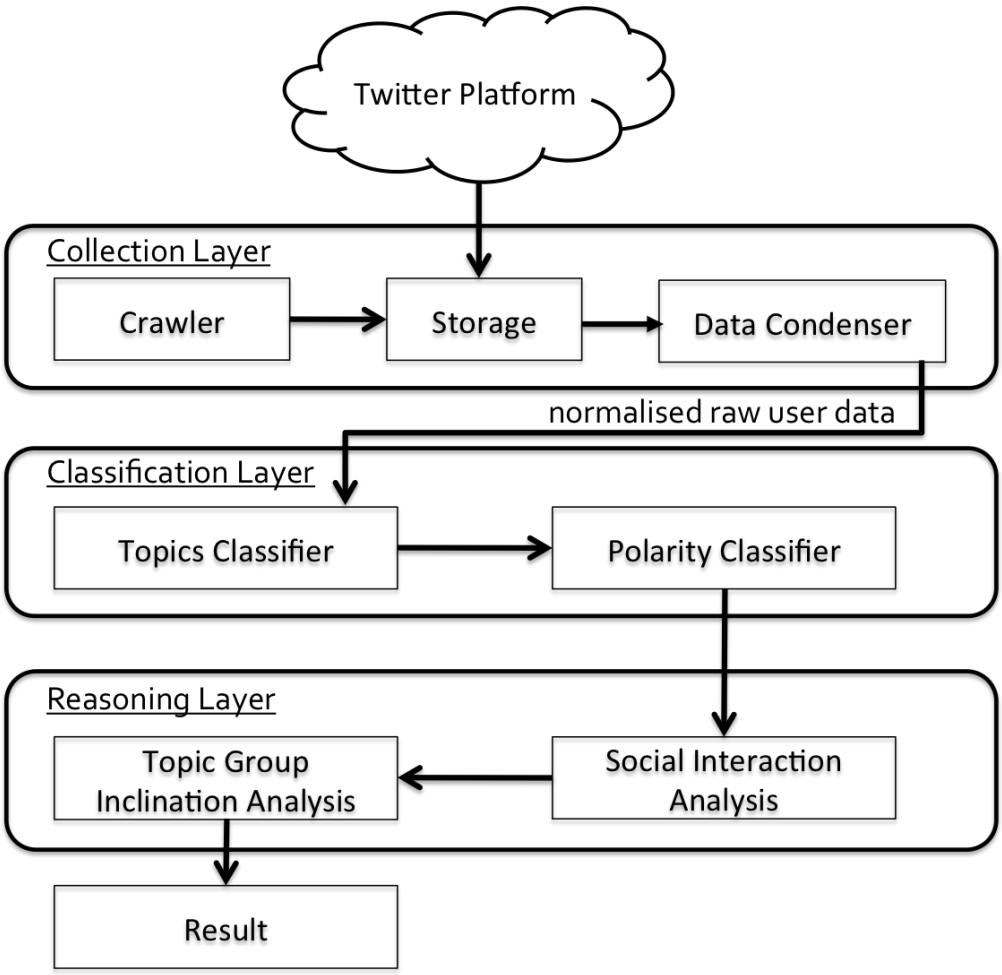

Figure 3.1: Overview of the proposed three-layered framework for user inclination analysis. The framework provides three layers of abstraction through the collection, classification and reasoning layers

3.1 Collection Layer

The collection layer contains components that retrieve the data from the micro-blogging system (Twitter in this thesis), process the raw data into a pre-defined format and convert the data into numerical parameters.

3.1.1 Crawler

The crawler is responsible for retrieving the user data from the Twitter platform. All retrieved tweets will be placed in storage for future use.

The crawler is designed to be a lightweight but robust daemon such that it can be easily deployed on multiple machines to increase throughput.

3.1.1.1 Crawler Interface

Although this research employs Twitter as the only data source provider, it can be easily changed from Twitter to other data source provider. The new data source crawler only needs to be implemented by the following crawler interface (Python pseudo-code).

```
class BaseCrawler(object):
  def _init(source_name, base_url, retries, proxies, timeout);
  def _start_task(uri, wait_time, **kwargs)
  def _retry(**kwargs)
  def _abort(**kwargs)
```

Initialize

When a custom cralwer starts, it will call **BaseCrawler**'s initialization function to construct a new crawler object. A set of parameters will be passed to the crawler object to define what the crawler is and assign basic settings to it. The definition of each parameters is listed here:

- ***source_name***: Specify the name of data source, e.g., Facebook or Plurk.
- ***base_url***: Specify the api base url of data source provider.
- ***reties***: Specify how many times should the cralwer retry if the connection is disconnected or timeout.
- ***proxies***: Specify a list of proxies if needed. This is especially useful if the data source has anti-crawling mechanism. By switching different proxies, it can provide certian level of remediation of being blocked by the data souce provider.
- ***timeout***: Specify connection timeout to the crawler in seconds. When the timeout exceeds, it will trigger *_retry* function.

Start the task

This is the main processing component in the crawler; it is in charge of fetching data from the Internet and handling errors by retring certain number of iterations or aborting. The user must specify the following parameters while implementing his crawler:

- ***uri***: Specify URI of the data feed. The crawler will combine it with *base_url*. For example, if base_url: **https://api.twitter.com/1.1** and URI: **/statuses/user_timeline.json** is given, the crawler will try to fetch **https://api.twitter.com/1.1/statuses/user_timeline.json**
- ***wait_time***: Specify the waiting time for completing data retrival from the data soruce provider. This is differrent from **timeout** since sometimes the connection is alive but the data source server is overloaded so it cannot return the data within a reseanable time.
- ****kwargs***: This is an optional parameter but providing flexibilities for different data crawler implementations. The develop can provide data source specific parameters for further usage since we cannot predefine all necessary parameters that cover all providers.

Once the data are successfully retrieved from the Internet, the developer must transform the raw data into platform specific document schema, which will be subsequently described.

Retry and abort

retry function is responsible for retrying to estiblish the network connection (network timeout) or fetch the data (data timeout) when the error occours. If the number of retries has exceeded, the crawler will consider this task is inoperable and call *abort* function.

Whenever the *abort* function is called, the task is marked as failed. The developer can perform self-defined clean-up procedures in this function, such as removing all meta-data from the disk or sending alerts to administrators.

3.1.2 Storage

To support fast lookup and a flexible schema to adapt to the rapid changes of different platforms, the proposed system will take advantage of a distributed key-value database system, MongoDB[1], which allows us to change the table schema without altering the entire table, thereby avoiding database locking issues and improving the system availability.

Scalability is another concern for any system that manages a considerable amount of data. A distributed database system provides a simple procedure for adding new nodes to the system to accommodate rapidly growing data.

The component only transforms the fetched data based on our document schema; it does not perform any data cleansing during the process. Users can conduct as many trials as they wish since this design ensures that the stored data in each trial are original data.

Storage data schema

Since the platform allows the crawler component to be implemented and replaced by the developer, it is crucial that the retrieved data is transformed into same document schema so the underlaying components can parse the stored data without knowing changes in crawler componenet. The document schema is defined as the following json format:

```
{
   "source": <string>,
   "crawler_id": <string>,
   "fetch_time": <unix timestamp>,
   "doc_id": <string>,
   "doc_time": <unix timestamp>,
   "doc_url": <string>,
   "doc_author": <string>,
   "text": <string>,
   "source_extra": <string>
}
```

The definition of each field is listed as follows:

[1] http://www.mongodb.org/

- ***source***: Specify a name or unique id to data source providers.
- ***crawler_id***: Specify crawler information. This is for debug tracing and the format should be "CRAWLER_IP/Process_ID".
- ***fetch_time***: Specify the actual fetching time in unix timestamp format.
- ***doc_id***: Specify an unique ID to the document.
- ***doc_time***: Specify the actual document creating time in unix timestamp format. For example, the posting time of the specific tweet.
- ***doc_url***: Specify the direct link to the document.
- ***doc_author***: Specify the author of the document.
- ***text***: Specify the raw content of the document.
- ***source_extra***: Specify data source related information. For example, lists of "retweet" or "likes" userid and document id

A sample of json stored in data storage is shown as follows:

```
{
    "source": "twitter",
    "crawler_id": "127.0.0.1/337",
    "fetch_time": 1341497601,
    "doc_id": "4d1473e93068f2aae63db10389534f32",
    "doc_time": 1341252467,
    "doc_author": "igrigorik",
    "doc_url":
        "https://twitter.com/igrigorik/status/219854736160071682",
    "text": "It's Time to Fix HTTPS: http://bit.ly/LLgkul -
        great presentation, nodding all along.",
    "source_extra": ""
}
```

The user relationship is very important for the social interaction analysis and, hence, needs to be stored by a uniform document schema, which is listed as follows:

```
{
    "source": <string>,
    "user": <string>,
    "location": <string>,
    "relationship": {
        "following": <list>,
        "followed_by": <list>
    },
    "extra_info": <string>
}
```

The definition of each field is listed as follows:

- ***source***: A name or unique id to data source providers.
- ***user***: User name on the provider platform
- ***location***: Location of the user (if available)
- ***relationship***: Store the user's relationship. It can be extended based different data source providers.
- ***extra_info***: Additional infomation provided by data source.

A sample json of user relationships on Twitter is described as follows:

```
{
    "source": "twitter",
    "user": "sample1",
    "location": "London, UK",
    "relationship": {
        "following": ["user1", "user2"],
        "followed_by": ["user99"]
    },
    "extra_info": "sample user on Twitter"
}
```

3.1.3 Data Condenser

The data condenser reads the raw data from the database. The raw data contain noise such as slang words, emoticons or random characters, such as Lemme (an abbreviation of "let me"), :D and "!@#*". It is the data condenser's responsibility to remove these noises. The data condenser is also responsible for converting selected fields of tweets, such as stemmed terms (which reduce inflected words to their root form, such as "jumps" to "jump") and normalising selected fields of tweets into numerical parameters for the classification layer.

3.2 Classification Layer

The classification layer is the major interface that processes the formatted data from the collection layer. The layer generates a set of classified topic groups for the reasoning layer, which allows users to review their desired results. Details of this layer will be discussed in Chapters 4 and 5.

3.2.1 Topic Classifier

The topic classifier has an important role in the proposed framework. First, the framework needs to have the ability to retrieve the topic(s) from each document. An external knowledge base will be provided to increase the accuracy of extraction. Second, a classification algorithm will be investigated and applied to the data with their extracted topic(s). A set of classified topic groups will be generated for further processing.

3.2.1.1 Topic Classifier Interface

Although this research employs bisecting K-means as the classification algorithm, it can be easily changed to other unsupervised algorithms such as dbscan. The newly applied algorithm needs to be implemented by the following topic classifier interface, which ensures that the unified documents are correctly fed and processed.

```
class BaseTopicClassifier(object):
   def _init(algo_name, **kwargs);
```

```
def _fit(data)
def _fit_predict(data)
def _write_topic_output(object, **kwargs)
```

Initialize

When the custom topic classifier is constructed, the initial function of **BaseTopicClassifier** will be called. Basic settings will be set to the object for topic classification tasks. The definition of each parameters are listed as follows:

- ***algo_name***: Specify the name of unsupervised algorithm.
- ****kwargs***: It allows the developer to give a set of parameters that related to the algorithm.

Fit and Predict

These two functions are the core of custom implemented topic classification algorithm component. The developer should implement the algorithm in *_fit* function and put prediction related code in the *_fit_predict* function. The only allowed parameter is *data*, which typically is a N-demission NumPy array in Python implementation.

- ***_fit***: Perform user designed unsupervised classification algorithm.
- ***_fit_predict***: Perform user-designed unsupervised algorithm and predict cluster index for each sample.

Write the result

It is essential to apply a constant output format for the data pipeline. This consistency is even more important when the component of the data pipeline is allowed to be replaced by users. The malformed or corrupted output file will prevent the following components from recognising the content and may cause the entire process to stop unexpectedly. Hence, the document schema of the topic classifier is defined as follows:

```
{
    "doc_id": <string>,
    "doc_time": <unix timestamp>,
    "doc_topic_id": <string>,
    "doc_author": <string>,
    "text": <string>,
    "topic_classifier_extra": <string>
}
```

The definition of each field is listed as follows:

- ***doc_id***: The unique ID for the document. If the other component needs raw information, it can use this ID to locate the document in storage.
- ***doc_time***: Document original creating time in unix timestamp format.
- ***doc_topic_id***: The cluster id assigned by topic classification algorithm
- ***doc_author***: Author of the document.
- ***text***: The content of the processed document.
- ***topic_classifie_extra***: Specify topic classification algorithm related information. For example, the cluster ID where the document belongs to.

A sample json output of the topic classifier is shown as follows:

```
{
    "doc_id": "4d1473e93068f2aae63db10389534f32",
    "doc_time": 1341252467,
    "doc_topic_id": "cluster_1",
    "doc_author":
        "https://twitter.com/igrigorik/status/219854736160071682",
    "text": "It's Time to Fix HTTPS: http://bit.ly/LLgkul -
        great presentation, nodding all along.",
    "topic_classifier_extra": "cluster_size:34"
```

```
}
```

3.2.2 Polarity Classifier

The polarity classifier is fed by the topic classifier. A sentiment analysis and clustering algorithm will be investigated to extract the polarity of documents and classify them into different categories. A well-established sentiment lexicon will be utilised to improve the accuracy.

3.2.2.1 Polarity Classifier Interface

The user might want to conduct different experiments on various kind of polarity classifiers. To achieve this goal, the polarity classifier needs to provide a general interface for the developer to implement. The base object provides a basic but overwriteable *read_topic_output* function, which ensures that the input file is valid with the upper component but also leaves the flexiabilities to the developer.

```
class BasePolarityClassifier(object):
    def _init(algo_name, **kwargs);
    def _fit(data)
    def _predict(data)
    def _write_polarity_output(object, **kwargs)
```

Initialize

When a custom polarity classifier is constructed, the initial function of **BasePolarityClassifier** will be called. Basic settings will be set to the object for polarity classification tasks. The definition of each parameter is listed as follows:

- ***algo_name***: Specify the name of polarity classification algorithm.
- *****kwargs***: It allows the developer to give a set of parameters that related to the algorithm.

Fit and Predict

The concept is the same as the previously mentioned **BaseTopicClassifier**. The developer should implement the custom polarity classifier in the *_fit* function and put prediction related code in *_predict* function.

- ***_fit***: Build a polarity classifier from the training set.
- ***_predict***: Predict polarity class for input data.

Write the result

Since the results of the polarity classifier will be employed by components in the reasoning layer, the output file format should be defined so it can be utilised by custom implementations.

```
{
    "doc_id": <string>,
    "doc_time": <Unix timestamp>,
    "doc_topic_class_id": <string>,
    "doc_polarity": <string>,
    "doc_author": <string>,
    "text": <string>,
    "topic_classifier_extra": <string>,
    "polarity_classifier_extra": <string>
}
```

The definition of each field is listed as follows:

- ***doc_id***: The unique ID to the document.
- ***doc_time***: Document original creating time in Unix timestamp format.
- ***doc_topic_cluster_id***: The cluster id assigned by topic classification algorithm.
- ***doc_polarity***: The polarity assigned by polarity classifier.
- ***doc_author***: Author of the document.
- ***text***: The content of the processed document.

- ***topic_classifie_extra***: Topic classification algorithm related information.
- ***polarity_classifier_extra***: Specify polarity classification related information.

3.3 Reasoning Layer

The reasoning layer consists of a set of components that will infer the outcome of the classification layer based on the user's on-line social interaction. Two major components are described as follows:

3.3.1 Social Interaction Analysis

Most on-line platforms have unique implicitly or explicitly defined relationships amongst users. For instance, Facebook has an explicit "friend" relationship, which means that both users agree that they know each other to some extent. Conversely, Twitter only has implicit user relationships: follows and following. It cannot be inferred that two users are friends if one user follows another user's Twitter. This component will construct a social interaction graph based on these relationships for subsequent analysis.

3.3.1.1 Social Interaction Graph Schema

The proposed social interaction analysis method needs two types of relationship: 1) Inter-user relationship and 2) User-Document relationship. Fortunately, almost all on-line social network platforms have these relationships. It means that the social interaction graph required by this platform can be generated by different data source providers as long as the it follows predefined schema. The graph schema is defined as follow:

```
{
    "inter-user-rel": [
        {
            "from_user": <string>,
            "to_user": <string>,
            "action": <string>
```

```
        }
    ],
    "user-doc-rel": [
        {
            "from_user": <string>,
            "to_doc": <string>,
            "action": <string>,
            "action_time": <Unix timestamp>
        }
    ]
}
```

The graph schema consists of two sections: inter-user relationship and user-document relationship. First, the meaning of the fields in the inter-user relationship section is described as follows:

- ***from_user***: The ID of the user who initial a social action.
- ***to_user***: The ID of the social action recipient from *from_user*
- ***action***: Social action such as "following"

The description of the user-document relationship is presented as follows:

- ***from_user***: The ID of the user who perform an action on specific document.
- ***to_doc***: The document ID where *from_user* performs action on.
- ***action***: Document actions such as "retweet", "reply" or "likes"
- ***action_time***: When the document action was performed.

3.3.2 Topic Group Inclination Analysis

This component will be the final stage of the framework. The user inclination, which combines use of the social interaction graph in each topic group, will be analysed. The final results will provide a set of topic groups, in which each user in the group is labelled with their inclination towards the topic group.

3.3.2.1 Topic Group Inclination Analysis Result Schema

The main reason that we need a unified format is that the final stage result will be presented to the platform's users, for instance, via the web frontend. Hence, the output of the final result should have a schema so the frontend does not need to be aware of changes in the backend components. The final result schema is listed as follows:

```
{
    "group_id": <string>,
    "group_topic": <string>,
    "inclination": {
        <polarity>: [
            {
                "user_id": <string>,
                "is_leader": <true|false>,
                "doc_ids": [<string>]
            }
        ]
    }
}
```

The topic group inclination analysis result schema is defined as follows:

- ***group_id***: The ID of the topic group.
- ***group_topic***: The predicted topic of the group.
- ***inclination***: A dictionary that stores a set of user polarities.

  polarity: Depending on how the analyser is designed. For example, it can be "positive", "negative" or "unknown"

  user_id: ID of the user.

  is_leader: Is the user classified as a group leader by algorithm.

  doc_ids: A list of user's document IDs which are related to this topic.

- ***action_time***: When the document action was performed.

A sample output is given as follow:

```
{
    "group_id": 71,
    "group_topic": "Detroit Tigers",
    "inclination": {
        "positive": [
            {
                "user_id": "_Pesch",
                "is_leader": false,
                "doc_ids": ["be8d3c75774edcf67c08be987145686f",
                    "7011f727df5dd05945541befff6f9b7e"]
            }
        ]
    }
}
```

3.4 Summary

The process flow can be summarised as follows. The crawler retrieves the raw data from the micro-blogging system and stores it in a database. Subsequently, the data condenser fetches the raw data from the database. The data condenser removes all noise from the raw data and transforms the cleaned data into a specific format. The topic classifier applies the information retrieval and machine learning classification method to extract the topic(s) and classify the topic groups, which are fed to the polarity classifier to determine the document's inclination towards the topic group to which it belongs. The reasoning layer will analyse user interactions within each topic group. The final result will be generated to help the user understand the user inclination status in each group.

This chapter has outlined the topic group inclination analysis framework proposed in this thesis. The layered approach has been implemented and evaluated with real-world Twitter data. The conclusions in Chapter 7 will discuss the aspects of the generality of the proposed framework

using other data sources from different on-line social network platforms. The next three chapters illustrate the technical details of topic extraction and clustering (Chapter 4), polarity clustering (Chapter 5) and user social interaction and topic group inclination analysis (Chapter 5).

Chapter 4

Topic Group Mining

Chapter 2 illustrated that the major obstacle in recent years to clustering short text documents is that current systems lack knowledge supports, which makes it difficult to achieve an acceptable level of clustering accuracy, as discussed in [Kulkarni et al., 2009]. This chapter discusses how an external knowledge base can be used as a support source to enrich short text documents.

4.1 Proposed Topic Group Mining Method

For short text documents, low term frequency and frequent abbreviation usage are the principle characteristics that directly impact the performance of the bag-of-words model in clustering tasks. Low term frequency only results in inverse document frequency terms in the TF-IDF model, thereby resulting in poor clustering performance, which will be discussed in Section 4.7. The use of abbreviations results in several different representations of the same word. This means that the bag-of-words model treats them as different words, which is not appropriate and will be discussed in Section 4.7.1.

We have adopted the concepts from short text annotation in [Ferragina and Scaiella, 2010] and adapted the concept of document-enrichment strategies from [Hotho et al., 2003] and [Wang et al., 2009] in this thesis because their performance with short text documents has already been proven in several datasets. Through the identification of suitable Wikipedia topics, disambiguation processes for short text documents and

good document-enrichment strategies, we are able to improve the performance of document clustering in the case of short text documents. Our approach is a combination of the following three main tasks:

1. Identification of Wikipedia topics: The main purpose of this step is to identify Wikipedia topics in short-text documents.

2. Document Enriching: This step enriches tweets with Wikipedia topics identified in the first step in each short-text document. Two enrichment strategies will be provided in Section 4.7.

3. Document Clustering: This step identifies topic groups within the documents for both enriched documents from Step 2 and for original documents using a text-clustering algorithm.

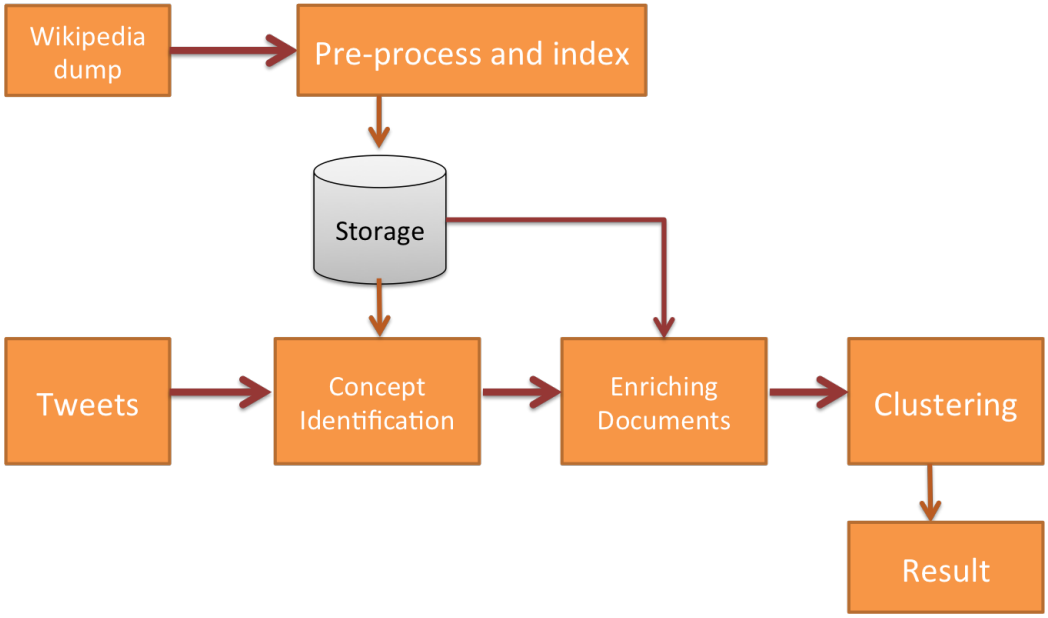

Figure 4.1: The detailed process of topic group mining

As shown in Figure 4.1, the process begins with a pre-processing Wikipedia data dump. We extract all anchors (clickable texts used to link to other Wikipedia pages that were carefully labelled by users, as shown in Figure 4.2) and links in the Wikipedia article and index these into a catalogue. To increase the querying efficiency, we also index all Wikipedia

pages, their content and categories into another catalogue. Then, during the topic identification process, all tweets in our dataset are searched for their related Wikipedia topics. Next, each tweet is enriched with two different strategies based on their related topics. We will discuss this in further detail in a later section. These enriched tweets are also stored in the database for efficiency. Finally, all tweets are clustered with bisecting k-means clustering based on the similarity of their enriched contents.

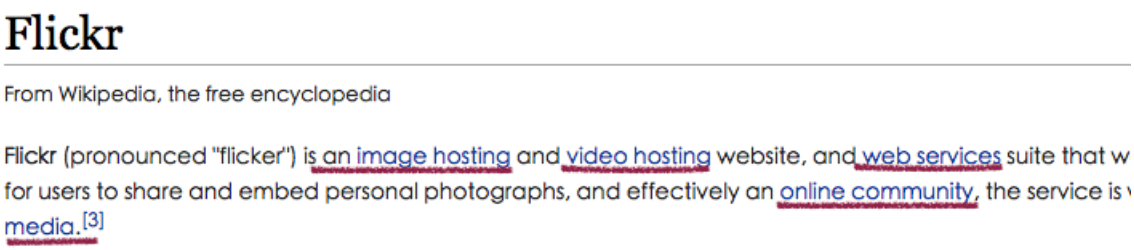

Figure 4.2: An example of Wikipedia Anchors

4.2 Identification and Disambiguation of Wikipedia Topics

We use anchors, which are text used to describe the connection between Wikipedia pages, as our main resource. Basically, an anchor uses words or phrases to describe a Wikipedia page that it links to. An anchor generally uses the title, synonym or acronym of the page. Using this anchor text, we are able to identify Wikipedia topics in the given document not only by the topic in the title but also by phrases, acronyms or synonyms that refer to those topics.

Anchor	Wikipedia Topics
`color`	Colors (film), Colours (film), Colours TV, Colours (TV channel), Colors (TV channel), "Color", Color (manga), Colors (magazine), The Colour, composition, Coloration, Color (band), Colour (band), The Colour (band), Color (album), Colour (Andy Hunter album), Colour (The Christians album), Colours (Adam F album), Colours (Baccara album), Colors (Between the Buried and Me album), Colours (Christopher album), Colours (1972 album), Colours (1987 album), Colours (1991 album), Colours (Eloy album), Colours (Graffiti6 album), Colours (Mark Norman album)
`picture`	Picture (mathematics), Picture (string theory), Picture (band), Picture (album), Pictures (Atlanta album), Pictures (Jack DeJohnette album), Pictures (John Michael Montgomery album), Pictures (Katie Melua album), Film, Pictures (Leon Bolier album), Pictures (Timo Maas album), guilty Chinese scholartree,tree (graph theory), Pictures (film), "Pictures" (short story),

Table 4.1: Examples of anchors and their related Wikipedia topics

In general, each anchor often refers to two or more Wikipedia topics. Thus, we must implement a word sense disambiguation process to select the most appropriate page referred to by an anchor. In this Wikipedia topic identification process, we can split the process into four steps:

1. Pre-processing Wikipedia
2. Anchor Identification
3. Topic Disambiguation
4. Topic Filtering

Each step will be explained in the following sections.

4.3 Pre-processing Wikipedia

We use an English Wikipedia article dump from 4 July, 2012, which contains 4,012,083 articles and is approximately 8.2 GB compressed. To make the processing tractable using the compute resource available it was necessary to take a sample of 1,000,000 pages from the full set of pages (25% of the entire Wikipedia dump). To ensure that the distribution of selected topics reflects the entire Wikipedia structure, we calculate the distribution of the first letter of Wikipedia topics as shown in Figure 4.3. Then, articles from the Wikipedia dump are selected based on their starting letter. For example, Figure 4.3 shows that there are 8.82% of Wikipedia articles that start with the letter "T". Hence, 88,466 articles, whose title starts with the letter "T" will be randomly selected $(4,012,083 \times 0.25 \times 0.0882 = 88,466)$.

Finally, we pre-process and index these pages into two main catalogues to speed up the query.

- Anchor Dictionary: We extract all links and their anchors in Wikipedia pages and construct an anchor dictionary. The anchor dictionary is not like a traditional dictionary. It contains only two important pieces of information: 1) anchors and 2) their corresponding Wikipedia topics.

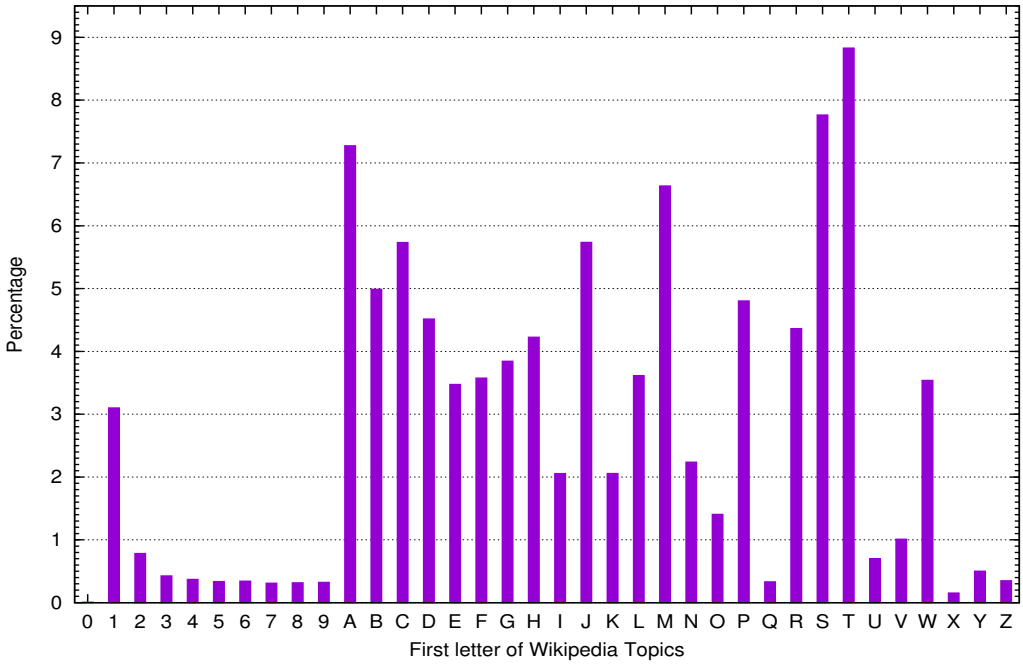

Figure 4.3: Distribution of first letter of Wikipedia page titles

- Wikipedia Pages: We also index the content of the Wikipedia pages, their categories and inbound links for efficient querying.

4.4 Anchor Identification

To identify Wikipedia topics related to each tweet, we need to identify all anchors that appear in the tweet. We follow the steps in Algorithm 1 to find the anchors. $lp(a)$ is the link probability, which can be calculated using the following equation:

$$lp(a) = \frac{link(a)}{freq(a)} \tag{4.1}$$

where $link(a)$ is the number of anchors used as a link and $freq(a)$ is the number of anchors that appear in all of the documents in the collection. Also, in order to utilise most Wikipedia topics, we calculate the number of terms in Wikipedia topics as shown in Figure 4.4. It indicates that 98.48% of Wikipedia topics are less than 6 terms. Hence, in our algorithm we set n-gram to 6 to maximize the coverage.

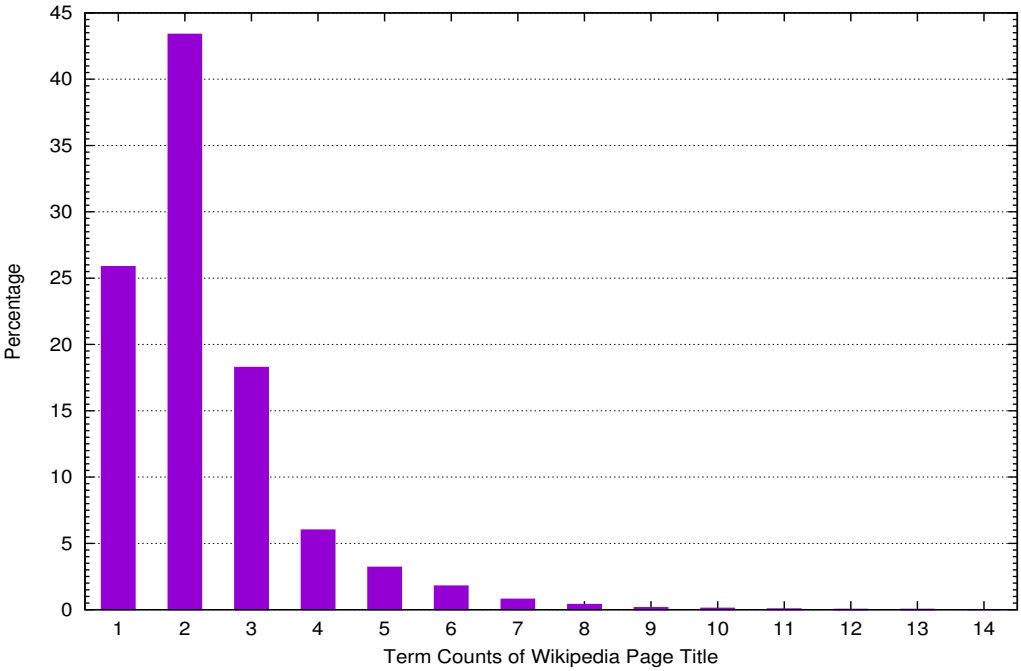

Figure 4.4: Distribution of term count of Wikipedia topics

4.5 Topic Disambiguation

Each anchor in the set of candidate anchors that we obtained from the previous sections could refer to several Wikipedia topics. Therefore, in this step, we disambiguate these topics and assign the most appropriate topic to each anchor.

The same anchor as a term may have different meanings in different contexts and may link to different Wikipedia topics depending on the context of the document. Therefore, we must utilise the disambiguation process for selecting appropriate Wikipedia topics. We use a voting scheme adapted from [Ferragina and Scaiella, 2010]. The concept behind this voting scheme is that every anchor has to vote for all Wikipedia topics related to other anchors in the document, except for topics related to itself. Then, the topics that are given a top e-rank based on their voting score are selected as candidate topics. Finally, we select the most appropriate topic using a commonness score. The details of this voting scheme are as follows:

First, we calculate the relatedness between two Wikipedia topics using the normalised Google distance $rel(p_a, p_b)$ between the inbound link of

Algorithm 1 Document Anchor Identification Algorithm

Require: input document d

```
A =ngrams(d, n=6)
for each word ∈ A do
  if word ∉ dictionary then
    A=A{word}
  end if
end for
for a1 ∈ A do
  for a2 ∈ A do
    if a1 ≠ a2 and substring(a1, a2) and lp(a1) < lp(a2) then
      A=A{a1}
    end if
  end for
end for
```

p_a and p_b, as suggested in [Milne and Witten, 2008a], where p_b is the Wikipedia topic corresponding to anchor b, given $b \in A\{a\}$, as follows:

$$\begin{aligned} linkMax &= log(max(\|inboundlink(p_a)\|, \|inboundlink(p_b)\|)) \\ linkD &= log(\|inboundlink(p_a) \cap inboundlink(p_b)\|) \\ linkMin &= log(min(\|inboundlink(p_a)\|, \|inboundlink(p_b)\|)) \\ rel(p_a, p_b) &= \frac{linkMax - linkD}{log(W) - linkMin} \end{aligned} \tag{4.2}$$

where W is the number of articles in Wikipedia.

Next, we calculate the voting score of anchor b to topic p_a by averaging the relatedness between all corresponding topics of anchor b to topic p_a, with a prior probability known as commonness $Pr(p_b|b)$, as shown in the following equation:

$$vote_b(p_a) = \frac{\sum_{p_b \in P_g(b)} rel(p_a, p_b) \cdot Pr(p_b|b)}{\|P_g(b)\|} \tag{4.3}$$

Then, the total score assigned to p_a can be calculated as

$$rel_a(p_a) = \sum_{b \in A\{a\}} vote_b(p_a) \tag{4.4}$$

Using only this score to assign the topics to the anchors may not be sufficient because, as mentioned in [Milne and Witten, 2008b], balancing the score and commonness is the main factor that affects the performance. We are primarily concerned with computational efficiency in this thesis because the platform needs to process a large number of tweets. Therefore, we perform disambiguation using only the threshold method to filter out unrelated topics. We use the same parameter setting as proposed in [Ferragina and Scaiella, 2010], which have the best result in their experiments. This is performed as follows:

- Remove all topics that have $rel_a(p_a) < \delta$, where $\delta = 0.3$
- Remove all topics that have

$$\frac{rel_a(p_{top\_a}) - rel_a(p_a)}{rel_a(p_{top\_a})} > \epsilon \tag{4.5}$$

  in which p_{top_a} is a topic corresponding to the anchor a that obtains the highest rel score and $\epsilon = 0.30$
- Finally, the topic that has the highest commonness $Pr(p_a|a)$ is assigned to an anchor a

4.6 Topic Filtering

However, after the disambiguation step is applied, uncorrelated topics may still be present. Therefore, we use a final topic filtering step to remove all topics that are not related to other topics. We filter out unrelated anchors based on the coherence between selected topics from the topic disambiguation step. To calculate the coherence score, we use the average relatedness between selected topics as follows:

$$coherence(a \rightarrow p_a) = \frac{1}{\|S\| - 1} \sum_{p_b \in S\{p_a\}} rel(p_a, p_b) \tag{4.6}$$

where S is the number of selected topics. Next, we filter out unrelated anchors that satisfy the following condition:

$$\frac{coherence(a \rightarrow p_a) + lp(a)}{2} < \epsilon \tag{4.7}$$

where $\epsilon = 0.2$ and $lp(a)$ is the link probability of an anchor a.

4.7 Short-Text Document Enriching

A tweet is an extremely short text document that has a very low term frequency and that generally consists of only important words. This is because tweets are limited to 140 characters. Therefore, most terms in tweets tend to appear only once. For users to fully express their thoughts, they need to select only the important words that express all the information that they want to communicate.

Given the characteristics of these short text documents, various problems are produced when applying TF-IDF weighting to model the document. First, the low term frequency in each document results in only inverse document frequency terms. Second, most tweets tend to contain only important words. In some cases, important words will have a lower inverse document frequency compared to unimportant words. An example is given to explain this issue. The TF and TF-IDF vectors of the tweet

"Switch now! Flickr is much faster!"

are given below.

$$TF = (\text{switch:1, now: 1, flickr:1, faster:1}) \tag{4.8}$$

$$TF-IDF = (\textbf{switch:0.51}, \text{ now: 0.31, flickr:0.41, } \textbf{faster:0.56)} \tag{4.9}$$

Clearly, the term "Flickr" is more important than "switch" and "faster" in the original tweet, but the TF-IDF vector indicates otherwise. This issue means that TF-IDF cannot reflect the real importance of the term and results in a poor clustering performance for this phrase. Moreover, another drawback of the bag-of-words model is that it cannot reflect the term-related relationships. Documents with highly related terms will have the same cosine similarity as those with unrelated terms. This can be a problem because highly related terms should have higher cosine similarly. For example, given two pairs of terms "Flickr"-"SmugMug" and "Google"-"Lake", it is obvious that the first pair is related to a photo service, whereas the second pair does not have any relation. However, the cosine similarity will still be the same between the two pairs, which will be illustrated in the following subsection.

[Hotho et al., 2003] succeeded in using three different strategies to enrich the TF-IDF vector with background knowledge based on WordNet. These two strategies consist of adding corresponding WordNet concepts, replacing terms with WordNet[1] concepts and replacing the term vector with a concept vector. Additionally, [Wang et al., 2009] obtained good results by enriching their documents with semantic-related terms based on Wikipedia knowledge. In our proposed method, we adapted the strategies of [Hotho et al., 2003] and [Wang et al., 2009] to enrich the tweets using Wikipedia knowledge.

4.7.1 Strategy 1: Add Wikipedia topics

We replaced and supplemented terms in each tweet with the corresponding Wikipedia topic. The rationale for this is that one of the problems that reduces the performance of the bag-of-words model is that each Wikipedia topic can be mentioned by several different words/phrases. For example, there are many words/phrases that refer to the topic "Google", such as "Google Inc.", "GOOG", and "google", as shown below.

"Google hits its new high in the stock market"

"GOOG hits its new high in the stock market"

This problem leads to low cosine similarity between any two documents because words such as "GOOG" and "Google" are treated as different words. Therefore, to reduce the errors caused by different representations of the same topic, we replace them with the Wikipedia topic, which transforms them into the same representation. Moreover, as we mentioned previously when discussing the problems in TF-IDF weighting, we also add related Wikipedia topics into tweets to provide important terms with a higher score.

Furthermore, the TF-IDF problem in the short text context still needs to be addressed. Adding the related Wikipedia topics allows the important term to have a higher score. Take the tweet "Switch now! Flickr is much faster!", which was mentioned in the previous section, as an example. After applying this strategy, which adds the related Wikipedia topic to the tweet, a new TF and TF-IDF will be generated as follows:

[1] http://wordnet.princeton.edu/

$$TF = \text{(switch:1, now: 1, flickr:2, faster:1)} \tag{4.10}$$

$$TF - IDF = \text{(switch:0.27, now: 0.28, } \textbf{flickr:0.71}\text{, faster:0.34)} \tag{4.11}$$

4.7.2 Strategy 2: Add Wikipedia topics and categories

We extend the first strategy by adding categories of Wikipedia topics corresponding to each term in a tweet. This is performed because another problem of the bag-of-words model is that this model cannot capture semantic relationships between two related terms. Adding Wikipedia categories solves the problem of semantic relationships between tweets. For example, given two tweets that only contain one on-line photo service's name, such as "Flickr" or "SmugMug", the cosine similarity between the two terms is 0. However, if we add *"Photography websites"*, which is one of the common categories between these two terms, then we will obtain a cosine similarity that is greater than 0. The difference observed when adding a Wikipedia topic is illustrated as follows:

1. First, the TF-IDF values of both tweets are the same because there is only one term in each tweet. They can be represented as follows:

$$Vec_{flickr} = (flickr : 1)$$

$$Vec_{smugmug} = (smugmug : 1)$$

   The cosine similarity between the above vectors can be calculated as follows:

$$sim_{cosine}(Vec_{flickr}, Vec_{smugmug}) = \frac{Vec_{flickr} \cdot Vec_{smugmug}}{\left\|Vec_{flickr}\right\| \|Vec_{smugmug}\|} = 0$$

2. Then, the common category *"Photography websites"* is added to both tweets. The new TF-IDF vector will be changed to

$$Vec_{flickr} = (flickr : 0.5, photographywebsites : 0.5)$$

$$Vec_{smugmug} = (smugmug : 0.5, photographywebsites : 0.5)$$

Finally, the new cosine similarity of the two above vectors, which is considerably higher than the previous value, is re-calculated as follows:

$$sim_{cosine}(Vec_{flickr}, Vec_{smugmug}) = \frac{Vec_{flickr} \cdot Vec_{smugmug}}{\left\|Vec_{flickr}\right\| \|Vec_{smugmug}\|} = 0$$

Adding Wikipedia categories can overcome the problem of semantic relationships in the bag-of-words model. Therefore, we also add Wikipedia categories to documents in addition to replacing and adding Wikipedia topics in Strategy 2.

4.8 Document Clustering

After tweets are enriched with Wikipedia knowledge, we mine these tweets by clustering them into groups based on topics. However, before we can apply the clustering algorithm, we need to pre-process the enriched tweets resulting from applying the two strategies mentioned in the previous section into TF-IDF vectors using the following five steps.

1. *Tweet Filtering.* Because certain tweets in our dataset do not contain any useful information, these tweets are outliers and can reduce the performance of the document clustering process. Tweets that meet the following conditions will be removed.

   - Tweets that only include webpage link(s)
   - Tweets that only include stop word(s)
   - Tweets that only includes username(s)

2. *Stop Word Filtering.* Removing stop words helps improve the performance of the model for the clustering and classification tasks. We used the NLTK stop word list and added extra stop words that mostly occur only on Twitter, such as “lol” and “huh”, to the list.

3. *Word Stemming.* Stemming is a method that attempts to reduce a word into its root form. The effects of stemming in the text clustering and TF-IDF models are shown in [Kantrowitz et al., 2000]. One of the key advantages of stemming is that it reduces

the size of the dictionary. Moreover, it enables us to match the same word with its different forms.

4. *Tweet Dictionary.* We construct a dictionary of words that appear in the tweets after stemming and filtering out words that occur less than five times and words that occur in more than half of all of the tweets in the dataset.

5. *TF-IDF Vectors* After we process all of the tweets using the above four steps, we convert the contents of the tweets into TF-IDF vectors using Equation 2.4.

After this pre-processing, we apply bisecting k-means clustering. Numerous studies have shown that affinity propagation is the best among several clustering algorithms in short text clustering tasks. However, there are many criticisms of affinity propagation for problems with large datasets. This issue of the scalability of affinity propagation was mentioned in [Fujiwara et al., 2011] and [Zhang et al., 2008a]. Therefore, due to the size of our dataset, we decided to use bisecting k-means clustering because of its scalability and efficiency. The details of this clustering approach are shown in Algorithm 2.

4.9 Conclusion

The problems of topic identification and classification can be addressed using Wikipedia, which is a well-organised, crowd-sourced service, as the background knowledge base. The series of processes presented in this section has provided a way to identify topic(s) from short text documents (tweets). The process also has the ability to remove ambiguous topics and find the most suitable topic during the process. The next chapter will illustrate the document inclination mining technique, which will process the results generated in this section.

Algorithm 2 Bisecting k-means Clustering Algorithm

Input: The dataset $\mathbb{D}$, Number of interaction $ITER$ for the bisecting step and desired number of clusters k

Output: A set of clusters R=$\{R_0, R_1, \ldots, R_{k-1}\}$

Initalization: Let V= $\mathbb{D}$, R=$\{\}$

1: **for** counter = 1 to $k-1$ **do** /* Clustering Step */
2: **for** i = 1 to $ITER$ **do** /* Bisecting Step */
3: 1. Randomly select two data points from V as starting centroids
4: 2. Find 2 partitions from the set V using the standard k-means algorithm
5: **end for**
6: a) Select the cluster that has the minimal squared sum error from Bisecting step as V_1
7: b) Assign the remaining partition to V , $V = V_2$
8: c) Add V_1 to the desired cluster set $R = R \cup V_1$
9: **end for**
10: Add V_2 to the desired cluster set $R = R \cup V_2$
11: **return** R

Chapter 5

User Inclination Mining

In this chapter, we determine the inclination of users based on their tweets. For this purpose, two types of user inclination are used: "positive" and "negative". If users in a specific topic group support or love the topic of the group to which they belong, the user's inclination will be recognized as "positive"; the inclination will be "negative" for the opposite case. Our user inclination algorithm treats each user as the central unit for inclination analysis. Unlike traditional sentiment classification, which only focuses on documents, user inclination provides a new perspective. Two key factors of user inclination will be analyzed in this thesis: documents and user on-line relationships.

5.1 Research Problem Definition

In this thesis, two assumptions are proposed:

1. Users express their inclination towards the topic and related topics consistently in tweets. For example, the user should share the same inclination towards "Democratic Party" and "Tax Deduction".

2. Each member in a topic group will only have one inclination in each short-text document towards the topic group. In other words, in this thesis, we will ignore any post that contains two or more inclinations.

Benefiting from the the topic classifier described in the previous section, all short-text documents grouped in the same topic group are con-

sidered to be related to the specific topic. Then, we define the topic of each topic group as T, and we collect a set of users $U_T = \{u_i | 1 \leq i \leq number\ of\ users\ in\ the\ topic\ group\}$. The definitions related to the research problem to be solved are as follows:

DEFINITION 1: ***User Relationship Graph*** on topic T is represented by $G_T = (V_T, E_T)$. The nodes $U_T = \{u1, u2, ..., u_n\}$ represent a set of users; the edge with a solid line is an inter-user social relationship, which can be indirect or direct for different micro-blogging platforms; and the dashed line is a user-short-text-document social relationship, which is formed by the users and documents in the group.

The types of social relationships vary on different micro-blogging platforms. In general, two types of relationships can be observed:

1. **Inter-user relationships:** A virtual on-line social network relationship between two users. For example, *following* is the major inter-user relationship on Twitter, whereas *friends* and *following* are inter-user relationships on Facebook.

2. **User-Short-Text-Document relationships:** These relationships illustrate user's behaviours on on-line social network platforms, e.g., when a user leaves a comment on another user's tweet or retweets some tweets that interested him.

A social network can be modelled as $G_T = (V_T, E_T)$, where the vertices represent users and the edges represent relationships between users, as shown in Figure 5.1. A user relationship graph can be directed or undirected based on the design of the individual on-line social network platform. For example, a directed graph can be used to model asymmetric relationships on Twitter. In contrast, symmetric relationships, e.g., Facebook's friend relationships, can be modelled by an undirected graph. If two users form an inter-user relationship, they will be defined as *neighbours* and linked together in G_T. Take Twitter as an example; *following* is a type of asymmetric inter-user relationship.

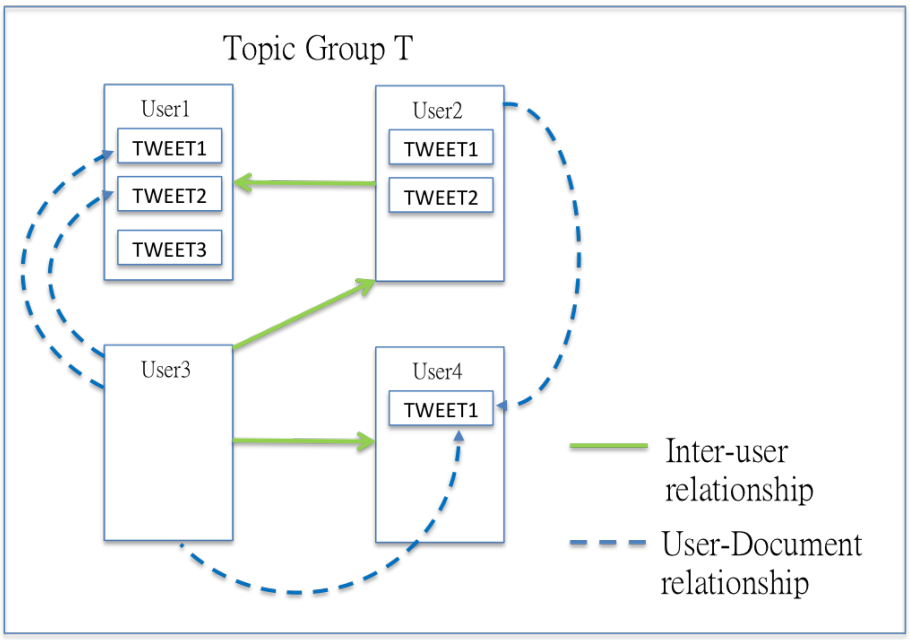

Figure 5.1: An indirect user relationship graph on topic q

Moreover, the user-short-text-document relationship involves an *interaction degree* between two nodes. In other words, the edge of the user relationship graph will be weighted. The greater the interaction between two users is, the stronger the influence will be. In this thesis, we analyse Twitter. Three types of user-short-text-document relationships on Twitter can be identified as follows:

- Retweet: A user copies his interested tweet to his own tweet stream.
- Reply: A user leaves a comment on a tweet.
- Favorite: A user shows his likeness or support of a tweet.

We will classify all users in G_T based on the characteristics mentioned above. In this thesis, the goal is to classify each user in every topic group with positive or negative inclinations towards the group's topic. A graph-based method will be employed by utilising the content of the document and user relationship. The target function is given in 5.1, where $\Phi(u_i)$ represents the inclination of u_i. Documents related to the group's topic of u_i are $D_{u_i}T$, and r is in $\{positive, negative\}$.

$$P(\Phi(u_i = r|D_{u_i}T, G_T), for\ all\ users\ in\ U_T \tag{5.1}$$

To summarize, function 5.1 allows us to classify a set of users in topic groups and to determine whether they support the group to which they belong with help from their interaction relationships.

DEFINITION 2: ***Inter-user relationship***: A virtual on-line social network relationship between two users. This relationship varies between different on-line social network platforms.

DEFINITION 3: ***User-short-text-document relationship***: An on-line interaction between a user and a document (post). This relationship also varies between different on-line social network platforms.

DEFINITION 4: ***Interaction degree***: Level of influence between two users based on how frequently a user interacts with other related documents (tweets). A higher score represents a stronger influence.

5.2 Short-Text Document Inclination Classification

The goal of short-text document inclination classification is to identify the inclination of the user by analysing the inclination of their tweets. Each user in topic group G_T has at least one tweet that is related to the group topic. Namely, the user might express his inclination towards the group topic. If all documents in the topic groups are classified into binary inclinations, i.e., positive and negative, then we can, in theory, also identify the user's inclination. As mentioned in the previous section, one of our hypotheses is that each short-text document has only one inclination towards the topic. Each short-text document will be transformed into a vector space after the inclination extraction step. A supervised machine learning algorithm will be used to construct the model. Figure 5.2 details this procedure.

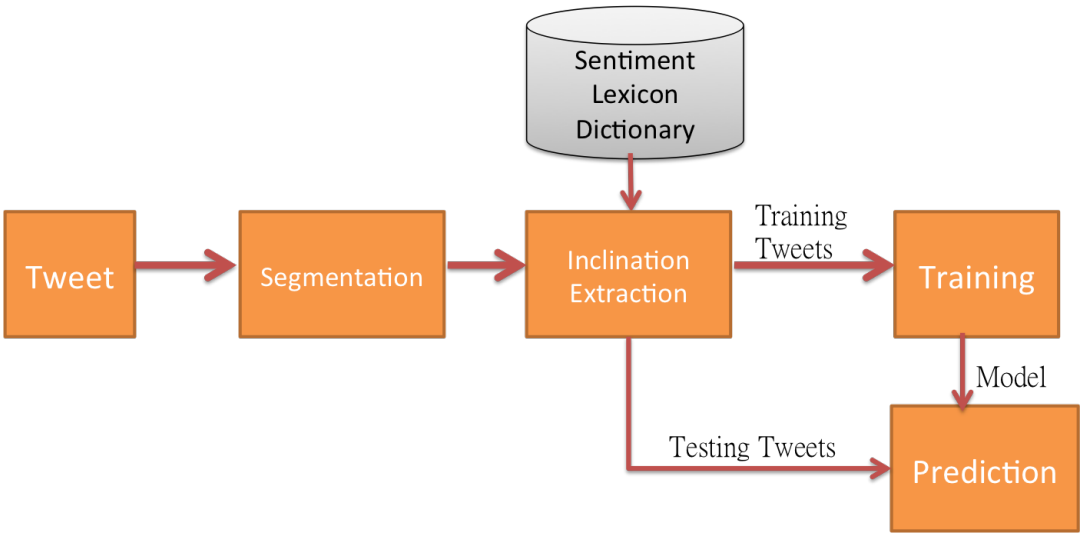

Figure 5.2: The process of document inclination classification

As shown in Figure 5.2, documents will first be segmented for further processing. Then, a small portion of the segmented documents will be selected and trained for predicting user inclination. Two supervised machine learning algorithms have been implemented in this thesis: (a) a Bag-of-Words model + SVM and (b) a Naïve Bayes model. An external resource was introduced into the process to identify the polarity of the day phrase and the overall inclination. In general, the resource will be a well-organized and formatted lexicon database, in which each word is carefully tagged with its polarity. In this thesis, we adopt the MPQA (Multi-Perspective Question Answering) subjectivity lexicon database[1] provided and compiled by the University of Pittsburgh and the Sentiment Lexicon from the University of Illinois at Chicago[2] (UIC).

5.3 Bag-of-Words model + SVM

The primary feature of the bag-of-words model is that it handles unordered data. Because it is a word-based model, the grammar and ordering of words do not affect the model. Each document is segmented by NTLK[3] (Natural Language Toolkit), a well-known natural language processing toolkit written in Python; any unnecessary symbols are removed.

[1] http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
[2] http://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html#lexicon
[3] http://www.nltk.org/

Then, a popular supervised machine learning algorithm, a Support Vector Machine (SVM), is introduced into our platform. SVM has been shown to be very efficient and accurate when applied to traditional document clustering. The goal of the SVM algorithm is to find the maximum margin, which is the smallest distance between samples and the boundary. To address this issue, the parameters need to be optimized to find the boundary. In this thesis, LibSVM[4] is used to classify the inclination of documents.

5.4 Naïve Bayes Classifier

Naïve Bayes classification, which is a supervised machine learning method that is widely used in various fields of research, is derived from the Bayes theorem. In this thesis, given a document D that is to be processed by the Naïve Bayes classifier, the classifier will predict that D belongs to the class that has the highest posterior probability. Hence, $P(D)$ will be maximized. The class C for which $P(c|D)$ is maximized is called the maximum posteriori hypothesis. Equation 5.2 shows that only $P(D|c)P(c)$ needs to be maximized because $P(D)$ is a constant for all classes.

$$C_{map} = \arg\max_{c \in \mathcal{C}} P(c|D) = \arg\max_{c \in \mathcal{C}} \frac{P(D|c)P(c)}{P(D)} = \arg\max_{c \in \mathcal{C}} P(D|c)P(c) \tag{5.2}$$

, where C = {Unknown/Neutral, Positive, Negative}

C_{map} from Equation 5.2 is the class with the maximum probability, and C is a set of inclination classes that consist of unknown/neutral, positive and negative.

5.4.1 Multinomial and Bernoulli Event Model

Two event models are commonly used in the Naïve Bayes classifier. The first is the multinomial event model, in which the document is transformed into an integer vector. Elements of the integer vector represent the frequencies of corresponding segmented terms in the document. The

[4]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

	Multinomial model	Bernoulli model
random variables	$D = t$, if t is in the document	$e_t = 1$ if t appears in the document
Vector Presentation	$D = \{t_1, ..., t_i, ..., t_n\}$, $t_i \in V$	$D = \{e_1, ..., e_i, ..., e_N\}$, $e_i = \{0, 1\}$
multiple occurrences	calculated	ignored

Table 5.1: The differences between the multinomial and Bernoulli event models

second is the Bernoulli event model, in which the document is transformed into a binary vector, where elements of the binary vector represent the absence or presence of corresponding segmented terms in the document. Table 5.1 compares the multinomial and Bernoulli models.

First, terms are segmented from each document using the lexicon database. Namely, terms that are only considered as patterns need to be listed if they are in the subjectivity lexicon database. Then, the two different extraction models mentioned above will be presented. When the multinomial model is used, every document D will be transformed into $D = \{t_1, ..., t_i, ..., t_n\}$, where $\{t_1, ..., t_i, ..., t_n\}$ is the frequency of terms that occur in document D. In contrast, the binary vector, which uses term absence as the element, is used in the Bernoulli model. Every document D will be transformed into $D = \{e_1, ..., e_i, ..., e_n\}$, where $\{e_1, ..., e_i, ..., e_N\}$ discards the term occurrence times in D and N is the total number of segmented terms.

In this thesis, we assume that each inclination class is conditional independent, which is logical in our case because the inclination of each tweet is independent of others to reduce the computational complexity in calculating $P(D|c)$. This assumes that each element of the vector as shown in 5.3 and 5.5 are conditionally independent to others. Finally, the multinomial and Bernoulli classification methods are implemented to classify the inclination of the document into three labels, namely, unknown/neutral, positive and negative, as shown in Equations 5.4 and 5.6.

$$Multinomial: \ P(D|c) = P(\{t_1, ..., t_i, ..., t_n\}|c) = \prod_{1 \leq i \leq n} P(t_i|c) \quad (5.3)$$

$$C_{map} = arg\max_{c \in \mathcal{C}} P(D|c)P(c) = \prod_{1 \leq i \leq n} P(t_i|c) \quad (5.4)$$

$$Bernoulli: \ P(D|c) = P(\{e_1, ..., e_i, ..., e_N\}|c) = \prod_{1 \leq i \leq N} P(e_i|c) \quad (5.5)$$

$$C_{map} = arg\max_{c \in \mathcal{C}} P(D|c)P(c) = \prod_{1 \leq i \leq N} P(e_i|c) \quad (5.6)$$

5.5 User Relationship Analysis

User relationship refers to the relationship between two or more users on on-line social network platforms. This type of relationship plays an important role in these platforms. From a regular user's perspective, he is able to leave comments to his connected friends or receive feedback from other users. In addition, this would even be helpful for corporate users. Such users can receive feedback from customers regarding specific products and make changes in a timely manner by analysing the comments on the on-line social network platform.

Furthermore, [Thelwall, 2010]'s study on emotion homophily on MySpace[5] found that the level of positive emotion exchanged and received between friends exhibits a weak but statistically significant level of correlation. Hence, we also introduce user relationship as a feature for the analysis process in this thesis. A comparison of the interaction relationships of two major on-line social networks, Facebook and Twitter, is presented in Table 5.2. In this thesis, the user relationship between two on-line social network users is defined as an inter-user relationship such as friends. Presumably, if an inter-user relationship exists, both users might know each other to some extent. The possibility of both users sharing the same inclination is very high based on [McPherson et al., 2001]'s research on the characteristics of homophily in social networks. In addition, a user-short-text-document relationship is also proposed in this

[5]https://myspace.com/

	Twitter	Facebook
User Relationships	Following	Friends
		Follows
Document Behaviours	Favorites	Like
	Retweet	Share
	Reply	Response

Table 5.2: Comparison of interaction relationships between Twitter and Facebook

thesis. It can also be used to identify the degree of interaction between two users.

5.6 User Relationship Graph

As mentioned in Chapter 1, one of the research questions that we wish to answer is can the social relationships in an on-line social network be utilised to improve the results of document inclination analysis? Hence, we construct a user relationship graph $G_T = (V_T, E_T)$ of topic T, where each node $u_i \in V_T$ represents an individual user and edge e_{ij} exists if there is an inter-user relationship between u_i and u_j. There are two types of inter-user relationships that can be constructed based on the platform of interest. For example, Facebook has *friends* and *follows* user interactions. Therefore, if the "*friends* inter-user relationship" is a symmetric relationship, an indirect user relationship is constructed as shown in Figure 5.1. In contrast, the "*follows* inter-user relationship" is an asymmetric relationship, in which we assume that the followees have a certain influence on their followers, in their opinions. The inter-user relationships will be a direct graph. For convenience, the user relationship graph UR will be formulated in Equation 5.7.

$$UR_{|v|\times|v|} = [u_{i,j}] \tag{5.7}$$

where the elements of UR are given by Equation 5.8.

$$u_{i,j} = \begin{cases} 1, \text{ if inter-user relationship exists in } u_i \text{ and } u_j \\ 0, \text{ otherwise} \end{cases} \tag{5.8}$$

5.7 Edge Pruning and Weighting

We assume that two people that have a closer relationship will have a greater influence on each other. One example is that family members can have a considerably stronger influence on other family members compared to a regular friend in common. Hence, an obvious method for determining the influence of the user is to utilise the user relationships. For example, user u_i on Twitter will press the *Favorites* or *Retweet* button on user u_j's tweet if u_i agrees with the content to some extent. Moreover, user u_i may use the *Reply* function to leave his comments to u_j. In this thesis, we will focus on the number of retweets and replies; the other interactions, such as favorites, will be ignored. Then, the frequency of these interactions will be calculated as the *interaction degree*, which represents the degree of influence between users in the user-short-text-document relationship.

The weight of the link between two users $w(u_i, u_j)$ in user relationship G_T will be combined scores of the retweet and reply ratios of two users. The detailed formula for calculating the combined score between two users is shown in Equation 5.9. $\beta_1, \beta_2, ..., \beta_n$ are adjustable variables that represent the levels of influence among different interactions. Function *interaction* corresponds to different types of user interactions in a user-short-text-document relationship such as *retweeting* or *replying* to other user's tweets on Twitter. This can be easily extended when the new relationship on a social network platform is taken into account. The value of $interaction_n(u_i, u_j)$ is the sum of users u_i that have interactions of type n with user u_j; $interaction_n(u_i, u_{others})$ represents the interaction between user u_i and users other than u_j.

$$\begin{aligned} w(u_i, u_j) = & \beta_1 \frac{interaction(u_i, u_j)}{interaction(u_i, u_{others})} + \beta_2 \frac{interaction(u_i, u_j)}{interaction(u_i, u_{others})} \\ & + ... + \beta_n \frac{interaction(u_i, u_j)}{interaction(u_i, u_{others})} \end{aligned} \tag{5.9}$$

, where $\beta_1 + \beta_2 + ... + \beta_n = 1$

To summarise, the details of the user relationship analysis algorithm are given in Algorithm 3.

5.8 User Inclination Analysis

In this section, we discuss how the inclinations of users are determined. By utilising the results from the previous section, the inclinations of short-text documents have been labelled, and user relationships are modelled by interactions found on Twitter. In addition, we made the assumption that it is independent between the content of an individual user and the labels of user's neighbours. We introduced the relation labelling technique [Kittler and Illingworth, 1986][Angelova and Weikum, 2006] shown in Equation 5.10, which is a graph-based classification algorithm, to this thesis. This technique can adjust a user's inclination based on his neighbours. When the iteration is stable, meaning that the magnitude of changes falls below a stop parameter π, the result will be the maximum probability for each user's inclination of the topic.

$$[P(\lambda(u_i) = c|D_{u_i}T, G_T)]^r = \Phi_{c,i} \sum_{\lambda(N(u_i))} [P(c|N(u_i))P(N(u_i))]^{(r-1)} w(u_i, u_j) \tag{5.10}$$

where $\Phi_{c,i}$ represents the sum of posterior probabilities of all possible labellings of neighbours. And $N(u_i)$ represents the neighbours of u_i, $w(u_i, u_j)$ is the weighted social influence of user j to user i, r is the number of iterations, and $0 \leq j \leq |N(u_i)|$.

Figure 5.3 is an example of the relaxation labeling process. It has two labels $L = \{\blacksquare, \blacktriangle\}$. Each node represents a short-text document, and the edge is the relationship between two short-text documents. First, the node d in Figure 5.3 (a) has an unknown label that needs to be classified. The matrix shown in 5.3 (b) indicates the probability of a relationship between neighbours observed from the training dataset. First, we label node d as "■" based on its textual features, as shown in 5.3 (c). Then, the algorithm will constantly adjust the label of d according to its neighbours. Finally, the maximum probability of each node is found while the iteration is stable. The node d is accordingly classified as "▲", as shown in 5.3 (d).

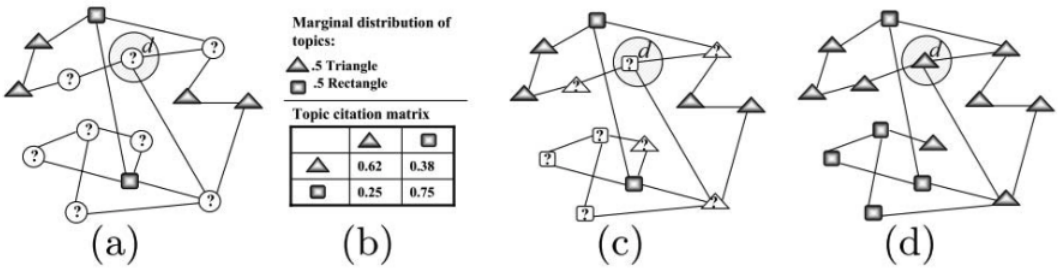

Figure 5.3: An example of the relaxation labelling process

Algorithm 3 User Relationship Analysis Algorithm

Input:

V_T: a set of users in the same generated topic group

G_T: a graph structure of social network

$action_N$: a set of user interaction parameters

Output:

UR: User Relationship of G_T

WUR: the weighted User Relationship matrix among V_T

$UR = 0$

$WUR = 0$

for each u_i in V_T **do**
  for each u_j in V_T **do**
    if exists an inter-user relationship between u_i and u_j in G_T **then**
      $UR[u_{ij}] = 1$
    end if
  end for
end for

/* Weighting link between u_i and u_j */
for i = 0 to $|V_T|$ **do**
  for j = 0 to $|V_T|$ **do**
    if exists a user-short-text-document relationship between u_i and u_j in G_T **then**
      for k = 0 to N **do** /* Types of interaction of platform */
        $w(u_i, u_j) = w(u_i, u_j) + action_n \times$ (counts of $interaction_k$ between u_i to u_j) / (counts of $interaction_k$ between u_i to other users)
      end for
    end if
  end for
end for
for each element in $WUR[i, j]$ **do**
  $WUR[i, j] = UR[i, j] * w(u_i, u_j)$
end for

Chapter 6

Opinion Identification In Topic Groups

In this chapter, we will propose a framework that can identify opinion leaders from specified topic groups. The framework is constructed in two phases. First, we construct a user-interaction graph in a specific topic group using the proposed algorithm. Then, we propose a two-stage classification method to assist us in finding opinion leaders. The framework is shown in Figure 6.1.

The process begins by constructing user-interaction graph G within the topic group generated in the previous chapter. Then, there is a two-stage classification process. First, the user will input how many opinion leaders, for instance K, that need to be identified from user-interaction graph G. Then first- and second-stage classification algorithm will be applied to graph G, which will generate a set of candidate opinion leader clusters. Finally, a selection mechanism is proposed to find the opinion leaders.

6.1 Constructing the user-interaction graph

The first step in identifying leaders within a specific topic group is to construct a user-interaction graph of the group. When *user A* retweets or replies to *user B*'s tweet T, we believe that *user A* has read all tweet T's

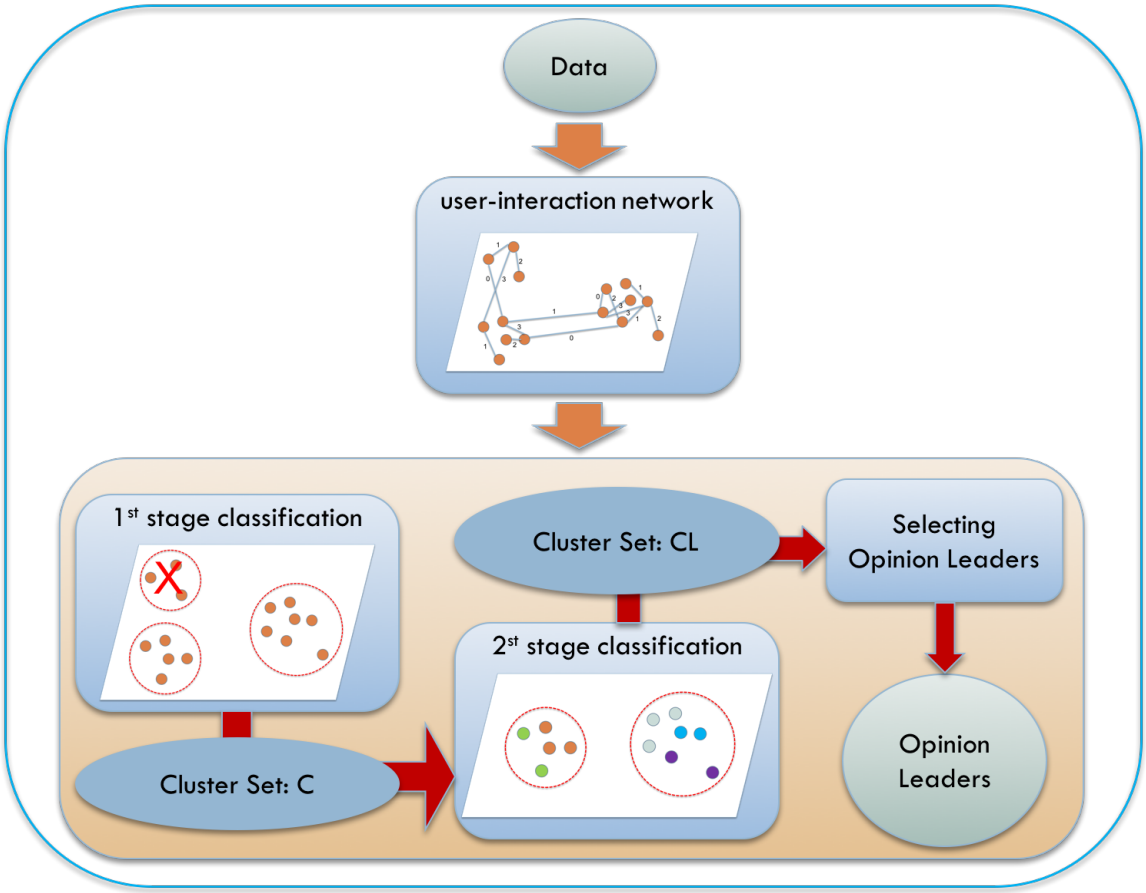

Figure 6.1: Proposed Opinion Leader Identification Framework

replies. *User A* has a degree of feeling as to whether he wants to retweet or reply to the tweet to express his own opinion and interact with others or seek answers. Hence, we assume that *user A* is impacted by *user B*, who posts tweet *T*. At the same time, *user B* can read the reply from *user A*. Thus, there is a possibility that *user B* will also be impacted by *user A*'s opinion. Based on the assumption previously described, we construct the user network as an undirected graph $G = (V, E)$, where V represents a set of users and E is a set of edges. If there is a retweet or reply interaction between V_i and V_j, the edges E_{ij} will be established. Each edge will be given a weight w, the value of which will be from 0 to 3.

Two factors need to be taken into account to assign weight w to the edges. First, by design, Twitter shows the latest retweet or reply at the top of thread so the viewer can obtain the latest information. Hence, earlier retweets or replies will be pushed to the bottom of the thread by

the latest ones. A user has lower possibility of reading earlier replies and retweets, which results in a lower impact on the current user. Meanwhile, when a user is browsing another user's tweets, he tends to reply fairly instantly once he spots the tweet that interests him, for example, when user A just replied or retweeted few minutes ago. The chance that user B reads and is impacted by the comment of user A is relatively high. On the other hand, if user A replied or retweeted at 7am, and user B starts to read the tweet at 9pm, there will have been many comments between 7am and 9pm, which make user A's comment difficult to be read by user B. Hence, user B is less likely to be impacted by user A. In my research, I assume two users that have similar replies or retweet periods have stronger impacts on each other than users who have different reply or retweet periods. Finally, the weight w_{ij} is given based on the reply or retweet period. The details on constructing the user-interaction graph are shown in Algorithm 4.

Algorithm 4 User-Interaction Graph Generation

Input:

U: a set of users

UR: a set of interaction between users

T: the average retweet/reply period of users in U

Output:

User-interaction graph $G(V, E)$

$w = 0$

1: **for** each U_i in U **do**
2: **for** each U_j in U where $i \neq j$ **do**
3: **if** UR_{ij} exists between U_i and U_j **then**
4: Connect the edge E_{ij} between U_i and U_j
5: Assign weight w to E_{ij} where $w = 4 - |T_i - T_j|$
6: **end if**
7: **end for**
8: **end for**

To begin, we will calculate the average retweet/reply period avg_p of each user based on their retweet/reply timestamp. This step helps us understand the regular retweet/reply period of each user. In this thesis,

I divided the retweet/reply period into four segments, and each segment has the corresponding value: $t_i = i$, where $i = 1, 2, 3, 4$. The main reason that time was divided into these segments is to map most users' ordinary daily activities. The four time segments are listed below:

- $t_1 = 1$ if $avg\_p = [07:00, 13:00)$
- $t_2 = 2$ if $avg\_p = [13:00, 19:00)$
- $t_3 = 3$ if $avg\_p = [19:00, 01:00)$
- $t_4 = 4$ if $avg\_p = [01:00, 07:00)$

Next, we can feed each user's avg_p into Algorithm 4. If there is an interaction between U_i and U_j, an edge will be established. Then, the weight of the edge will be calculated by $w_{ij} = 4 - |T_i - T_j|$. For example, if U_i is used to reply/retweet at the time period [07:00, 13:00) and U_j is at [19:00, 01:00), then w_{ij} will be calculated as $w_{ij} = 4 - |1 - 3| = 2$. Finally, the generated user-interaction graph would look like Figure 6.2.

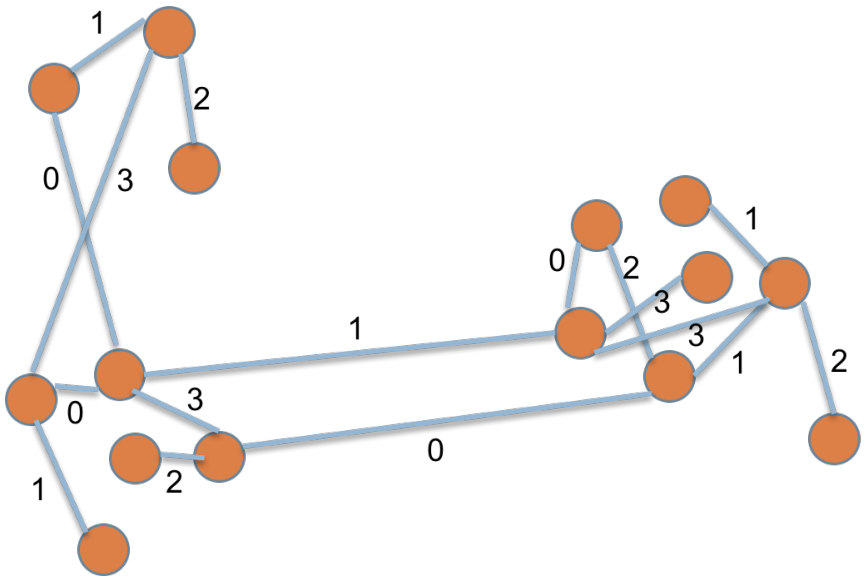

Figure 6.2: An example user-interaction graph

6.2 Opinion Leader Identification Algorithm

Once the user-interaction graph is built, we can start to identify opinion leaders from the graph. This process comrises three procedures: a) First-stage classification: Detecting network structure; b) Second-stage classification: Generating opinion leader candidates; and c) Selecting opinion leaders. Each procedure will be described in the following sections.

6.2.1 First-stage classification: Detecting network structure

In this stage, we proposed a modified hierarchical classification algorithm to eliminate the insignificant clusters to ease the computational load for the second stage. The overview of first-stage classification process is illustrated in Figure 6.3.

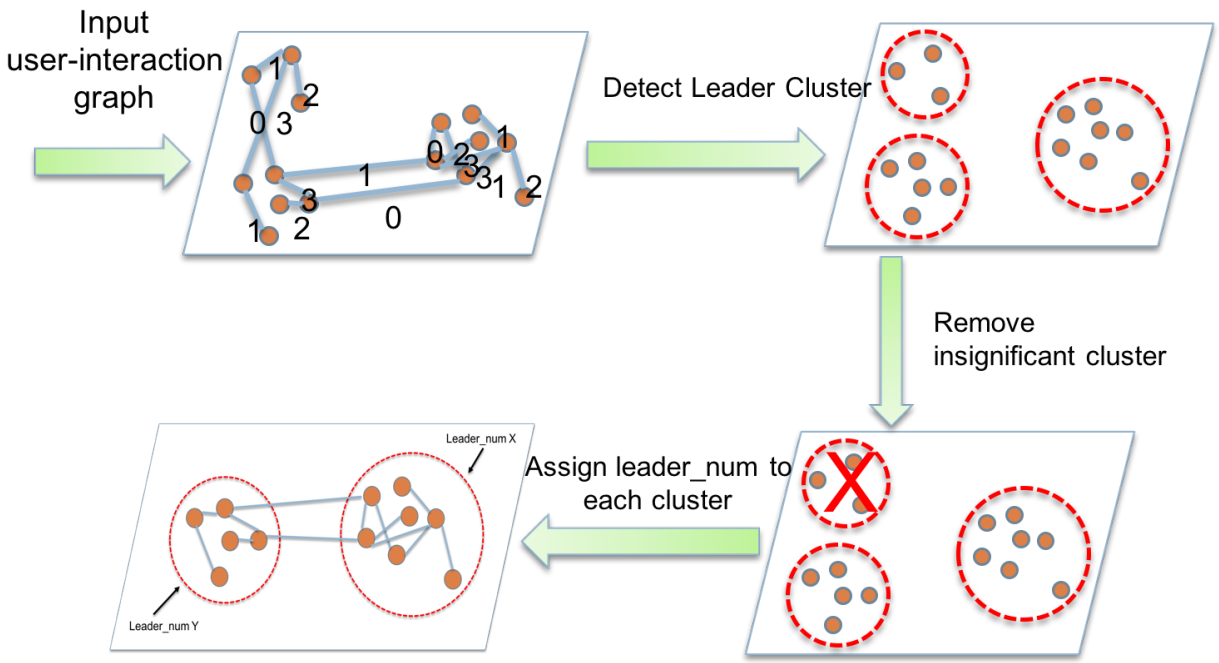

Figure 6.3: First-stage classification process

First, we have to define the similarity function. This function will calculate the similarity value based on two linking nodes' (users in our

case) interactions with their neighbours. The similarity value will be used as a weighting of the edge and is used to determine when to stop the classification process. In this research, we assume that a closer timing of users retweet/reply periods results in a higher impact between these users. Initially, the weight of each edge is based only on the retweet/reply period. By multiplying the similarity value by the retweet/reply period weighting, the total weighting is calculated and assigned to the edge.

Take a user-interaction graph $G = (V, E)$, where V is a set of users and E is a set of edges in the graph. If $(U_i, U_j) \in E$, $U_i, U_j \in U$ and $joint(U_i)$ represent all neighbouring nodes (users) of U_i (including U_i). The similarity function between U_i, U_j is shown at 6.1.

$$similarity(U_i, U_j) = \frac{|joint(U_i) \cap joint(U_j)|}{\sqrt{|joint(U_i)| \times |joint(U_j)|}} \times \frac{W_{U_i U_j}}{4} \tag{6.1}$$

After calculating the similarity value of all edges in G, we treat each node as an individual cluster. If the total weighting of the edge that connects between two nodes is larger than the total weighting of edges that connect to the node, then, the two clusters will be merged. For instance, edge e has the highest total weighting among all edges that connect to cluster A, and this edge connects to cluster B. Moreoever, if edge e also has the highest total weighting among all edges that connect to cluster B, then, we can merge cluster A and B into one cluster. Since the approach focuses on finding the connecting edges that have the largest total weighting, multiple nodes can be merged during the process. This process significantly improves the classification speed.

To determine when to stop the clustering process, we introduce an evaluation function EV to evaluate whether the process can be stopped. The evaluation function calculates the weight proportion sum of each cluster node's difference between the node in the same cluster and those not in the cluster. It indicates the overall similarity of nodes in the given cluster. The evaluation function is defined at 6.2:

$$EV(C) = \sum_{i=1}^{m} \left[\frac{AS_i}{SS} - (\frac{ES_i}{SS})^2 \right] \tag{6.2}$$

C is a set of clustered results where $C = \{c_i, c_2, \cdots, c_m\}$. AS_i is the sum of total weighting between nodes in c_i, and ES_i is the sum of total

weighting between nodes in c_i and the nodes in other clusters. AS_i and ES_i are defined in 6.3 and 6.4, respectively.

$$AS_i = \sum_{n,k \in C_i} similarity(n,k) \tag{6.3}$$

$$ES_i = \sum_{n \in C_i, k \in V} similarity(n,k) \tag{6.4}$$

SS represents the sum of total weighting between nodes in entire graph, which is shown in 6.6:

$$SS = \sum_{n,k \in V} similarity(n,k) \tag{6.5}$$

Finally, we can compare the values of the two clustering results by calculating the difference of each result's EV:

$$\Delta EV_{c \to c'} = EV(c') - EV(c) \tag{6.6}$$

and use $\Delta EV_{c \to c'}$ to decide whether to stop the clustering process. If $\Delta EV_{c \to c'}$ is negative, which means that the algorithm has already generated enough clusters and that it is the time to stop the process. When the clustering algorithm stops, two problems emerge. 1) Which clusters should be used to select the leaders? 2) How many leaders should be selected from each specific group?

For the first problem, when the clustering algorithm finishes the process, there will be two types of clusters, including 1) a cluster that has more than one node, and 2) a cluster that only has one node. In the case in which the cluster has only one user, two possible situations can happen.

1. This user is an isolated user who makes little contact with others. He is defined as an *outlier* and can be identified if they only connect to one cluster.

2. This user connects to multiple clusters. We define him as a *hotspot*. When it comes to selecting the leader, the *hotspot* will be selected first since this user may have a stronger impact on many clusters.

For the second problem, it will not be wise to select the leaders evenly from each cluster since some clusters may be less important, such as the *outlier* described above. If we can know which clusters are more significant than others, we can ignore insignificant clusters and select leaders only from significant clusters since leaders in these clusters can impact more users. Hence, we will select the significant clusters based on the given leader number K. If the number of users in a cluster is larger than *total users/K*, then, this cluster will be marked as a significant cluster. A set of significant clusters for a given user-interaction graph $G(V, E)$ can be defined as follows:

$$C_{sig} = \left\{ c_i \in C \middle| n_i \geq \frac{\sum_{j=1}^{m} n_j}{K} \right\} \tag{6.7}$$

where $C = \{c_1, c_2, \cdots, c_n\}$ is the clustered result of $G(V, E)$.

Then, we can select leaders from significant clusters based on the size of the clusters because in the common scenario, there is higher chance of having opinion leaders in the larger cluster. Additionally, selecting multiple opinion leaders in a large cluster can make more impact on users than in smaller clusters. Thus, we assign different numbers of leaders from each significant cluster based on the user size of total significant clusters. The number of selected leaders of each cluster is shown in equation 6.8:

$$leader\_num(c'_i) = K \times \frac{n'_i}{\sum_{j=1}^{m} n'_j} \tag{6.8}$$

where $C_{sig} = \{c'_1, c'_2, \cdots, c'_m\}$ and n'_i is the number of members in significant cluster c'_i.

6.2.2 Second-stage classification: Generating opinion leader candidates

At this stage, we apply a classification algorithm to each significant cluster to identify opinion leaders. In this research, each user is given a vector L where $L_i = (total\_tweets, t\_replied, pro\_deg, interact\_p)$. The definitions of each parameter are shown below:

- *total_tweets*: total number of tweets in specific topics, including replies and retweet
- *t_replied*: probability of being retweeted/replied to by others after tweeting
- *pro_deg*: expertise level of the user in a specific topic
- *interact_p*: probability of the user engaging with others

The reason that these parameters are chosen is listed below.

total_tweets: If a user is an opinion user, which means that his *total_tweets* must reach certain amount of tweets, he does not just tweet his opinion but also needs to interact with his followers, which can make him become a opinion leader. Hence, we take the total number of tweets in specific topics, including replies and retweets, into account.

t_replied: If a user is an opinion user, there will be a group of followers who will be impact by the leader. When the leader tweets, some followers will be impacted and try to retweet/reply to express their opinions. Thus, the probability of an opinion being retweeted/replied to is also selected as a parameter. This probability can be calculated by:

$$t\_replied = \frac{rt\_num}{tweet\_only\_num} \tag{6.9}$$

where rt_num is the number of tweets that are retweeted or replied to. Additionally, $tweet\_only\_num$ is the total number of a user's tweets, which does not include retweets or replies.

pro_deg: We believe that an opinion leader is only an expert in related topics. Hence, we introduce *pro_deg* into the vector. If a user's tweets in a specific cluster account for a large proportion of his total tweets, which means that he is an active user in the cluster, there is higher probability that this user has a greater interest or deeper research in a specific topic. Thus, he has certain profession level in this topic. *pro_deg* can be defined as:

$$pro\_deg = \frac{tweets\_topic}{total\_tweets} \tag{6.10}$$

interact_p: To be an opinion leader, the user needs to interact with his followers so he can keep his followers and attract new ones. As an

opinion leader, he will not only tweet or reply/retweet all the time. Normally, the rep_num should be larger than $tweet\_only\_num$. Hence, a user with higher $interact\_p$ is more likely to be an opinion leader. $interact\_p$ is calculated by:

$$interact\_p = \frac{rep\_num}{tweet\_only\_num} \tag{6.11}$$

where rep_num represents the total number of retweets/replies of the user.

As an opinion leader, the value of $total\_tweets$ should be moderate, and the value of $t\_replied$, $interact\_p$ and pro_deg will be high based on our observations. One possible explanation of the moderate value of $total\_tweets$ is that an opinion leader is normally an expert in a specific area and has a certain impact. Hence, he tweets his opinion carefully. Additionally, the reason that the value of $t\_replied$ is high is that average users usually tend to retweet/reply opinion leaders' tweets. The value of pro_deg indicates the level of expertise of a user, and this value should be high for the reasons above. Finally, to maintain his leadership, he needs to engage with his followers and try to attract new ones. Hence, the value of $interact\_p$ will be high as well. Once all parameters of the user vector are defined, we will apply k-means clustering algorithm to each cluster in C and get a set of candidates $CL = \{cl_1, cl_2, \cdots, cl_N\}$ where N is the number of clusters in C.

6.2.3 Selecting Opinion Leaders

When the second-stage classification process is finished, we will calculate the overall score of each cluster in CL and sort the scores to help us to select opinion leader candidates. The score function is defined below:

$$\begin{aligned} score(cl_i) = {} & ga\_total\_tweets \\ & \times ga\_t\_replied \\ & \times ga\_interact\_p \\ & \times ga\_pro\_deg \end{aligned} \tag{6.12}$$

$i = 1, 2, 3, \cdots, n$ where n is the number of clusters. The values of ga_total_tweets, $ga\_t\_replied$, $ga\_interact\_p$, and ga_pro_deg are defined as follows:

$$ga\_total\_tweets_i = \frac{\sum_{j=1}^{|cl_i|} total\_tweets_j}{|cl_i|} \tag{6.13}$$

$$ga\_t\_replied_i = \frac{\sum_{j=1}^{|cl_i|} t\_replied_j}{|cl_i|} \tag{6.14}$$

$$ga\_interact\_p_i = \frac{\sum_{j=1}^{|cl_i|} interact\_p_j}{|cl_i|} \tag{6.15}$$

$$ga\_pro\_deg_i = \frac{\sum_{j=1}^{|cl_i|} pro\_deg_j}{|cl_i|} \tag{6.16}$$

where $|cl_i|$ is the total number of users in cluster cl_i. Once the score of each cluster is sorted, we will select a number of users as the final opinion leaders from sorted clusters. The number of selected leaders is given by $leader\_num$, which is defined in equation 6.8. The selection process will have two steps: selecting from *hotspot* clusters and selecting from significant clusters.

First, we will select the leaders from *hotspot* clusters. In this case, $leader\_num$ will not be applied since users in *hotspot* clusters have greater impact than those in other clusters. We will try to identify as many users as opinion leaders as possible. For example, if the desired number of opinion leaders K is larger than the users in a *hotspot* cluster, all the users will be selected and $K = K - \left|cl_{hotspot}\right|$. If K is equal to the number of users of the *hotspot* cluster, then all users in the *hotspot* cluster are selected, and the selection process will be stopped. If K is smaller than the number of users in the *hotspot* cluster, all users in the cluster will be sorted by *degress*, which is the number of edges that connects to a user. Then, K users will be selected as opinion leaders. The selection procedure for *hotspot* clusters is shown in Algorithm 5.

If the value $remain\_Kr > 0$ from Algorithm 5, the second step will be applied. A cluster cr_i with the highest score will be selected from non-hotspot cluster will be selected. First, we will check if the pre-assigned $leader\_num_i$ is equal to or larger than $remain\_K$. If so, a number of $remain\_K$ users, who are ranked by degree, will be selected from cr_i, and the selection process will stop. If not, then only $leader\_num_i$, who are ranked by degree, will be selected and $remain\_K = remain\_K -$

$leader\_num_i$. The selection process will select the next highest score cluster cr_j until $remain\_K$ reaches 0. The second step of the selection procedure is shown in Algorithm 6.

Algorithm 5 Select Opinion Leaders from Hotspot Clusters

Input:

CH: a set of hotspot clusters from the result of second-stage classification

K: the desired number of opinion leaders

Output:

$OpList$: a set of selected opinion leaders

$remain\_K$: number of opinion leaders that still need to be selected

1: **for** each $ch_i \in CH$ **do**
2: calculate and assign the score to ch_i by $score(ch_i)$
3: **end for**
4: $CHS = sort\_by\_score(CH)$
5: **for** each $chs_i \in CHS$ **do**
6: **if** $|chs_i| > K$ **then**
7: $OpList \leftarrow$ select top K users, which has highest degrees $\in chs_i$
8: $remain\_K = 0$
9: END
10: **else if** $|chs_i| = K$ **then**
11: $OpList \leftarrow$ all users $\in chs_i$
12: $remain\_K = 0$
13: END
14: **else if** $|chs_i| < K$ **then**
15: $OpList \leftarrow$ all users $\in chs_i$
16: $remain\_K = K - |chs_i|$
17: **end if**
18: **end for**

Algorithm 6 Select Opinion Leader from Non-Hotspot Cluster

Input:

CN: a set of non-hotspot clusters from the result of second-stage classification

$remain\_K$: the desired number of opinion leaders

$OpList$: a set of selected opinion leaders from Algorithm 5

Output:

$OpList$: a set of selected opinion leaders

1: **for** each $cn_i \in CN$ **do**
2: calculate and assign the score to cn_i by $score(cn_i)$
3: **end for**
4: $CNS = sort\_by\_score(CN)$
5: **for** each $cns_i \in CNS$ **do**
6: **if** $leader\_num_i \geqslant remain\_K$ **then**
7: $OpList \leftarrow$ select top $remain\_K$ users, which has highest degrees $\in cns_i$
8: $remain\_K = 0$
9: **else if** $leader\_num_i < remain\_K$ **then**
10: $OpList \leftarrow$ select top $leader\_num_i$ users, which has highest degrees $\in cns_i$
11: $remain\_K = remain\_K - leader\_num_i$
12: **end if**
13: **if** $remain\_K = 0$ **then**
14: **Output** $OpList$
15: END
16: **end if**
17: **end for**

Chapter 7

Experiment and Evaluation

In Chapter 1, a set of claims were listed to describe the research questions of this thesis. Subsequently, an automated three-layered platform for topic group and user inclination clustering was proposed in Chapter 3 to support this thesis. In Chapters 4 and 5, two major layers of the platform were described. This chapter evaluates the accuracy of the two major layers in the proposed platform. To support the claims of this thesis, a set of experiments were planned and conducted to provide evidence.

7.1 Experiments on Topic Group Clustering

One of the objectives of this thesis is to cluster documents with similar topics from an on-line social network platform (Twitter) into groups. The proposed approach adapts and integrates several different complementary text clustering algorithms, while using an organised knowledge base (which is an abstraction and extraction of Wikipedia web pages). Two types of evaluations will be conducted: (1) ground truth-based and (2) survey-based evaluations. The first evaluation method provides us with the ability to automatically evaluate the generated results from a very large dataset, whereas the second method allows us to verify the generated results from the perspectives of a group of human experts.

7.1.1 Experiments

Three experiments were designed to evaluate the proposed topic group mining method. The three experiments were based on different strate-

gies that generate a set of classified topic groups from the same dataset. Then, an evaluation metric was provided and used to evaluate the performance of the generated topic groups. The details of each experiment are presented below:

- *[Experiment 1]* To evaluate our work, we needed a method for comparing our techniques. Therefore, prior to experimenting with our approach, we performed an experiment on a pure clustering algorithm without enrichment with Wikipedia knowledge. First, we pre-processed the collected tweets and modelled them with IF-IDF vectors. Then, we clustered them directly without any further pre-processing.

- *[Experiment 2]* In this method, we added a further pre-processing step with our tweet data by adding related Wikipedia topics, as explained in Strategy 1 in Section 4.7.1. Then, we modelled these enriched tweets with TF-IDF vectors prior to clustering.

- *[Experiment 3]* This method extended the topics from Strategy 2. This method adds not only Wikipedia topics that are related to each tweet in the pre-processing step but also Wikipedia categories of each Wikipedia topic into the tweets to solve the semantic relationship problem of the bag-of-words model as described in Section 4.7.2. Next, these pre-processed tweets were modelled using TF-IDF vectors. Finally, we clustered our data from these enriched tweets in the same way as in the two previous experiments.

7.1.2 Ground Truth-based Evaluation

To perform an evaluation based on our ground truth dataset, we manually labelled 400 tweets that consist of 20 tweets from 20 different topic groups. Then, we evaluated all three of the aforementioned methods by calculating the V-Measure score between true labels and labels that were assigned by a clustering algorithm. In this section, we first describe the test set that we used and then explain the details of our evaluation metric (V-Measure) in the following subsection.

Number of topics	20
Number of tweets	221
Number of terms	3712

Table 7.1: Statistics of ground truth dataset

7.1.2.1 Ground Truth Dataset

For the first step, to set up the clustering algorithm, we needed to identify the number of clusters. We manually selected 400 tweets from 20 groups as a ground truth dataset. The detailed statistics of the dataset are presented in Table 7.1. Then, we ran the clustering algorithm repeatedly with different numbers of clusters. By comparing the generated result to this dataset, we could determine the most appropriate number of topics.

7.1.2.2 Evaluation Metric

Typically, the basic criteria for a clustering result are homogeneity and completeness. The homogeneity criterion is satisfied. For all clusters, every member of each cluster comes from only one class that is defined as follows:

$$homogeneity = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{otherwise} \end{cases} \tag{7.1}$$

where $H(C|K)$ is the conditional entropy of the classes given assigned clusters and $H(C)$ is the entropy of the class and is defined as follows:

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{n_{c,k}}{n}\log\frac{n_{c,k}}{\sum_{c=1}^{|C|}n_{c,k}} \tag{7.2}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|}n_{c,k}}{n}\log\frac{\sum_{k=1}^{|K|}n_{c,k}}{n} \tag{7.3}$$

in which n is the total number of data points and $n_{c,k}$ is the number of data points from class c that clustered into cluster k.

The completeness criterion is quite the opposite of the homogeneity criterion. This criterion is satisfied if all members of a class are clustered into the same cluster. Mathematically, we can define the completeness

score as follows:

$$completeness = \begin{cases} 1 & \text{if } H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{otherwise} \end{cases} \tag{7.4}$$

where $H(K|C)$ is the conditional entropy of assigned clusters given the classes and $H(K)$ is the entropy of assigned clusters defined as follows:

$$H(K|C) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n}\log\frac{n_{c,k}}{\sum_{k=1}^{|K|}n_{c,k}} \tag{7.5}$$

$$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}n_{c,k}}{n}\log\frac{\sum_{c=1}^{|C|}n_{c,k}}{n} \tag{7.6}$$

in which n is the total number of data points and $n_{c,k}$ is the number of data points from class c that clustered into cluster k.

Note that if we focus on homogeneity, the completeness score will decrease simultaneously. For example, it is very easy to obtain a perfect score by clustering each data point in an individual group. In contrast, the homogeneity will be reduced if we focus our efforts only on the completeness score. For example, we can set the clustered group number to 1 to satisfy this criterion. Hence, the true quality of a clustering result will not be revealed if we only consider one of these metrics.

A good clustering result should simultaneously satisfy both homogeneity and completeness. For this purpose, we used the V-Measure as a metric for evaluating the clustering results. V-Measure, which was first introduced by [Rosenberg and Hirschberg, 2007], is the harmonic mean of the homogeneity and completeness scores bounded within the range of [0, 1]. Values closer to 1 indicate better clustering results. This measure can be defined as follows:

$$V_\beta = \frac{(1+\beta)\times h \times c}{(\beta \times h) + c} \tag{7.7}$$

where β is the weight, and if β is set to less than 1, the homogeneity is given a greater weight. If β is set to greater than 1, the completeness is given a greater weight. In our experiment, we weight them equally by setting $\beta = 1$ as suggested in [Rosenberg and Hirschberg, 2007].

7.1.3 Results

The data given to each experiment consist of a fixed dataset containing approximately 1.2 million public tweets collected from Twitter. For each experiment, we performed bisecting k-means multiple times with different settings of the number of k clusters, ranging from 64 to 4160 (increase by 256). Next, we evaluated the clustering results with our test set using the V-Measure as an evaluation metric. Figure 7.1 shows the V-Measure scores for experiments 1, 2 and 3 with different settings for the number of k-clusters.

Figure 7.1 clearly demonstrates that the method used in experiment 2 (topics) and the method used in experiment 3 (topics + categories) outperformed the method used in experiment 1 (baseline). The second experiment is clearly better than the first method for every setting of the number of k clusters. The highest V-Measure that method 1 can obtain is 0.674 at $k = 3904$, whereas the highest V-Measure of experiment 2 is 0.747 at $k = 3392$. The difference between the highest peak for each is 7.3%.

Furthermore, experiment 3 obtains a clearly higher performance than does experiment 1, as shown in Figure 1. Compared with the best performance of the other methods, the performance of experiment 3 increases by 14.7% with a V-Measure of 0.821 at $k = 3648$. Using these results obtained from our test set, we can conclude that Experiments 2 and 3 exhibit dramatic improvements over method 1, with V-Measures of 0.747, 0.821 and 0.674. These results confirm that using Wikipedia as a resource for enriching tweets can improve performance in topic group mining.

7.1.4 Evaluation Based on Survey

Evaluating the results using computer-based metrics may not be sufficiently reliable because humans might have different perspectives regarding the result. To thoroughly evaluate the results, we employ 10 people who do not have any bias towards the selected topics from the generated topic groups. These people will determine whether a tweet is correctly classified in the specific topic group. They are asked to rate the

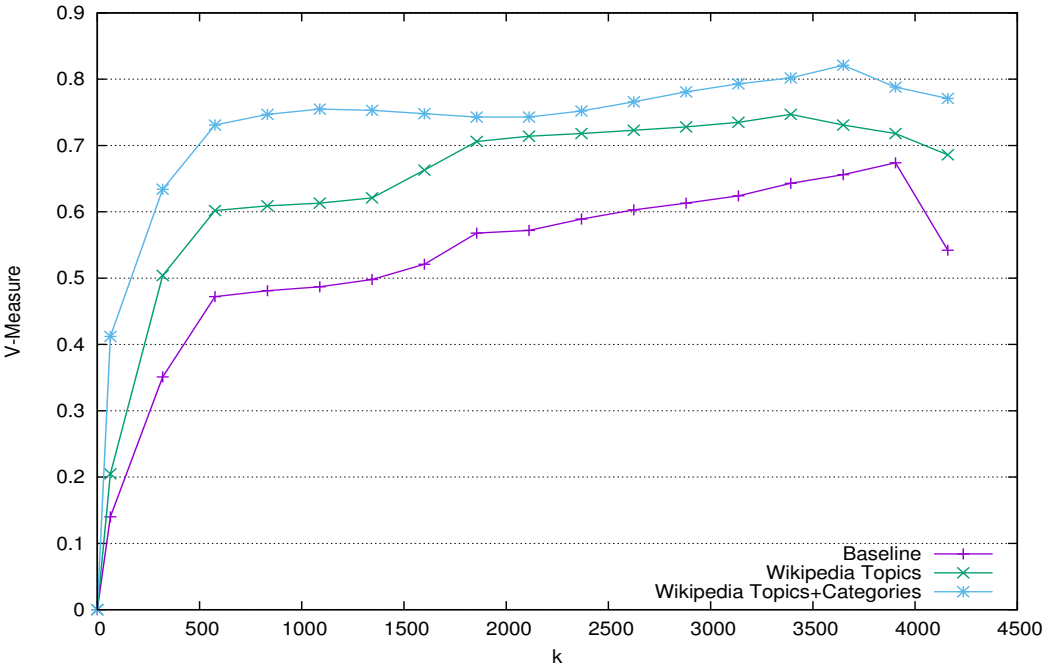

Figure 7.1: Comparison of the results between Experiments 1,2 and 3

degree of relevance of each tweet by score. The explanation of the score is as follows:

- Score 1: not relevant to the topic at all
- Score 2: may be relevant / I am not sure
- Score 3: slightly relevant
- Score 4: relevant
- Score 5: very relevant

Because there are numerous generated topic groups, we randomly select 5 non-overlapping groups for each person to evaluate. For the content of the topic group, we randomly select 10 tweets from each group as our evaluation data. Namely, each human examiner will have 50 tweets from each experiment. In total, there are 500 tweets from 50 different topic groups for each experiment that will be scored.

7.1.5 Survey-based Evaluation Results

The results of the baseline method (method 1) are shown in Figure 7.2. Based on the survey results, only 13% of the tweets are scored as very relevant, and 28% are scored as relevant. A total of 34% of the tweets are classified as slightly relevant to their topic, and 13% of the tweets are difficult for the human examiner to determine weather they are related to the topic. Notably, 12% of the tweets were considered to be not at all relevant to the topic.

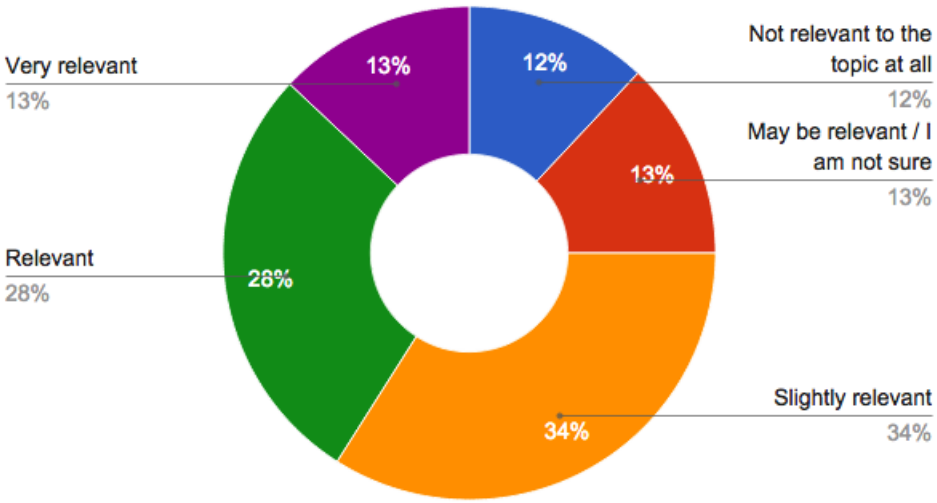

Figure 7.2: Human evaluation results of baseline algorithm (method 1)

Moreover, Figure 7.3 indicates a dramatic increase in the relevant tweets (including slightly relevant, relevant and very relevant) from 75% in method 1 to 89%. A total of 42% of the tweets give human examiners strong confidence in classifying them as very relevant. Furthermore, 22% of the tweets have a lower confidence of being scored as relevant, and 25% of the tweets are classified as slightly relevant. Note that the number of not relevant or potentially relevant tweets significantly decreases from 25% in method 1 to 11%, and only 4% of the tweets are identified as not at all relevant.

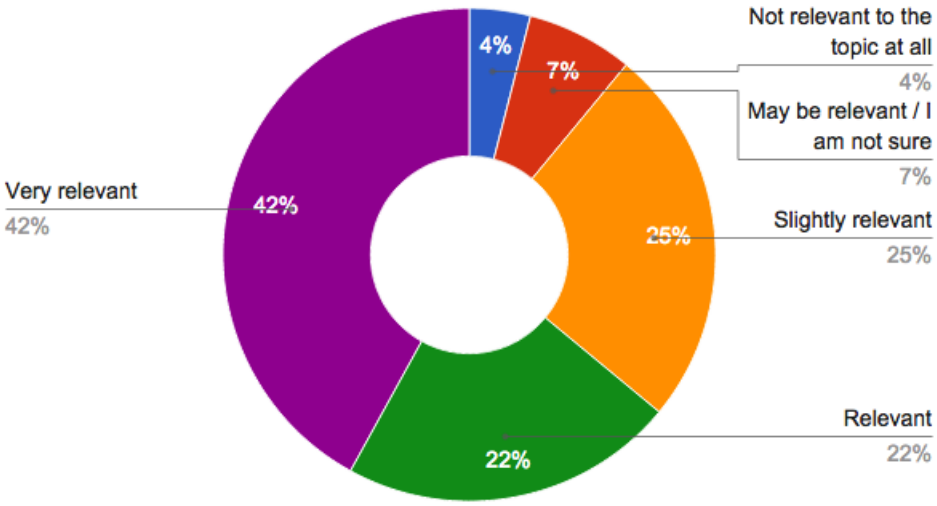

Figure 7.3: Human evaluation results of baseline+Wikipedia topic algorithm (method 2)

Finally, the results of method 3 are shown in Figure 7.4. The improvement between method 3 and method 1 (baseline) is compelling. The very relevant ratio is 30% higher and 14% lower than the not relevant or potentially relevant groups, respectively. However, if we compare the results of method 2 and method 3, there is only a 3% increase in the relevant group and a 1% improvement in the very relevant group. The survey also reveals that 43% of tweets are very relevant and that 20% are relevant. However, the number of tweets that are classified as not relevant increases by 1%, but the difference in these changes is not highly significant between either method. Based on the survey results, we can claim that both proposed methods 2 and 3 perform well in the experiments.

7.2 Analysis of Topic Group Clustering

There are two main reasons for the improvement in the results using the two approaches based on enriching the original documents using Wikipedia. First, adding Wikipedia topics has a positive effect on the TF-IDF model and can overcome the problem that this model suffers from concerning short text documents, as mentioned in previous sec-

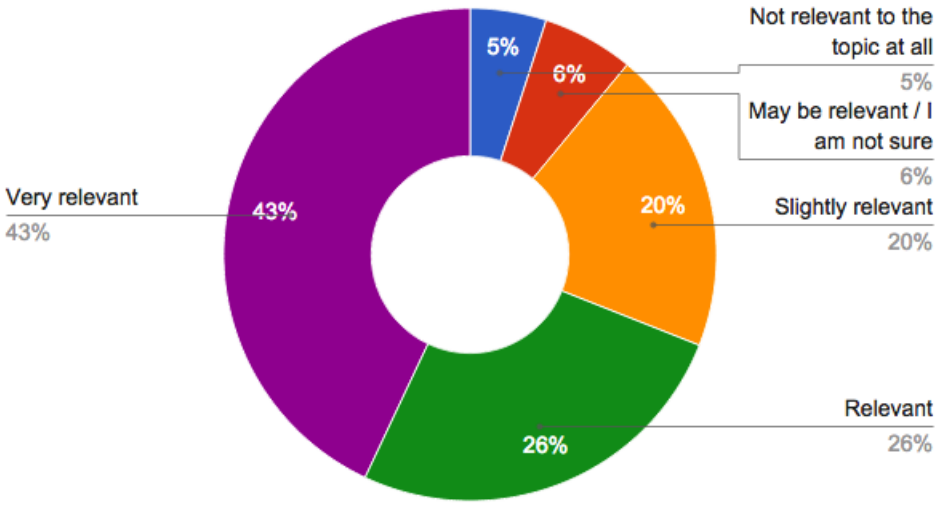

Figure 7.4: Human evaluation results of baseline+Wikipedia Topic+Wikipedia Categories algorithm (method 3)

tions. Second, the third approach provided the best results because adding Wikipedia categories overcomes one of the pitfalls of the TF-IDF model, namely, the problem of finding semantic relatedness between terms in documents. Consequently, the improvement in the TF-IDF model results in better clustering performance.

Furthermore, given that the background knowledge is already in place (i.e., the extraction and index from the Wikipedia pages), it takes a total of 22 to 32 hours (depending on the different document enrichment strategies) to analyse 1.2 million tweets on an Intel i7 processor machine with 16 GB memory. This approach is therefore relatively efficient and suitable when employed to analyse large amounts of short-text communication. This approach can also be processed in parallel. For example, to process all collected tweets in 1 year (200+ million tweets) on a 20-node cluster, it will take only approximately two weeks to complete the analysis.

7.2.1 Effect of Short-text Document Enrichment on TF-IDF

The enrichment of the document is the main reason that approaches 2 and 3 can overcome the problem faced by the bag-of-words model. In general, baseline methods cannot reflect the true importance of terms, as

discussed in Chapter 3. We now present an example from the real dataset. In our corpus, the word "Twitter" appears in 91,821 tweets, whereas the word "week" only appears in 4,372 tweets. This result means that the word "week" received a higher TF-IDF weight than the word "Twitter", which is not appropriate. Consequently, the baseline method provided the worst results in both our test set and survey because the tweets are represented with an inappropriate model.

Consider that the tweet listed below is clustered incorrectly by the baseline method and that we can can enrich it by adding Wikipedia topic(s) to make it cluster into the correct group.

My week on Twitter: 15 new followers, 4 RTs, 2 faves, 1 sore thumb and no life

The TF-IDF vector shown below is the vector after it has been transformed into the TF-IDF model.

$$DOC_{base} = (life : 0.27, thumb : 0.36, follower : 0.21, \textbf{week: 0.34}, \\ sore : 0.27, new : 0.24, \textbf{twitter: 0.21}) \tag{7.8}$$

Clearly, the main topic of this tweet is the user's life on Twitter. The terms "Twitter" and "life" should be considered as important words. However, the term that has the highest TF-IDF weight in the baseline method is "week". We then enrich the original tweet with the Wikipedia topic "Twitter", and the term "Twitter" immediately becomes the highest weighted term, as shown below.

$$DOC_{enrich} = (life : 0.27, thumb : 0.36, follower : 0.21, \textbf{week: 0.31}, \\ sore : 0.27, new : 0.24, \textbf{twitter: 0.38}) \tag{7.9}$$

This improvement in the model results in a higher-quality clustering process because the clustering algorithm that we used, k-means clustering, attempts to cluster similar tweets together. Consequently, with the TF-IDF vector of the baseline method, the k-means algorithm will group

	$centroid_{Twitter}$	$centroid_{week}$
DOC_{base}	0.16	0.57
DOC_{topic}	0.41	0.21

Table 7.2: Cosine distances between centroids and TF-IDF vectors

a tweet into a group in which the word "week" is important. However, with the TF-IDF vector of the other two methods that used Wikipedia knowledge, they will group this tweet into a group in which the word "Twitter" is important. In other words, k-means clustering attempts to assign a tweet into the closest group, namely, the group with the centroid closest to the tweet in the vector space.

For example, given two centroids of the terms "Twitter" and "week",

$$centroid_{Twitter} = 0.513$$

$$centroid_{week} = 0.507$$

The cosine distance between these centroids and document vector, DOC_{base} and DOC_{topic}, is shown in Table 7.2. This indicates that the incorrect TF-IDF weighting calculated by the baseline method clearly affects the cosine distance. As shown in Table 7.2, DOC_{base} is closer to $centroid_{week}$ rather than to $centroid_{Twitter}$. By applying the enrichment method, DOC_{topic} is now closer to $centroid_{Twitter}$ than to the incorrect $centroid_{week}$. The clustering algorithm can produce more meaningful results using only the distance between the tweet and centroid. Consequently, the results produced by methods that enrich tweets with Wikipedia topics are significantly better than those produced by the baseline method.

7.2.2 Effect of Short-text Document Enrichment with Semantic Relationships

The method of enriching documents with Wikipedia categories yielded the best V-Measure score, as shown in Figure 7.1. This method shows a considerable improvement over the baseline approach and is slightly better than enriching with only Wikipedia topics. The reason behind this

performance enhancement is that enriching documents with Wikipedia categories overcomes the problem of semantic relatedness in the TF-IDF model. In the TF-IDF model, we generally use the cosine distance to describe the relatedness between documents. A greater distance between two documents means less similarity and lower relatedness. Without adding these categories, the cosine distance is unable to reflect the relationships between any two semantically related terms. In this case, the distance between them is the upper bound of the cosine distance, which is 1. However, after adding the categories, the two semantically related terms become closer in the TF-IDF vector space in terms of cosine distance.

Two tweets that do not have any overlapping terms are given as an example:

$Tweet_1 =$ "We want to make sure your Flickr Pro service is uninterrupted" lol cool attempt to get more $ out of me,yahoo (flickr pro no longer exists)

$Tweet_2 =$ "Do you know how awesome SmugMug is? Service so good," they even share live status reports.

It is clear that $Tweet_1$ and $Tweet_2$ are discussing the "Flickr" and "SmugMug" services. Both of these services are popular on-line image services. However, the semantic relationship between them is impossible to identify using the baseline approach. The cosine distance between these tweets should theoretically be 1 using the baseline approach.

In the third approach, "photography website", which is one of the common categories that contains both topics in Wikipedia, is added. Adding the common category can create a semantic relationship between two tweets. From the TF-IDF vector perspective, the cosine distance between the two tweets is 0.531. The distance is considerably closer than that obtained using the baseline approach.

Topic	Users	Tweets	Following edges
Giants	15,199	492	4,873
Tigers	12,933	361	3,674

Table 7.3: Statistics of the two datasets

7.3 Experiments on User Inclination Analysis

The user inclination analysis is another key component of the proposed framework, and it will be evaluated and analysed in this section. First, we evaluate and compare the precision rates of the proposed document inclination method and the baseline method using two different sentiment dictionaries. Then, we will demonstrate how the user relationship can improve and correct the generated results.

7.3.1 Data Collection

Two subsets of data were selected from clustered "baseball" topic groups. The data range between 14 October, 2012, and 5 November, 2012, and the 2012 World Series was held between 24 and 28 October. The first subset is "Tigers", and the second subset is "Giants". The total size of the collected data is 28,132 tweets published by 853 users on Twitter. We also take the user relationship into account, namely, "following" in our case. The details of both subsets are presented in Table 7.3.

7.3.2 Document Inclination Classification

In our experiments, two lexicon dictionaries were used: 1) the MPQA subjective lexicon from the University of Pittsburgh, which contains 2,718 positive terms and 4,912 negative terms, and 2) the Sentiment Lexicon from the University of Illinois at Chicago (UIC), which includes 2,041 positive terms and 4,818 negative terms. To simplify the computation, we also add the logarithm of the probabilities rather than multiplying them. The most probable class will still be the one with the highest probability after the logarithm. Therefore, we use the equation shown in 7.10, which is widely used in most naïve Bayes implementations.

Applied Method	Precision
UIC - Multinomial Bayes	81.72%
UIC - Bernoulli Bayes	82.31%
MPQA - Multinomial Bayes	79.17%
MPQA - Bernoulli Bayes	79.44%
Bag-Of-Word SVM	73.76%

Table 7.4: Performance of inclination classification of "Giants" dataset

$$C_{map} = \arg\max_{c \in \mathcal{C}} P(c|D) = \arg\max_{c \in \mathcal{C}} [\log P(c) + \sum_{1 \leq i \leq n} P(t_i|c)] \qquad (7.10)$$

, where C = {Unknown/Neutral, Positive, Negative}

The classification procedure will be performed as follows. First, we segment each tweet and extract all subjective terms to generate a vector model. In other words, the terms that are taken into consideration are those that appear in the lexicon dictionary. Then, we apply both Bernoulli and multinomial naïve Bayes approaches to the model. We can classify the inclination of each tweet by calculating the maximum posterior probability. However, the calculated probability will be very low or zero when the data are sparse. Hence, we use Laplace smoothing to avoid this situation. The results are shown in Tables 7.4 and 7.5. Every tweet is labelled as unknown/neutral, positive or negative after classification. The precision of inclination classification can be as high as approximately 82%. In contrast, the traditional bag-of-words SVM model does not perform well, with a precision of only approximately 73%. Furthermore, certain tweets could not be identified due to the lack of textual features found in short tweets, which is the natural limitation of Twitter.

We found that the UIC lexicon dictionary outperformed the MPQA lexicon dictionary on both datasets. We suspect that this result might be a consequence of the different corpus sources used by these dictionaries. The UIC dictionary uses on-line forums, posts and reviews, whereas MPQA uses regular documents. The nature of the UIC dictionary is closer to our dataset, which is also retrieved from the Internet. Therefore,

Applied Method	Precision
UIC - Multinomial Bayes	82.27%
UIC - Bernoulli Bayes	83.85%
MPQA - Multinomial Bayes	81.11%
MPQA - Bernoulli Bayes	81.49%
Bag-Of-Words SVM	72.16%

Table 7.5: Performance of inclination classification of "Tigers" dataset

this product produces a slightly higher precision than does the MPQA dictionary.

Note that the inclination classification of tweets in this stage does not represent the overall user inclination towards the topics but rather only labels each tweet with its inclination. To take advantage of the user relationship analysis in the next section, we have to transform the document-level inclination classification into user-level inclination classification. To achieve this goal, we will calculate all of the positive, negative and unknown/neutral inclination probabilities of each user. All probabilities of tweets that are published by the same user will be summed by inclination and divided by the number of tweets for each inclination. Then, the inclination with the highest probability will be considered as the user's inclination towards the topic.

7.3.3 User Relationship Analysis

The only intra-user relationship on the Twitter platform is "following". News agencies and celebrities that have many followers may have a stronger influences than regular users. However, we still need to experimentally verify our assumption. Additionally, there are three types of user-short-text-document relationships that need to be simultaneously considered based on the interaction frequency, which is called the interaction degree, as mentioned in the previous chapter. However, the interaction degree can only show the closeness degree of two users as opposed to similar inclination degrees. For example, users u_i and u_j have a high interaction degree between them, but this could be caused by a situation whereby whenever u_i tweets, u_j always responds with an op-

posite opinion. Therefore, we want to identify the importance of each user-short-text-document relationship such that we can apply different weights to each relationship.

There are three user-short-text-document relationships in Twitter: reply, retweet and favourites. Therefore, the interaction degree among two users can be formulated as shown in Equation 7.11.

$$w(u_i, u_j) = \beta_1 \frac{reply(u_i, u_j)}{reply(u_i, u_{others})} + \beta_2 \frac{retweet(u_i, u_j)}{retweet(u_i, u_{others})} + \beta_3 \frac{favourite(u_i, u_j)}{favourite(u_i, u_{others})} \tag{7.11}$$

, where $0 \leq \beta_1, \beta_2, \beta_3 \leq 1$ and $\beta_1 + \beta_2 + \beta_3 = 1$

We then examine the relation between inclination and three user-short-text-document relationships on Twitter. We choose 200 users and observe their interaction degree $w(u_i, u_j)$. Figure 7.5 shows that *retweets* consistently result in the same inclination. Namely, if user u_i *retweets* u_j's tweet, there is a high probability that u_i shares the same inclination with u_j. Conversely, *reply* and *favourites* do not demonstrate consistent inclinations by the user. In this thesis, we only focus on the *reply* frequency between two users; the inclination of the replied content will be ignored. Namely, if user u_i replies to u_j's tweet with an opposite opinion, it will still be calculated in the interaction degree. According to the data we selected, I have decided and assumed that the *reply* relationship is less important than *retweet*, whereas *favourites* relationships vary in groups with different interaction degrees. Hence, the values of β_1, β_2 and β_3 will be ordered as $\beta_1 > \beta_2 > \beta_3$.

7.4 Discussion of User Inclination Analysis

7.4.1 Topic-related Document Selection

To adjust the dependency between the given topics and selected documents, we introduce the SO-PMI mentioned in Chapter 2 to this thesis.

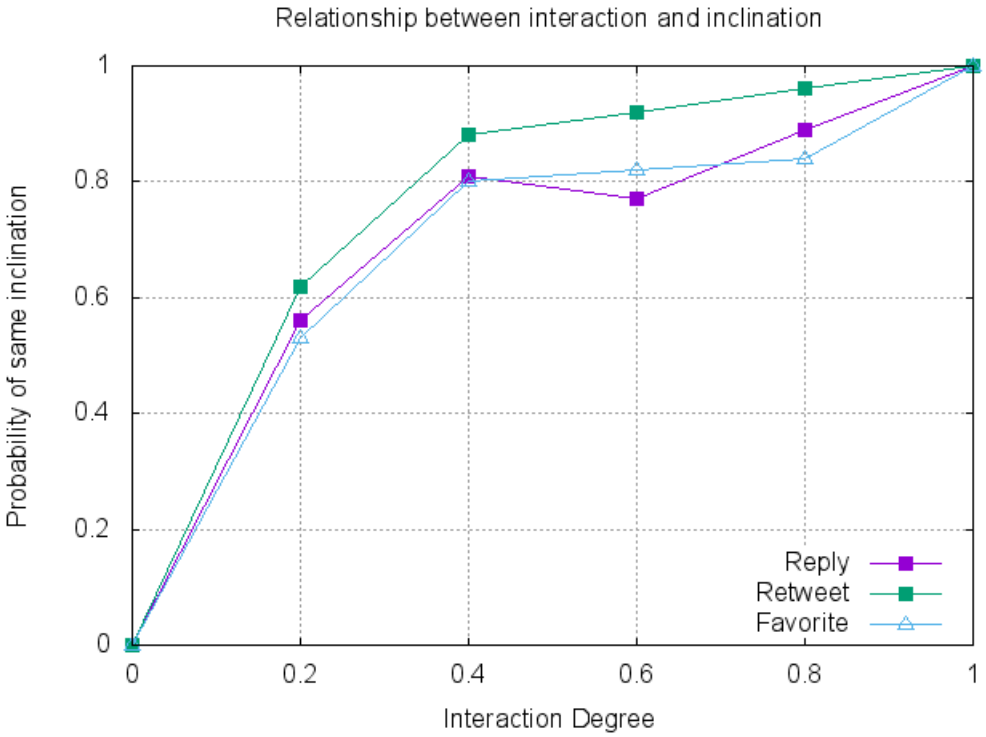

Figure 7.5: Consistency of inclination on user interactions among 200 users

The assumption is that two highly related terms will have a high co-occurrence probability. First, we calculate the PMI of all topics of related tweets towards the topics, which in our case are "Giants" and "Tigers", as PMI(rt, "Giants") and PMI(rt, "Tigers"). Then, the related tweets with a PMI smaller than 0.01 will be considered irrelevant and ignored. Finally, we can calculate the SO-PMI to determine the relatedness between the related topic and given topics (Tigers and Giants), as shown in Equation 7.12.

$$SO-PMI(rt) = PMI(rt, \text{“}Giants\text{”}) - PMI(rt, \text{“}Tigers\text{”}) \qquad (7.12)$$

, where rt is related topic

Table 7.6 shows the similarity of the related topic and given topics. The table clearly indicates that 5 topics ("San Francisco", "AT&T", "Lou Seal", "Bruce Bochy" and "Champion") are related to topic *Giants,* whereas 5 related topics are related to topic *Tigers* ("Detroit", "Miguel Cabrera", "Paws", "Jim Leyland" and "Prey"). All selected topics using

Related Topic	PMI(rt, "Giants")	PMI(rt, "Tigers")	SO-PMI(rt)
San Francisco	0.284024	0.164352	0.119672
Detroit	0.053524	0.074934	-0.021410
AT&T Park	0.042319	0.017432	0.024887
Miguel Cabrera	0.003942	0.049542	-0.045600
Paws	0.001137	0.004377	-0.003240
Lou Seal	0.007436	0.001642	0.005794
Fox	0.048194	0.034965	0.013229
Bruce Bochy	0.163531	0.064593	0.098938
Jim Leyland	0.054391	0.132313	-0.077922
Prey	0.054952	0.310320	-0.255368
Champion	0.359241	0.035256	0.323985

Table 7.6: SO-PMI similarity between related topics and given topics (Giants and Tigers)

SO-PMI are highly related to the given topics with the exception of *Fox.*

7.4.2 User Relationship Analysis

In this thesis, we introduce user relationship as another analysis factor for potential use in inclination mining. In this section, we present some examples that demonstrate that inter-user-level inclination mining outperforms user-short-text-document level inclination mining, as shown in Table 7.7. Positive, negative and unknown/neutral are represented by the symbols "+", "−" and "/", respectively. For example, the tweet of user 1 contains the negative term "sad". If the text content of the tweet alone is considered, it will be classified as a negative tweet by the training dataset and marked as "−". However, the tweet is actually positive regarding the given topic. In addition, the tweet of user 2 includes both given topics. If we only want to know the inclination towards a specific topic, e.g., Tigers, we can utilise the user relationship to identify the inclination of the tweet. With the inter-user relationship classification, we can see that user 2 exhibits positive sentiments towards the Tigers and negative sentiments towards the Giants.

In particular, classifying the inclinations of the tweets of user 3, user

4 and user 5 towards the topic is difficult due to the use of metaphors or sarcasm in the tweets. The text itself appears to indicate a negative inclination, but the actual hidden inclination of the tweet is positive. Notably, the tweet of user 6 does not contain any inclination terms in the content. The tweet will be classified as unknown/neutral by the text classification. By examining the user relationships and other tweets of user 6, we can determine that user 6's tweet has a negative inclination towards the topic. To overcome the aforementioned problem, our proposed method takes the inter-user and user-short-text-document relationships between two users into consideration. We are able to adjust the inclination of the tweet by utilising both relationships.

User	Tweet	Ground Truth	Inter-User Analysis	Inter-User+User-Short-Text-Document
1	World Series tonight & no one to watch it with :(#Sad but GO TIGERS :)	+	+	−
2	Congrats Giants! You slaughter Tigers bcuz of the pure luck! I won't see you next year!	+	+	−
3	Good for you, Tigers! Save me tons of money for just watching the game on TV.	−	−	+
4	I can't believe my mom still believe in Tigers?	−	−	+
5	Tigers just proves they are as good as Peanut's team!!	−	−	+
6	New member of college baseball: Detroit Tigers	−	−	/

Table 7.7: Example of inclination classification for "Tigers" topic

7.5 Experiments on Opinion Leader Identification Framework

In this section, we would like to evaluate the proposed opinion leader identification framework on two different use cases. In first use case, we pick a specific topic for experiment. Then, the generated result from the proposed framework will be compared to Twitter's built-in search function. In the second use case, we will create a mock business scenario that can help us to evaluate if proposed framework can assist commercial user externally.

7.5.1 Use Case: Opinion Leaders for "Headphones"

We select "headphones" as the topic for our experiment. The goal of the experiment is to find a set of opinion leaders that are familiar with headphones. We expect the selected opinion leaders to be knowledgeable about headphone attributes, such as functionality, sound quality, price and related technology.

Luckily, we had already collected a large number of tweets, which are tagged with related topics by proposed algorithm as described in the previous chapter. These data also come with metadata such as author and tweet timestamps. Thus, we will utilise this resource and select the tweets that are tagged with "headphones" from our database as our experimental dataset. Overall, the experimental dataset contains 164,719 tweets posted by 67,294 users.

7.5.1.1 Tweet Filtering and Cleaning

As described at the beginning of this chapter, all tweets are directly collected from Twitter. We need to perform data filtering and cleaning on the experimental dataset to produce useful data for analysis. The user and his tweets that meet the following criteria will be removed from the experimental dataset.

- **Private User**: A user on Twitter can set his timeline as "protected", which means that only his followers are allowed to view the

timeline. This status makes it impossible to obtain the srequired attributes to determine whether this user is an opinion leader.

- **Inactive User**: A user who has fewer than 100 tweets means that he is not every active on Twitter. To be an opinion leader, a user needs to actively interact with his followers on Twitter.
- **Low-impact User**: A user whose tweets have been retweeted less than two times on average. This status indicates that this user does not have a lot of impact on other users.
- **Low-profile User**: A user whose number of Twitter accounts followed is more than double the number of followers is considered "low-profile". This status shows that he tends to obtain information rather than express his opinion.

A total of 63,314 users and their tweets are removed after the filtering and cleaning process. Interestingly, approximately 75% of the removed users were filtered out by the **Low-impact User** status. The average tweet count of users in the experimental dataset is illustrated in Figure 7.6.

7.5.1.2 Result of the Proposed Framework

In this experiment, we set the number of opinion leaders to be identified as 50. First, the processed experimental dataset is fed into the opinion leader identification framework described on Chapter 6. Then, we use Twitter's built-in search engine to obtain the top 50 users in the "headphone" topic. Finally, the result of the proposed framework and Twitter built-in search engine are compared.

After 50 opinion leaders are generated by the leader identification framework, we examine the results before comparing them to those of Twitter's built-in search engine. Table 7.8 illustrates the basic attributes of the top 10 generated opinion leaders on "headphones."

Then, we check the opinion leaders' profiles listed in Table 7.8 manually, and a brief introduction of each leader is presented below.

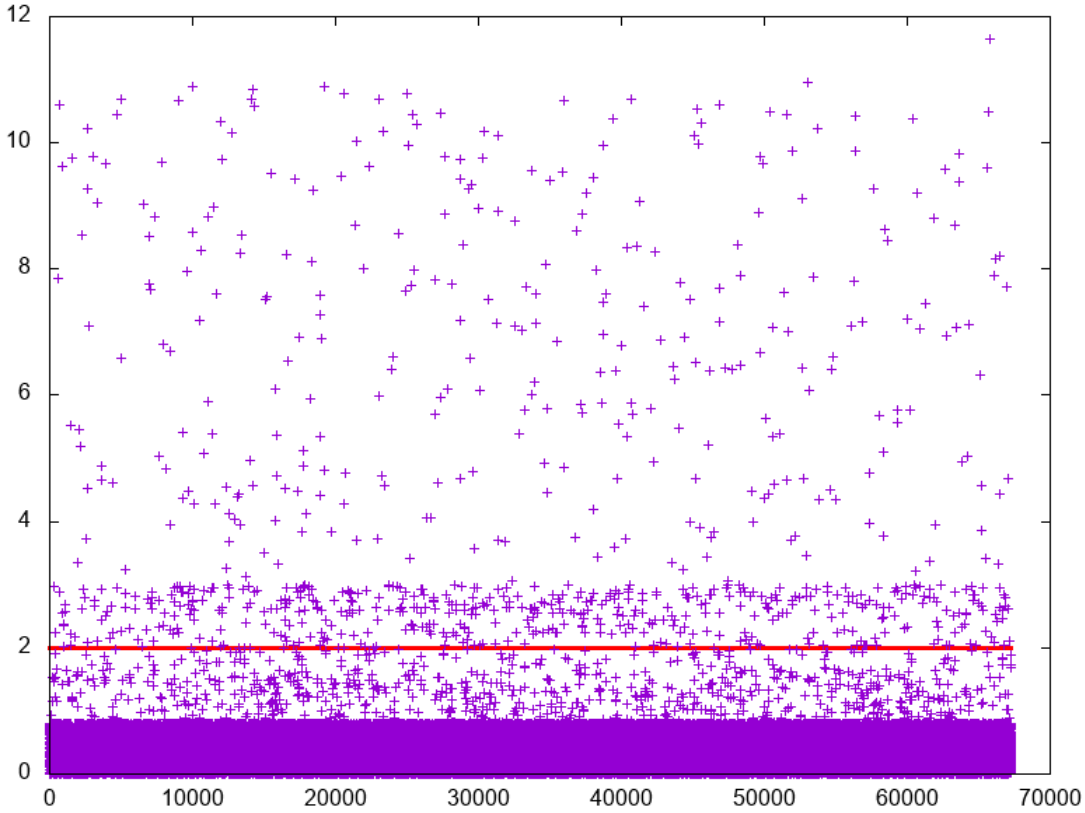

Figure 7.6: The average retweet per user in the experimental dataset

- **mashable**: This verified account belongs to a large tech-related news website (http://mashable.com/) and has more than 7 million followers on Twitter. This account has a powerful impact on its followers. This account reports on headphones, especially high-tech ones, as well as news and reviews from time to time.

- **SennheiserUSA**: This is a verified account that represents one of the major headphones manufacturers that produces high-quality products embedded with new technologies such as high-res bluetooth transmission and noise reduction.

- **verge**: This is a verified account that belongs to a famous consumer reviews website (https://www.theverge.com/). These consumer reviews are known to be simple but easy to understand, so this account is well-liked by many readers and has a great impact on them.

- **OPPODIGITAL**: This account is operated by OPPO Digital,

Rank	ID	Avg_Retweet Tweet	Followings	Followers
1	mashable	33.82	2,434	7,303,012
2	SennheiserUSA	7.78	1,240	120,931
3	verge	45.30	120	1,843,131
4	OPPODIGITAL	2.57	318	6,013
5	PCMag	6.19	1,317	313,924
6	whathifi	7.48	395	29,137
7	techradar	4.11	291	172,421
8	smsaudio	3.41	1	49,821
9	AVMag	3.92	1,709	17,530
10	HDtracks	2.18	1,309	8,212

Table 7.8: Top 10 "headphones" opinion leaders generated by the proposed framework

which mainly builds and sells blu-ray players. It only has a limited relationship with headphones.

- **PCMag**: This account is owned by an established magazine, "PC Magazine". Its product reviews have had a very powerful impact on its readership since the paper-based era. The same kind of impact remains in the on-line environment.

- **whathifi**: This account belongs to a professional Hi-Fi-focused website (https://www.whathifi.com/). All of its followers can be considered Hi-Fi enthusiastic or interested.

- **techradar**: This verified account is owned by a UK-based news/review website (http://www.techradar.com/). It has a dedicated section for introducing and reviewing audio/visual-related products.

- **smsaudio**: This account is owned by the trending headphones brand "SMS Headphones". It was created by Curtis "50 Cent" Jackson and is popular and well liked by teenagers.

- **AVMag**: This account belongs to a UK-based publisher "AV Magazine", which has been established for over 40 years. It focuses on reporting audio/visual-related news to its readers.

11 True Positives (TP)	39 False Negatives (FN)
39 False Positives (FP)	0 True Negatives (TN)

Table 7.9: Top 10 "headphone" opinion leaders generated by the proposed framework

- **HDtracks**: This account is owned by an on-line HD music/video store, which is not very related to the "headphones" topic.

As shown above, most of the top 10 opinion leaders selected by the proposed framework in the "headphones" topics do have a significant impact on their followers except for two accounts: OPPODIGITAL and HDtracks. The account "OPPODIGITAL" contains a large amount of information about their Blu-ray product, whereas the account "HDtracks" mainly sells high-quality music and video. Both accounts rarely mention anything about "headphones".

7.5.1.3 Comparison of Results with Twitter

We compare the top 50 opinion leaders that are selected by our framework and Twitter search engine in this section. Selecting the top K related users for the specific topic, "headphones" in our case, is easy. The term is simply entered into Twitter's search bar, and the "People" tab is clicked on the results page. In this experiment, we define the result generated by the proposed framework as the prediction set and Twitter result as the truth set. The confusion matrix of the results is shown in Table 7.9.

where

- True Positive: The user selected as an opinion leader by the proposed framework also appears in Twitter's result.
- True Negative: The user marked as a regular user is really a regular user. In our case, all selected users are considered as leaders, and thus, the value will be 0.
- False Positive: The user selected as an opinion leader by the proposed framework does not appear in Twitter's results.

- False Negative: The user selected as an opinion leader by Twitter does not appear in the result of the proposed framework.

Then, we calculate the recall, precision and F-measure based on Table 7.9.

$$Recall = \frac{TP}{FN+TP} = \frac{11}{11+39} = \frac{8}{50} = 0.22$$

$$Precision = \frac{TP}{FP+TP} = \frac{11}{11+39} = \frac{8}{50} = 0.22$$

$$F-measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{0.22 \times 0.2}{0.22+0.22} = 0.22$$

The value of the F-measure, 0.22, indicates that the difference between the two results is significant. Only 11 out of 50 users were successfully identified as opinion leaders (if we believe that the result from Twitter search engine is the absolute truth). However, after carefully reviewing the results generated by Twitter's search engine, we found that despite the search engine using lots of parameters to generate better results, "opinion leader" status (or user impact) is not the major decisive factor. This finding puts us into a difficult situation for evaluating our proposed framework since we are not able to obtain a test dataset that is already tagged with topics and in which opinion leaders are selected as a ground truth. Nevertheless, we will try to give some explanations as to why the difference between the two results is significant.

We examine the top 10 users selected by Twitter's search engine with the term "headphones," and these users are as follows: *TunesHeadphones, MarshallHP, SennheiserUSA, wearefrends, Grado, HiFiHeadphones, KossHeadphones, ExtrmHeadphones, JambooHeadphone and brookslaich.* A comparison of these 10 users with our framework result can be summarized in Table 7.10.

Rank	**ID**	**Really an opinion leader?**	**In our leader list**	**Why it isn't an opinion leader**	In database?
1	TunesHeadphones	no	no	Very few tweets (fewer than 20)	no
2	MarshallHP	yes	yes		yes
3	SennheiserUSA	yes	yes		yes
4	wearefrends	no	no	Haven't updated for a long time	no
5	Grado	yes	no		no
6	HiFiHeadphones	yes	yes		yes
7	KossHeadphones	yes	no		no
8	ExtrmHeadphones	yes	yes		yes
9	JambooHeadphone	yes	no		no
10	brookslaich	no	no	Professional hockey player	no

Table 7.10: Top 10 users selected by Twitter with term "headphones"

In conclusion, the main reasons for the difference between Twitter and our framework can be summarised as follows:

- The Twitter search engine did not give the best results since many important opinion leaders are missing in its output. Thus, it make us very hard to evaluate the result generated by our framework because the faulty ground truth results. Initially, we believed that the results of Twitter's search engine would be very representative since Twitter has all of the user data and a large data team. However, these results can give us only a limited indication of opinion leader status. There should be a further search for a suitable dataset.

- Missing data causes losing opinion leaders. "Headphones" is a hot topic all over the world. Hundreds of thousands users are talking about this topic every day. Since our crawler is designed to fetch tweets blindly, we can be sure that not all "headphones"-related tweets will be collected. Additionally, the query limit of Twitter makes the crawling process even harder. After reviewing the comparison in Table 7.10, we attemped to add tweets into the experimental dataset from the following users: *Grado, Extrm-Headphones and JambooHeadphone*, who were previously not in our dataset. We fed the new experimental dataset into our framework and found that these users are listed among the top 50 generated results. This experiment shows the importance of missing data and that missing data poses a new technical challenge on how to scrape "significant" data from an extremely large platform.

- The results of the Twitter search engine clearly do not select the best opinion leaders. In contrast, our proposed framework did select more significant opinion leaders that were not in Twitter's result, such as *mashable, verge and techradar*.

7.5.2 Use Case: Mock Business Case

In this section, we design a mock business scenario to evaluate whether the proposed framework can be useful in commercial areas. We assume that a company wants to enter "Solar Power" market in US and that their

marketing manager wants to have an overview of public on-line opinion to this topic. Thus, this manager asked us to give him a list of opinion leaders on Twitter.

In this case, we selected all tweets tagged with "solar power" as the input dataset. Then, the same "tweet filtering and cleaning" technique described in the previous sub-section is applied. Then, the filtered dataset is fed into the proposed framework with the desired number of leaders, 50. Then, a list of 50 opinion leaders is generated by the proposed framework for further analysis. At the end, we asked an expert in the "solar power" industry to help us rank the results by the user's relevance.

	Number of users
Relevant to the topic	42
Irrelevant to the topic	8
Possible competitors	17 (2 unknown)
Information providers	27

Table 7.11: Overview of generated opinion leaders in mock business case, reviewing by industry expert.

7.5.2.1 Selecting top K users as leaders

An overview of generated opinion leaders are listed in Table 7.11. In the commercial case, the marketing manager has only limited time to go through each leader's tweet in their timeline. Thus, we only provide the top 15 to 25 users from the generated list. This also gives us a chance to compile a confusion matrix to compare the generated results and the results ranked by the expert.

Select top 15 leaders:

We calculate the recall, precision and F-measure based on the confusion matrix shown in 7.12.

$$Recall = \frac{TP}{FN+TP} = \frac{9}{9+6} = \frac{9}{15} = 0.6$$

$$Precision = \frac{TP}{FP+TP} = \frac{9}{9+6} = \frac{9}{15} = 0.6$$

$$F-measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{0.6 \times 0.6}{0.6 + 0.6} = 0.6$$

9 True Positives (TP) Users selected by the framework also appear in the expert's top 15	6 false negatives (FN) Users in the expert's top 15 did not show in the top 15 identified by the framework
6 False Positives (TP) Users selected by the framework did not appear in the expert's top 15	0 true negatives (FN) Users in expert's top 15 are real regular users

Table 7.12: Confusion matrix of top 15 opinion leaders

Select top 20 as leaders:

We calculate the recall, precision and F-measure based on the confusion matrix shown in 7.13.

11 True Positives (TP) Users selected by the framework also appear in the expert's top 20	9 False Negatives (FN) Users in the expert's top 20 did not appear in the framework top 20
9 False Positives (TP) Users selected by the framework did not appear in the expert's top 20	0 True Negatives (FN) Users in the expert's top 20 are real regular users

Table 7.13: Confusion matrix using the top 20 as opinion leaders

$$Recall = \frac{TP}{FN + TP} = \frac{11}{9 + 11} = \frac{11}{20} = 0.55$$

$$Precision = \frac{TP}{FP + TP} = \frac{11}{9 + 11} = \frac{11}{20} = 0.55$$

$$F-measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{0.55 \times 0.55}{0.55 + 0.55} = 0.55$$

Select top 25 as leaders:

We calculate the recall, precision and F-measure based on the confusion matrix shown at 7.14.

17 True Positives (TP) Users selected by the framework also appear in the expert's top 25	8 False Negatives (FN) Users in the expert's top 25 did not appear in the top 25 of the framework
8 False Positives (TP) Users selected by the framework did not appear in the expert's top 25	0 True Negatives (FN) Users in the expert's top 25 are real regular users

Table 7.14: Confusion matrix using the top 25 opinion leaders

$$Recall = \frac{TP}{FN+TP} = \frac{17}{17+8} = \frac{17}{25} = 0.68$$

$$Precision = \frac{TP}{FP+TP} = \frac{17}{17+8} = \frac{17}{25} = 0.68$$

$$F-measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} = 2 \times \frac{0.68 \times 0.68}{0.68 + 0.68} = 0.68$$

The value of precision should be of interest to the marketing manager. If the manager organizes a targeted marketing event based on our result, the value of precision indicates the cost efficiency of the event, whereas the value of recall represents the coverage rate. For example, if the manager uses the top 25 opinion leaders as the target. Then, the precision of 0.68 means that 68% of the invested time will be helpful, whereas 32% of Twitter accounts are not opinion leaders and have little impact.

In the second experiment, we have a real result that is evaluated by an expert that can be used as a more reliable source for comparison with our framework's result instead of using Twitter built-in search engine. It is clear that the precision and recall are much better than in the first experiment where the Twitter built-in search results were used and treated as a perfect opinion leader list.

Additionally, if we have a look at the statistics on the top 50 opinion leaders generated by our result. We successfully identified 42 users that are relevant to the topic, which is evaluated by the expert. In total, 17 competitors were identified, thus saving the company's time by preventing the need to find these users manually. The best feature is that our

result provides 2 competitors that weren't even on their radar, which provides great value to the managers since they can conduct a study on the new competitors. Additionally, our result also provides 27 "information providers". This method will also be helpful in increasing the company's visibility on the market through these users.

Chapter 8

Conclusions

This thesis has demonstrated how the complex and labour-intensive on-line topic group classification and inclination analysis problem can be efficiently and relatively accurately solved using an automated three-layered platform without human interventions. Two key components, namely, topic grouping and inclination classification, were devised, implemented and evaluated. The three-layered platform was first introduced in Chapter 3, followed by topic group mining (Chapter 4) and user inclination mining (Chapter 5). A set of experiments that measured the accuracy of the framework were presented in Chapter 6. The evaluation indicated that the combination of the proposed methods delivers an accurate, scalable and adaptable platform. This chapter highlights the main accomplishments of this thesis.

8.1 Main Contributions

As a cross-disciplinary research work, this thesis covers many different aspects. The thesis began with a series of research claims that need to be considered:

1. The entire process, including data collection, topic extraction and clustering and user inclination clustering, should be automated with minimal user input. The use of this approach enables an efficient working platform for performing massive on-line crowd inclination analyses, which are typically time-consuming and human-labour-intensive tasks.

2. Conducting topic extraction and clustering on very short and unstructured documents in accordance with a crowd-sourced knowledge base provides greater accuracy than does applying traditional information retrieval techniques.

3. A user inclination analysis in accordance with the user relationship obtained from automatically generated topic groups that are clustered from a large number of very short and unstructured documents can be conducted to label the inclination of users' documents with promising precision. This provides the foundation for future studies.

4. Providing an on-line opinion leaders identification methods. It provides a promising result comparing to Twitter built-in search engine. It also generates a better result on a mock business case, which proves that it has a great potential to be used in real business scenarios.

Based on the claim in Chapter 1 and the evaluation in Chapter 6, these claims can now be considered addressed. The key contributions of this thesis are presented here.

8.1.1 Novel Mechanism for Automated User Inclination Platform

The most important accomplishment of this thesis is that it contributes an evaluated platform for the growing field of on-line social network user inclination analysis. There are many research issues that remain to be addressed in this area, particularly mass crowd opinion mining, which is receiving an increasing number of demands from industries, corporations and even governments. The novel three-layered platform proposed in this thesis utilises two fundamental methods: **topic identification and clustering** and **inclination mining and clustering**. This new platform enables a flexible pick-and-mix of suitable technologies to collaborate in a new system, while its generated results can be easily evaluated using our evaluation framework, as reported in this thesis.

8.1.2 High Accuracy in Topic Group Mining

Providing accurately classified topic groups is another major impact of this thesis. Mining topic groups in on-line social networks that use traditional social network analysis (SNA) approaches typically fails as most approaches model the network as a graph, which means that they can only utilise the explicit relationships provided by the target platform. The semantic relationships between the users are disregarded, which produces to low-accuracy results. To overcome this problem, the topics of each document are identified first in this thesis. Tweets are then clustered based on their topics. Furthermore, two different approaches are integrated for document enrichment, which adds Wikipedia topics and categories to the original tweet, into our platform as we identified two major problems with the bag-of-words model, as discussed in Section 4.7. The experiments conducted in Chapter 6 showed that the resulting topic groups clustered using enriched tweets were significantly better than those that used original tweets.

8.1.3 Novel Approach for Inclination Mining for Short-Text Communications

Analysing the inclination of a short text is considerably more difficult than that of traditional documents that use textual features as the 140 character limitation of a tweet is too short to identify sentiment in many cases. Hence, we attempt to utilise other features that are not immediately or obviously related to the content to analyse the opinions suggested by tweets.

Furthermore, various studies have indicated that social interactions are important factors between on-line social network users. This thesis proposes a novel inclination mining method that applies the content of tweets and user relationships which was not previously conducted. By utilising the subjective lexicon dictionary, an accuracy of 83% can be achieved by my proposed method, which we believe can be improved by using more sophisticated lexicon in the future. There are two remarkable advantages of my proposed method: (a) user relationships, which are features that are not immediately or obviously related to the content,

can be easily retrieved from on-line social network platforms, and (b) our proposed method is not restricted to Twitter and can easily be applied to different on-line social network platforms.

8.2 Strengths and Limitations of the Framework

The main strength of the proposed framework lies in its flexible architecture design (Chapter 3) for performing topic group mining (Chapter 4), crowd opinion analysis (Chapter 5) and opinion leader identification (Chapter 6) on a large scale. This finding has laid a foundation for data analysts to swiftly compare the results from different classification or inclination algorithms by switching components without strong programming expertise. The components are loosely coupled in the framework and new components can be easily implemented by common interfaces to produce the results with unified schemas.

While the proposed framework has several prominent features, it also has limitations. In the collection layer, the crawler only accepts the social platform that provides both user relationships and document relationships. Unfortunately, not every social network platform has explicit user relationships. Consider Zhihu[1], which is the largest Q&A social platform in China, as an example. Zhihu does not have a common "following" user relationship. Thus, the proposed platform cannot employ this platform as a data source provider. However, Zhihu can be accepted as a provider if the developer can convert user attributes from data to the self-defined user relationship. The topic classifier in the classification layer assumes that the classified clusters are assigned with an explicit topic. Hence, the replaced classifier requires the same ability to generate topic labels; otherwise, it will be a blocker to the processing data pipeline. The lexicon that is employed in the reasoning layer will greatly affect the final result. We recommend that the developer carefully choose the lexicon that is closely related to the nature of the data source. There is no graphical interface of the proposed framework. An easily operable

[1]https://www.zhihu.com/

web interface for end-users to 1) change the component from the repository, 2) tweak parameters of components in each layer and 3) provide the rich visualisation of the final result will be useful.

Bibliography

[Adamic and Adar, 2001] Adamic, L. A. and Adar, E. (2001). Friends and neighbors on the web. *SOCIAL NETWORKS*, 25:211–230.

[Agarwal et al., 2011] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Alba, 1973] Alba, R. D. (1973). A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3:3–113.

[Angelova and Weikum, 2006] Angelova, R. and Weikum, G. (2006). Graph-based text classification: Learn from your neighbors. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 485–492, New York, NY, USA. ACM.

[Bakshy et al., 2011] Bakshy, E., Hofman, J. M., Mason, W. A., and Watts, D. J. (2011). Everyone's an influencer: quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74. ACM.

[Beaumont, 2011] Beaumont, C. (2011). *New York plane crash: Twitter breaks the news, again.* `https://www.telegraph.co.uk/technology/twitter/4269765/New-York-plane-crash-Twitter-breaks-the-news-again.html`.

[Bodendorf and Kaiser, 2009] Bodendorf, F. and Kaiser, C. (2009). Detecting opinion leaders and trends in online social networks. In *Proceedings of the 2nd ACM workshop on Social web search and mining*, pages 65–68. ACM.

[Bollen et al., 2011] Bollen, J., Mao, H., and Zeng, X. (2011). Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8.

[Cha et al., 2010] Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, P. K. (2010). Measuring user influence in twitter: The million follower fallacy. *Icwsm*, 10(10-17):30.

[Chamlertwat et al., 2012] Chamlertwat, W., Bhattarakosol, P., Rungkasiri, T., and Haruechaiyasak, C. (2012). Discovering consumer insight from twitter via sentiment analysis. *J. UCS*, 18(8):973–992.

[Chau and Xu, 2007] Chau, M. and Xu, J. (2007). Mining communities and their relationships in blogs: A study of online hate groups. *Int. J. Hum.-Comput. Stud.*, 65(1):57–70.

[Cho et al., 2012] Cho, Y., Hwang, J., and Lee, D. (2012). Identification of effective opinion leaders in the diffusion of technological innovation: A social network approach. *Technological Forecasting and Social Change*, 79(1):97–106.

[Cucerzan, 2007] Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *In Proc. 2007 Joint Conference on EMNLP and CNLL*, pages 708–716.

[Davidov et al., 2010] Davidov, D., Tsur, O., and Rappoport, A. (2010). Enhanced sentiment learning using twitter hashtags and smileys. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 241–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Deerwester et al., 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407.

[Everitt et al., 2009] Everitt, B. S., Landau, S., and Leese, M. (2009). *Cluster Analysis*. Wiley Publishing, 4th edition.

[Ferragina and Scaiella, 2010] Ferragina, P. and Scaiella, U. (2010). 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*, CIKM '10, pages 1625–1628, New York, NY, USA. ACM.

[Flake et al., 2000] Flake, G. W., Lawrence, S., and Giles, C. L. (2000). Efficient identification of web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 150–160, New York, NY, USA. ACM.

[Fujiwara et al., 2011] Fujiwara, Y., Irie, G., and Kitahara, T. (2011). Fast algorithm for affinity propagation. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Three*, IJCAI'11, pages 2238–2243. AAAI Press.

[Gao et al., 2012] Gao, H., Li, Q., Bao, H., and Song, S. (2012). How shall we catch people's concerns in micro-blogging? In *Proceedings*

of the 21st international conference companion on World Wide Web, WWW '12 Companion, pages 505–506, New York, NY, USA. ACM.

[Garey and Johnson, 1990] Garey, M. R. and Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA.

[Gibson et al., 1998] Gibson, D., Kleinberg, J., and Raghavan, P. (1998). Inferring web communities from link topology. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems: links, objects, time and space—structure in hypermedia systems*, HYPERTEXT '98, pages 225–234, New York, NY, USA. ACM.

[Girvan and Newman, 2002] Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

[Golder and Macy, 2011] Golder, S. A. and Macy, M. W. (2011). Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, 333(6051):1878–1881.

[Hartigan, 1975] Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 99th edition.

[Hotho et al., 2003] Hotho, A., Staab, S., and Stumme, G. (2003). Text clustering based on background knowledge. *Institute AIFB, Universität Karlsruhe*.

[Hwang et al., 1993] Hwang, C.-L., Lai, Y.-J., and Liu, T.-Y. (1993). A new approach for multiple objective decision making. *Computers & operations research*, 20(8):889–899.

[Jiang et al., 2011] Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. (2011). Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 151–160, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Kantrowitz et al., 2000] Kantrowitz, M., Mohit, B., and Mittal, V. (2000). Stemming and its effects on tfidf ranking (poster session). In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '00, pages 357–359, New York, NY, USA. ACM.

[Kim and Hovy, 2004] Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Kittler and Illingworth, 1986] Kittler, J. and Illingworth, J. (1986). Relaxation labelling algorithms-a review. *Image Vision Comput.*, 3(4):206–216.

[Kleinberg, 1999] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632.

[Krebs, 2002] Krebs, V. (2002). Mapping networks of terrorist cells. *CONNECTIONS*, 24(3):43–52.

[Kulkarni et al., 2009] Kulkarni, S., Singh, A., Ramakrishnan, G., and Chakrabarti, S. (2009). Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 457–466, New York, NY, USA. ACM.

[Kumar et al., 1999] Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. (1999). Trawling the web for emerging cyber-communities. In *Proceedings of the eighth international conference on World Wide Web*, WWW '99, pages 1481–1493, New York, NY, USA. Elsevier North-Holland, Inc.

[Kunegis et al., 2009] Kunegis, J., Lommatzsch, A., and Bauckhage, C. (2009). The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 741–750, New York, NY, USA. ACM.

[Larson, 1996] Larson, R. (1996). Bibliometrics of the World Wide Web: An exploratory analysis of the intellectual structure of cyberspace. *Ann. Meeting of the American Soc. Info. Sci.*

[Li and Du, 2011] Li, F. and Du, T. C. (2011). Who is talking? an ontology-based opinion leader identification framework for word-of-mouth marketing in online social blogs. *Decision Support Systems*, 51(1):190–197.

[Lin et al., 2013] Lin, Y., Li, H., Liu, X., and Fan, S. (2013). Hot topic propagation model and opinion leader identifying model in microblog network. In *Abstract and Applied Analysis*, volume 2013. Hindawi Publishing Corporation.

[Liu et al., 2005] Liu, B., Hu, M., and Cheng, J. (2005). Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA. ACM.

[Ma, 1998] Ma, B. L. W. H. Y. (1998). Integrating classification and association rule mining. In *Proceedings of the 4th.*

[McPherson et al., 2001] McPherson, M., Smith-Lovin, L., and Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444.

[Mihalcea and Csomai, 2007] Mihalcea, R. and Csomai, A. (2007). Wikify!: linking documents to encyclopedic knowledge. In *In CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM.

[Milne and Witten, 2008a] Milne, D. and Witten, I. H. (2008a). An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *In Proceedings of AAAI 2008*.

[Milne and Witten, 2008b] Milne, D. and Witten, I. H. (2008b). Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 509–518, New York, NY, USA. ACM.

[Mokken, 1979] Mokken, R. J. (1979). Cliques, clubs and clans. *Quality & Quantity*, 13(2):161–173.

[Navigli, 2012] Navigli, R. (2012). A quick tour of word sense disambiguation, induction and related approaches. In *Proceedings of the 38th international conference on Current Trends in Theory and Practice of Computer Science*, SOFSEM'12, pages 115–129, Berlin, Heidelberg. Springer-Verlag.

[Newman, 2004] Newman, M. E. (2004). Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330.

[O'Connor et al., 2010] O'Connor, B., Balasubramanyan, R., Routledge, B. R., and Smith, N. A. (2010). From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11:122–129.

[O'Reilly, 2005] O'Reilly, T. (2005). What is web 2.0: Design patterns and business models for the next generation of software.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web.

[Pak and Paroubek, 2010] Pak, A. and Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 1320–1326.

[Palla et al., 2005] Palla, G., Derényi, I., Farkas, I., and Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818.

[Pang et al., 2002] Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

[Qin et al., 2005] Qin, J., Xu, J., Hu, D., Sageman, M., and Chen, H. (2005). Analyzing terrorist networks: A case study of the global salafi jihad network. In *Intelligence and Security Informatics*, volume 3495 of *Lecture Notes in Computer Science*, pages 287–304. Springer Berlin Heidelberg.

[Rangrej et al., 2011] Rangrej, A., Kulkarni, S., and Tendulkar, A. V. (2011). Comparative study of clustering techniques for short text documents. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 111–112, New York, NY, USA. ACM.

[Romm et al., 1997] Romm, C., Pliskin, N., and Clarke, R. (1997). Virtual communities and society: Toward an integrative three phase model. *International Journal of Information Management*, 17(4):261 – 270.

[Rosenberg and Hirschberg, 2007] Rosenberg, A. and Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In *EMNLP-CoNLL*, volume 7, pages 410–420.

[Schwartz and Wood, 1993] Schwartz, M. F. and Wood, D. C. M. (1993). Discovering shared interests using graph analysis. *Commun. ACM*, 36(8):78–89.

[Shafiq et al., 2013] Shafiq, M. Z., Ilyas, M. U., Liu, A. X., and Radha, H. (2013). Identifying leaders and followers in online social networks. *IEEE Journal on Selected Areas in Communications*, 31(9):618–628.

[Shen et al., 2006] Shen, D., Sun, J.-T., Yang, Q., and Chen, Z. (2006). Latent friend mining from blog data. In *Data Mining, 2006. ICDM '06. Sixth International Conference on*, pages 552 –561.

[Smoot et al., 2011] Smoot, M. E., Ono, K., Ruscheinski, J., Wang, P.-L., and Ideker, T. (2011). Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27(3):431–432.

[Steinbach et al., 2000] Steinbach, M., Karypis, G., Kumar, V., et al. (2000). A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, MA.

[Thelwall, 2010] Thelwall, M. (2010). Emotion homophily in social network site messages. *First Monday*, 15(4).

[Triantaphyllou et al., 1998] Triantaphyllou, E., Shu, B., Sanchez, S. N., and Ray, T. (1998). Multi-criteria decision making: an operations research approach. *Encyclopedia of electrical and electronics engineering*, 15(1998):175–186.

[Tumasjan et al., 2010] Tumasjan, A., Sprenger, T., Sandner, P., and Welpe, I. (2010). Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pages 178–185.

[Turney, 2002] Turney, P. D. (2002). Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Wang et al., 2009] Wang, P., Hu, J., Zeng, H.-J., and Chen, Z. (2009). Using wikipedia knowledge to improve text classification. *Knowl. Inf. Syst.*, 19(3):265–281.

[Wasserman and Faust, 1994] Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and applications*, volume 8. Cambridge university press.

[Wu and Huberman, 2004] Wu, F. and Huberman, B. A. (2004). Finding communities in linear time: A physics approach. *European Physical Journal B*, 38:331–338.

[Xu et al., 2014] Xu, W. W., Sang, Y., Blasiola, S., and Park, H. W. (2014). Predicting opinion leaders in twitter activism networks: The case of the wisconsin recall election. *American Behavioral Scientist*, 58(10):1278–1293.

[Yang et al., 2007] Yang, B., Cheung, W., and Liu, J. (2007). Community mining from signed social networks. *Knowledge and Data Engineering, IEEE Transactions on*, 19(10):1333 –1348.

[Zafarani et al., 2014] Zafarani, R., Abbasi, M. A., and Liu, H. (2014). *Social media mining: an introduction*. Cambridge University Press.

[Zhang et al., 2008a] Zhang, X., Furtlehner, C., and Sebag, M. (2008a). Distributed and incremental clustering based on weighted affinity propagation. In *Proceedings of the 2008 conference on STAIRS 2008: Proceedings of the Fourth Starting AI Researchers' Symposium*, pages 199–210, Amsterdam, The Netherlands, The Netherlands. IOS Press.

[Zhang et al., 2008b] Zhang, Y., Fan, B., and bin Xiao, L. (2008b). Web page classification based on a least square support vector machine with latent semantic analysis. In *Fuzzy Systems and Knowledge Discovery,*

2008. FSKD '08. Fifth International Conference on, volume 2, pages 528–532.

Appendix A

Stop Word List

a's, able, about, above, according, accordingly, across, actually, after, afterwards, again, against, ain't, all, allow, allows, almost, alone, along, already, also, although, always, am, among, amongst, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anyways, anywhere, apart, appear, appreciate, appropriate, are, aren't, around, as, aside, ask, asking, associated, at, available, away, awfully, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, believe, below, beside, besides, best, better, between, beyond, both, brief, but, by, c'mon, c's, came, can, can't, cannot, cant, cause, causes, certain, certainly, changes, clearly, co, com, come, comes, concerning, consequently, consider, considering, contain, containing, contains, corresponding, could, couldn't, course, currently, definitely, described, despite, did, didn't, different, do, does, doesn't, doing, don't, done, down, downwards, during, each, edu, eg, eight, either, else, elsewhere, enough, entirely, especially, et, etc, even, ever, every, everybody, everyone, everything, everywhere, ex, exactly, example, except, far, few, fifth, first, five, followed, following, follows, for, former, formerly, forth, four, from, further, furthermore, get, gets, getting, given, gives, go, goes, going, gone, got, gotten, greetings, had, hadn't, happens, hardly, has, hasn't, have, haven't, having, he, he's, hello, help, hence, her, here, here's, hereafter, hereby, herein, hereupon, hers, herself, hi, him, himself, his, hither, hopefully, how, howbeit, however, i'd, i'll, i'm, i've, ie, if, ignored, immediate, in, inasmuch, inc, indeed, indicate, indicated, indicates, inner, insofar, instead, into, inward, is, isn't, it, it'd, it'll, it's, its, itself, just, keep, keeps, kept, know, knows, known, last, lately, later, latter, latterly, least, less, lest, let, let's, like, liked, likely, little, look, looking, looks, ltd, mainly, many, may, maybe, me, mean, meanwhile, merely, might, more, moreover, most, mostly, much, must, my, myself, name, namely, nd, near, nearly, necessary, need, needs, neither, never, nevertheless, new, next, nine, no, nobody, non, none, noone, nor, normally, not, nothing, novel, now, nowhere, obviously, of, off, often, oh, ok, okay, old, on, once, one, ones, only, onto, or, other, others, otherwise, ought, our, ours, ourselves, out, outside, over, overall, own, particular, particularly, per, per-

haps, placed, please, plus, possible, presumably, probably, provides, que, quite, qv, rather, rd, re, really, reasonably, regarding, regardless, regards, relatively, respectively, right, said, same, saw, say, saying, says, second, secondly, see, seeing, seem, seemed, seeming, seems, seen, self, selves, sensible, sent, serious, seriously, seven, several, shall, she, should, shouldn't, since, six, so, some, somebody, somehow, someone, something, sometime, sometimes, somewhat, somewhere, soon, sorry, specified, specify, specifying, still, sub, such, sup, sure, t's, take, taken, tell, tends, th, than, thank, thanks, thanx, that, that's, thats, the, their, theirs, them, themselves, then, thence, there, there's, thereafter, thereby, therefore, therein, theres, thereupon, these, they, they'd, they'll, they're, they've, think, third, this, thorough, thoroughly, those, though, three, through, throughout, thru, thus, to, together, too, took, toward, towards, tried, tries, truly, try, trying, twice, two, un, under, unfortunately, unless, unlikely, until, unto, up, upon, us, use, used, useful, uses, using, usually, value, various, very, via, viz, vs, want, wants, was, wasn't, way, we, we'd, we'll, we're, we've, welcome, well, went, were, weren't, what, what's, whatever, when, whence, whenever, where, where's, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, who's, whoever, whole, whom, whose, why, will, willing, wish, with, within, without, won't, wonder, would, would, wouldn't, yes, yet, you, you'd, you'll, you're, you've, your, yours, yourself, yourselves, zero

Appendix B

Positive and Negative Word - UIC

Here we randomly list 150 positive and negative words from UIC sentiment lexicon as an example. The full list can be downloaded at http://www.cs.uic.edu/∼liub/FB opinion-lexicon-English.rar

B.1 Positive Words

accolades, achievable, adjustable, admire, affirm, agreeable, amenable, amenity, amity, amusing, articulate, assurances, beautifully, better-than-expected, brave, bravo, brighter, captivate, carefree, charming, cheerful, cherished, cleanest, comely, commitment, compact, complementary, contrasty, cost-effective, cure, defender, deginified, delightfully, diplomatic, distinction, diversified, effective, effortless, elate, embolden, enchanting, enough, entertain, entertains, entrust, ergonomical, excellence, excellently, exceptional, excites, exemplary, exonerate, expansive, fabulous, fantastic, fashionable, fast-paced, favorable, feasible, fervor, finer, fortune, fulfillment, futurestic, glamorous, gold, gorgeous, gracious, gratitude, harmoniously, headway, honorable, humour, illuminating, important, intriguing, jubilantly, judicious, keenly, leads, luster, luxury, marvelousness, merry, miracles, nice, nicely, oasis, openly, outperforming, outperforms, overtakes, overtook, peppy, playful, pleases, precious, preeminent, preferes, premier, pretty, prominent, promoter, properly, pros, proud, reasoned, recomend, recommendation, recommended, recover, recovery, rejuvenated, reputation, resilient, resound, resplendent, restored, restructuring, reverently, savior, serene, simplifies, smooth, speedy, splendid, stunning, substantive, tender, trusty, unforgettable, unlimited, unselfish, upgradeable, user-friendly, valiantly, viewable, virtuous, well-backlit, wellbeing, well-bred, well-made, wholesome, willingness, winnable, won, wonderous, woo, worked, yay

B.2 Negative Words

admonition, aggression, ailing, anti-israeli, anti-white, back-logged, barbarous, battering, bemoan, bickering, blasphemous, bleeding, blindside, bogus, brazenness, brittle, butcher, choleric, combative, concerns, contrariness, corrosion, coward, cracks, creaks, crushing, deaf, deceptive, delirious, deplorably, desecrate, despondence, destroyer, diabolically, disaster, disdainfully, dishearten, disintegrate, disorganized, displaced, dispute, dissatisfaction, douchebag, eccentricity, exagerated, extravagantly, extremist, extremists, faithless, flairs, frantic, frustratingly, galls, gangster, gibberish, gimmick, growl, guile, harasses, hateful, heckles, hothead, humiliation, illusory, impediment, impose, impossible, imprudence, incoherently, incommensurate, indecency, ineffectual, inescapably, infamy, infuriated, injudicious, insult, irrelevance, lanky, leak, leery, lunaticism, lurk, madman, malevolent, militancy, misbecoming, mispronounce, mulish, obnoxiously, offender, outcry, outrageousness, oversights, painfull, panders, pathetic, perilously, perplexing, prevaricate, pricier, prison, prisoner, procrastinate, raping, reactionary, renounce, reprimand, reproach, restrict, reviled, revoltingly, ruinous, rumbling, sabotage, sadness, scam, scant, scarily, scary, shiver, shortcomings, sinful, skulk, smother, snarl, snobs, soapy, spiteful, squeak, strangely, stressfully, sullen, tamper, taut, terror-genic, tortures, touted, tragically, tumble, uncaring, uneasy, unfeeling, unreasonable, unsettle, unsound, unworkable, vengeful, warped, worries

Appendix C

Positive and Negative Word - MPQA

Here we randomly list 150 positive and negative words from MPQA subjectivity lexicon as an example. The full list can be downloaded at http://mpqa.cs.pitt
.edu/lexicons/subj_lexicon/

C.1 Positive Words

accommodative, advantages, affirmation, agile, amazing, apt, aristocratic, astonished, astounding, attainable, attune, awestruck, benefit, benefits, booming, calm, candid, capably, catalyst, catchy, charming, chic, clearer, cogent, commonsensible, compelling, competent, compromises, confidence, conviction, cooperation, credence, cute, dazzled, dear, definite, definitive, devoted, dextrous, earnest, effortless, eligible, equitable, established, exclusive, expert, exquisite, fantasy, first-rate, flawless, fond, foremost, glitter, glossy, goodwill, hale, harmony, heartening, heartily, heroic, high-quality, hospitable, humane, immaculate, improving, incontrovertible, indelible, indescribable, inestimable, influential, innocence, inoffensive, instructive, instrumental, intrigue, invaluable, inventive, invincible, inviolable, inviolate, kind, large, loyal, lustrous, marvellous, mature, moderate, modest, ovation, patriotic, peerless, persuasive, posh, praising, precise, preponderance, pro-Beijing, pro-peace, prosperity, prosperous, qualified, recognition, record-setting, resolved, reverent, righteous, risk-free, savvy, scrupulous, seasoned, settle, simplified, smarter, solidarity, sophisticated, sound, splendid, sprightly, squarely, stars, stately, strides, stylish, sufficient, supporter, supportive, sworn, talented, tender, terrific, terrified, thorough, top, tranquility, unconditional, undoubted, unparalleled, uttermost, valuable, versatile, vivacious, well-informed, well-managed, well-positioned, well-run, wide, will, wink, wish, wry

C.2 Negative Words

acrimonious, aggrieved, aghast, apathetic, apocalyptic, audacious, avariciously, bad, barbaric, barbarity, barbarous, bashful, bloodthirsty, bullies, byzantine, cash-strapped, caustic, cheerless, collapse, combative, complacent, congested, counterproductive, crowded, cumbersome, deadbeat, deadly, deprived, despairing, destroyer, diabolic, difficulties, diffidence, disadvantageous, disaffected, disarray, disconsolate, discrimination, dissatisfactory, dissidents, divergent, dizzy, draconian, egocentric, enemies, estranged, expensive, faithless, farcical-yet-provocative, fathomless, feeble, feeblely, feverish, fidgety, foreboding, fractious, fraud, frenetic, fudge, graceless, grouchy, hawkish, hazy, horrendous, ignorant, illusory, impersonal, inattentive, incomprehensible, inconclusive, incredulous, indignant, indiscriminating, inept, inexcusable, inflated, inflexible, injudicious, innuendo, insignificant, insular, intrusive, invasive, jaundiced, lack, lackadaisical, lesser-known, life-threatening, lukewarm, macabre, misbegotten, mischief, misunderstanding, mocking, morbidly, needy, negligible, nightmarish, non-confidence, objectionable, oblivious, oddities, payback, pedantic, poison, premeditated, prohibitive, protests, protracted, punishable, racist, ranting, reckless, refusal, regret, repressive, reticent, satirical, scandalized, secretive, shabby, sick, so-cal, stagnant, surrender, syndrome, taboo, tedious, terrible, ulterior, unclear, uncontrolled, unconvincing, uneasy, unexpected, unfounded, uninsured, unjustifiable, unkind, unlawful, unlawfulness, unneeded, unproductive, unprofitable, unsuspecting, untested, upsetting, urgency, virulent, worthless