MAKING SENSE OF MICRO-POSTS FOR ORGANIZATIONS AND BUSINESSES: LIVE EVENT AND USER COMMUNITY DETECTION

HADI AMIRIEBRAHIMABADI

(M.Eng), University of Tehran

A THESIS SUBMITTTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

NUS GRADUATE SCHOOL FOR INTEGRATIVE

SCIENCES AND ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2013

To mitra for her love, endless support, encouragement, and dedication.

To my parents, Mohammad and Madineh, whose words of support and encouragement and push for tenacity have been my inspiration throughout my life.

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Hadi Amiri

Digitally signed by Hadi Amiri
DN: cn=Hadi Amiri, o=NUS, ou,
email=amiri_eb@yahoo.com,
c=SG

01/25/2013

Acknowledgments

This thesis would not have been possible without the support of many people. I would like to express my sincere gratitude to my adviser, Prof. Tat-Seng Chua, for the continuous support of my PhD study and research, for his patience, motivation, immense knowledge, and above all his honest and serious behavior. His guidance helped me in all the time of my research and writing of this thesis.

I would like to express my deepest appreciation to Prof. Chew Lim Tan and Prof. Min-Yen Kan for their support and guidance during my PhD study. I was the teaching assistant for Prof. Kan's Information Retrieval module. I want to thank him for sharing his teaching experience with me and also for his great discipline in managing the class. Furthermore, he helped me to make life-changing decisions and I am always grateful to him for his guidance. I would also like to thank my Thesis Advisory Committee, Prof. Hwee Tou Ng and Prof. Anindya Datta, who guided me through my PhD journey.

I thank my labmates in the Lab for Media Search (LMS) for their friendships, for the times we had together, for the weekly meetings we had to discuss over our research problems, for the jogging sessions after the meetings, and for all the fun we had during my PhD study.

I would like to thank my wife Mitra Mohtarami for her accompany and endless support. Without her, this thesis would have not been possible.

Last but not the least, I would like to thank my parents, Mohammad Amiri and Madineh Mehrbakhsh, for supporting me spiritually throughout my life.

Contents

Abstract

As a massive repository of User Generated Content (UGC), social media platforms are arguably the most active networks of interactions, content sharing, and news propagation that best represent the everyday thoughts, opinions and experiences of their users. Rapid analysis of such contents is thus critical for user-centric organizations and businesses as the relevant social media contents provide actionable insights for such organizations.

This thesis focuses on *online* discovery and analysis of the social media contents for organizations. We propose algorithms to effectively harvest relevant data about a given organization from social media, identify the emerging and evolving discussions about the organization as well as its user community. Our mining algorithms utilize information about *current keywords*, *users*, *micro-posts*, *topics*, and *opinions* about organizations to tackle the above issues.

We target the following challenges in online analysis of social media contents for organizations: (a) mining opinion words from UGCs, (b) intelligent data harvesting for organizations through real-time discrimination of their relevant contents, (c) online learning of the evolving and emerging topics about organizations, and, finally, (d) community detection for *ambiguous* organizations (those with the *polysemy* problem that the acronym and key terms of the organizations are shared with many entities).

We propose a unified framework to tackle the above issues. In particular, we propose a semantic similarity measure to mine slang and the so-called urban opinion words from UGCs. Furthermore, we propose to identify and monitor the *known accounts* and *key-users* of organizations, in addition to crawling with the *fixed keywords* of organizations like their names etc, to harvest a more representative distribution of data about them. Our intelligent data harvesting approach utilizes the *context* of organizations (characterized by the content and user information) in order to accurately identify their relevant micro-posts. We also propose an effective topic modeling algorithm with *temporal continuity* and *sparsity constraint* to mine the topics and their evolution through time. Finally, we propose a user community detection algorithm to discriminate user communities of the ambiguous organizations.

Extensive experiments on different kinds of UGCs show the effectiveness of the proposed approaches. We show that mining slang and urban opinion words can significantly improve the performance of sentiment classification on UGCs. Furthermore, we show that users and content information are the key factors in judging the relevance of micro-posts to organizations. We show that our data harvesting approach leads to obtaining more relevant contents about organizations that in turn leads to more accurate topic detection for organizations. Furthermore, we show that *topical relations* among users can significantly improve the performance of community detection for organizations in social media.

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Social media portals like Twitter[1], Facebook[2], Google+[3], and more recently Pinterest[4] along with various forum sites are simply means of virtual interactions among people. On these portals users create, share, exchange, and spread contents among their friends and other users in virtual communities and networks.

Social media contents are mainly expressed in the form of *micro-posts*. A micro-post is a very small piece of user generate text (e.g. less than 140 characters in case of Twitter) that usually has low information content and thus prone to miss-interpretation. Figure 1.1 shows a sample of micro-posts (called *tweets* in Twitter's terminology) obtained from Twitter. Each micro-post has the following components: author or user, text content, date and time, and some meta-data information like keywords that start by the "#" symbol. Such keywords are called *hashtags* in Twitter's terminology.

[1] http://www.twitter.com/

[2] http://www.facebook.com/

[3] www.plus.google.com/

[4] http://pinterest.com/

Hashtags are a way to categorize micro-posts into topics. Furthermore, users in social networks are connected to their friends. In particular, each user has a set of users who follows him (his *followers*) and a group of users who he follows (his *followees*). So, the user graph can be easily modeled using these information. Figure 1.2 shows a very small sub-graph of Twitter's user graph. Each node in this graph represents a user and each edge represents a relationship between two users (either followee or follower relationships).

Social media services are extremely popular among online users. For example, as reported by Twitter in March 2012[5], it has more than 140M active users who are, in total, tweeting an average of 340M tweets per day. Such a huge user generated content (UGC) represents the everyday thoughts, opinions, and experiences of the users and provides tremendous opportunities for research in a this area (Aggarwal, 2011).

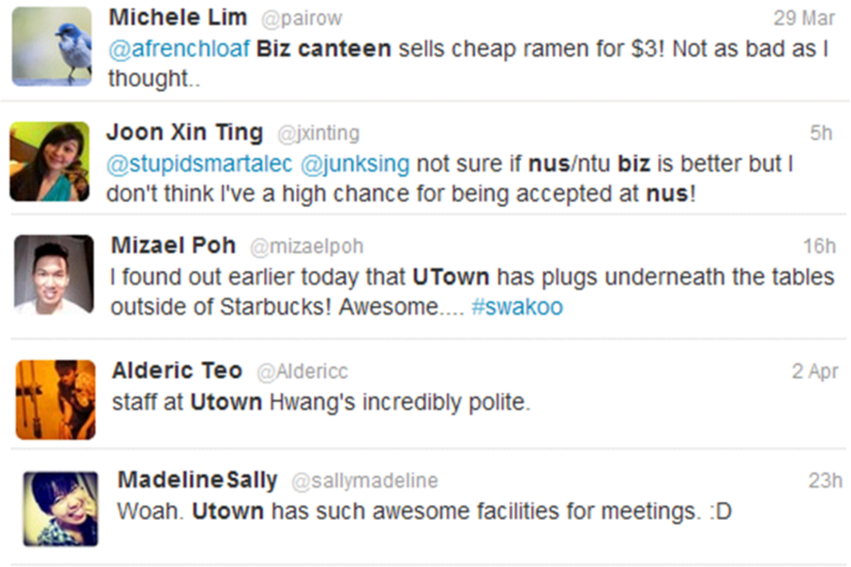
One of the most interesting phenomena happening in social media is their ability to spread micro-posts which may aggregate to form large-scale distributed conversations, topics, or events. This makes social media as an excellent mean for real-time news propagation.

1.2 Motivation

Micro-posts may reflect and reveal information about organizations such as the companies, banks, government organizations, and universities etc. As an example, Figure 1.3 shows the verified Twitter account of the Optus telecommunication company[6]. The biography of this account and its

[5] http://blog.twitter.com/2012/03/twitter-turns-six.html

[6] http://www.optus.com.au/. Optus is the second largest telecommunications company in Australia.



Michele Lim @pairow 29 Mar
@afrenchloaf **Biz canteen** sells cheap ramen for $3! Not as bad as I thought..

Joon Xin Ting @jxinting 5h
@stupidsmartalec @junksing not sure if **nus**/ntu **biz** is better but I don't think I've a high chance for being accepted at **nus**!

Mizael Poh @mizaelpoh 16h
I found out earlier today that **UTown** has plugs underneath the tables outside of Starbucks! Awesome.... #swakoo

Alderic Teo @Aldericc 2 Apr
staff at **Utown** Hwang's incredibly polite.

MadelineSally @sallymadeline 23h
Woah. **Utown** has such awesome facilities for meetings. :D

Figure 1.1: Some sample micro-posts, sampled from twitter

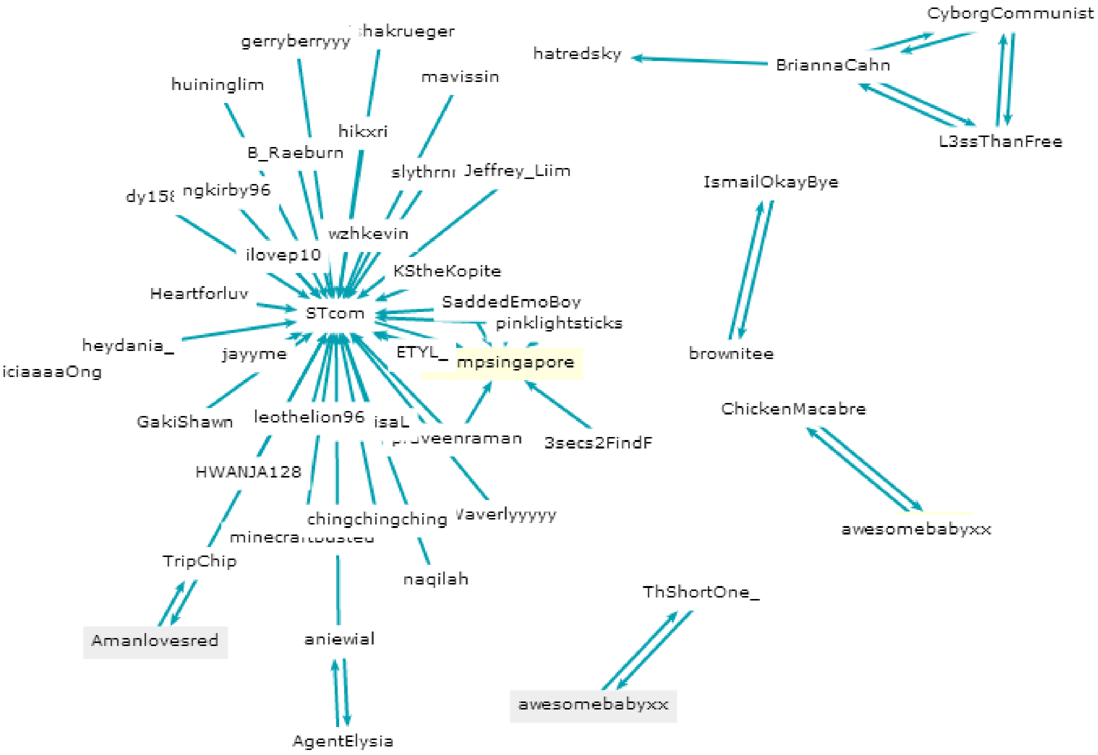
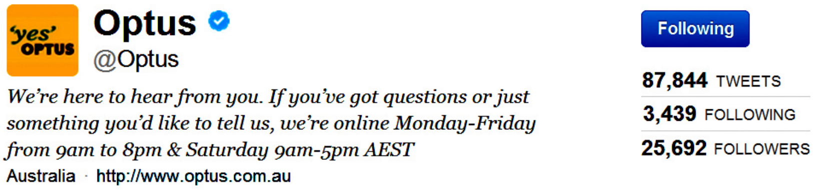


Figure 1.2: Illustration of a small user graph, sampled from twitter



Optus

@Optus

We're here to hear from you. If you've got questions or just something you'd like to tell us, we're online Monday-Friday from 9am to 8pm & Saturday 9am-5pm AEST

Australia · http://www.optus.com.au

87,844 TWEETS

3,439 FOLLOWING

25,692 FOLLOWERS

Figure 1.3: Optus's online department on twitter[7]

activity level indicate that user-centric businesses are spending substantial resources to know what their customers are saying about them. In fact, online discussions about organizations provide important and timely indicators on the spontaneous and often genuine views of the users, fans, and customers of the organizations. It is thus invaluable for organizations to keep track of such live feedback to provide better (personalized) services to their users and identify the public opinion about their services, products, and, in general, all the topics related to the organization. In fact, this information helps organizations to obtain actionable insights from social media. In this thesis we aim to make sense of micro-posts for organizations by identifying what users are saying about the organizations (current topics) and how they feel about those topics (the sentiment of short texts).

However, there are several key challenges in making sense of micro-posts for organizations which we aim to address them in this thesis:

1.2.1 Sentiment Analysis on Short Text

The first challenge is about identifying the public opinions about different aspects of organizations. In fact, sentiment analysis is what user-centric organizations care about the most. Such organizations need to monitor public opinions about their services, products, and brand from social media and other user generated contents (such as reviews) to provide better services to their users. Sentiment detection in user generated contents is challenging as: (a) such short texts provide relatively less content information, and (b) opinions in micro-post are usually expressed with slang and the so-called urban opinion words (such as *delish*, *yummy*, and *yuck* etc) that are not available in standard sentiment dictionaries. In fact, previous research mainly utilize standard sentiment lexicons supported by external knowledge (e.g. emoticons) for this task (Liu, Li, and Guo, 2012; Zhang et al., 2011). However, slang and urban opinion words as strong subjectivity clues are frequently used in user generated contents and need to be automatically identified to improve sentiment detection in micro-posts.

Our objective is to automatically identify such subjectivity clues and detect their sentiment orientation as positive or negative.

1.2.2 Intelligent Data Harvesting

Given an organization, the second challenge is the effective crawling of a live and representative distribution of data about the target organizations. This is a challenging issue because such relevant content is rapidly changing in the social media context. Most current crawling methodologies rely on a fixed list of keywords or a few previously-known keywords such as the name of the organization. However such methodologies cannot cover all the discussions and topics related to the organization. In fact, our investigation shows that using only a fixed list of keywords may cause many relevant micro-posts to be missed due to the lack of such keywords in their content. This in turn results in: (a) many undiscovered topics or, at the very least, (b) late detection of emerging topics due to insufficient relevant data for topic detection.

This challenge is mainly about the online detection of relevant information (relevant keywords and micro-posts) about organizations from large data streams through time. This task is more challenging if we consider the polysemy problem in social media content in which the acronyms of organizations are often shared by many entities. For example, *NUS* is shared between *National University of Singapore*[8], *National Union of Students*[9] and *Nu-Skin*[10] company etc. Thus the purely keyword-based approaches may simply return many irrelevant micro-posts. Such disambiguation task is challenging because: (a) users often use the acronym forms instead of the complete names of the organizations in the social media context (probably due to the length limit imposed by social media portals), (b) micro-posts are usually short and provide little information for disambiguation, and (c) individual users may simultaneously involve in topics about several ambiguous organizations that share the same acronym. To the best of our knowledge, previous research has given less attention to this issue.

Thus, our objective is to propose an effective approach to obtain more relevant micro-posts about a given organization.

1.2.3 Topic Discovery and Monitoring

The fourth challenge is about *online* clustering of the relevant streaming data into coherent set of topics. This is challenging because the input data is of streaming type and hence, in contrast to the traditional topic modeling techniques like LDA (Landauer, Foltz, and Laham, 1998) and LSA (Blei, Ng, and Jordan, 2003), we don't have access to the whole data to perform a high quality clustering. In contrast, with streaming data, new topics as

[8] http://www.nus.edu.sg/

[9] http://www.nus.org.uk/

[10] http://www.nuskin.com/. NU Skin develops and sells personal care products and dietary supplements.

well as the old ones can be introduced or vanished respectively at any point of time. As such, temporal topic detection and monitoring algorithms need to be developed for online analysis of streaming data.

Furthermore, the early detection of *emerging* topics is critical for user-centric organizations as, in case of necessity, it helps them to quickly perform corrective actions before the topics become viral and out-of-control.

Our objective is thus to design an online algorithm for topic modeling (detection and monitoring) in the context of social media. We need our algorithm to be able to keep track of topics through time.

1.2.4 User Community Detection

The fifth challenge is about *online* detection of users with respect to the target organization and the individual topic about the organization. The latter case is more desirable as it provides fine-grained information about users' interest (e.g. topic-sensitive lists of users for the organization).

User community detection for ambiguous organizations is a challenging task because: 1) users often use acronyms instead of completed names of organizations on social media probably due to the length limit imposed by the service providers, and 2) individual users may potentially be involved in discussions on topics that are common for ambiguous organizations. To the best of our knowledge, this is the first work that targets at user community disambiguation in social media.

We note that social relations among users are good indicators of user community. However, social networks are inherently dynamic. That is, new users may join the *user graph* of the organization and old users may stop participating and leave the graph at any time. Furthermore, new links are build upon each new follower or followee relations, and old links disappears as their users stop interacting with each other. This leads to dynamic changes in the structure of the user graph and the user communities of the organization.

Therefore, our objective is to design algorithms to discriminate the user community of ambiguous organizations.

1.3 Problem Definition

Given an organization, the problem we aim to address in this thesis is to harvest relevant micro-posts about the target organization effectively, model the topics related to the organization coherently and keep track of them through time, identify the user community of the organization and rank the users based on their importance and influence in the organizations, and determine the opinion of the users about the topics related to the organization. All the above tasks should be done in an online manner and through time.

In short, the problem we deal with in this thesis is *online* discovery and analysis of relevant *keywords*, *users*, *micro-posts*, *topics*, and *opinions* related to a given organization from social media.

1.4 Contributions

We summarize the contributions of this thesis as follows:

- We propose a principled approach to mine *new* opinion words (including slang and urban opinion words) from user generated contents (see Chapter 4).

- We propose an effective approach to intelligent data harvesting for organizations in the social media context (see Chapter 5).

- We propose a novel adaptation of online sparse coding algorithms to mine the *emerging* and *evolving* topics for organizations (see Chapter 5).

- We propose an effective approach to identify the user community of organizations (see Chapter 6).

Given an organization, the proposed framework provides a clear view of the current status of the organization in the social media context. This is what we refer to as the *sense of organization in social media.*

1.5 Findings

The finding of this thesis, based on different set of experiments and empirical evaluations, are listed as follows. We found that:

- mining slang and urban opinion words and phrases can significantly improves the performance of sentiment classification on user generated contents,

- learning the sentiment orientation of words through time leads to a more accurate polarity inference than learning the sentiment orientation without considering the time factor, while more recent new opinion words leads to greater improvement in sentiment classification performance,

- the combination of users and content information leads to effective discrimination of relevant micro-posts for organizations specially for those with polysemy problem,

- *key-users* of organizations are useful clues to elicit more relevant content about organizations from social media. We show that this result in turn leads to:
  - higher performance of topic modeling algorithms, and
  - earlier detection of emerging topics.
- the performance of topic modeling for organizations improves when topics are not allowed to dramatically change in two consecutive time points, and,
- topical relations among users is an effective mean to detect the user community of organizations.

1.6 Thesis Structure

This thesis is organized as follows: Chapter 2 provides background information and reviews the related works on sentiment analysis, topic modeling and community detection in social media. Chapter 3 presents an overview of this thesis and propose a unified framework for making sense of microposts for organizations. Chapter 4 explains our approach for mining slang and urban opinion words and phrases from user generated contents. Chapter 5 elaborates our approach for harvesting representative distribution of data about organizations and mining the evolution of their topics through time. Chapter 6 explains our approach for mining user community of potentially ambiguous organizations, and, finally, Chapter 7 concludes this thesis and discuss the possible future works.

Chapter 2

Literature Review

2.1 Sentiment Analysis

In this section, we review the previous research on sentiment analysis in including opinion lexicon construction, subjectivity and sentiment classification tasks.

2.1.1 Lexicon Construction

Mining opinion words from user generated content is a crucial prerequisite for effective sentiment analysis. This task includes the detection of new opinion words as well as inferring their polarities. Previous research in this area can be mainly divided into dictionary- and corpus-based approaches. Dictionary-based approaches like (Hu and Liu, 2004; Kamps et al., 2004; Takamura, Inui, and Okumura, 2005; Hassan and Radev, 2010) utilize dictionaries like WordNet to mine opinion words, whereas corpus-based approaches use synthetic and co-occurrence patterns in text for this purpose (Vasileios and Kathleen, 1997; Turney and Littman, 2003; Kanayama and Nasukawa, 2006; Velikovich et al., 2010; Amiri and Chua, 2012). Dictionary-based methods are precise but, in contrast to corpus-based approaches, unable to find informal or so-called urban opinion words. We investigate some of these approaches in this section.

As a dictionary-based approach Hu and Liu (2004) considered the synonyms and antonyms of seeds in dictionaries like WordNet as new opinion words and repeated this process until no new word could be found. Dictionary-based methods are unable to find informal opinion words as they are restricted to the words in dictionaries. To address this problem, corpus-based approaches use synthetic and co-occurrence patterns in text.

As a corpus-based approach, Hatzivassiloglou and McKeown(1997) used conjunctions like ”*and*”, ”*but*” etc with seeds where, for example, *but*” was used as an evidence of opposite polarity (”*simplistic but well-received*”). So if we know the polarity of one of the words in a conjunctive phrase, we can deduce the polarity of the other word. Turney and Littman (2003) proposed to determine the polarity of a word by comparing its tendency toward positive or negative seeds. In particular, given a word w, they determined its polarity score as the PMI between w and positive seeds minus the PMI between w and negative seeds. A positive polarity score indicates a positive word, and negative otherwise.

Velikovich et al. (2010) proposed a graph propagation (GP) technique to perform polarity inference in the graph context. They considered words as the nodes of a graph and weighted edges based on the cosine similarity between the context features of their nodes (extracted from Web n-grams for each node). They computed both positive and negative scores for each unlabeled node based on the maximum weighted paths between the node and seeds. The polarity of each unlabeled node was then computed as the difference between its positive and negative scores.

Amiri and Chua, (2012a) showed that the polarity association among and between seeds and unlabeled words improves the performance of the above techniques. They modeled the polarity inference problem as a semi-supervised learning task in the graph context where seeds and unlabeled words were treated as labeled and unlabeled nodes respectively. They showed that both labeled and unlabeled data are important for learning the polarity scores.

Amiri and Chua, (2012b) showed that "*time*" is another important factor for mining sentiment words. This is because (a) considering the corpus-based approaches, the estimated polarity of opinion words vary with respect to the time that the synthetic and co-occurrence patterns are computed, (b) new opinion words come out at different times as UGC is growing, and (c) though rarely happen, opinion words may change their sentiment orientation through time.

2.1.2 Sentiment Detection

2.1.2.1 Subjectivity Classification

Subjectivity analysis is a well-studied field of research with wide variety of applications (Wiebe et al., 2004; Liu, 2010; Pang and Lee 2008a). Research in subjectivity analysis has been performed at different level of granularity and from different linguistic and computational perspectives (Yu and Hatzivassiloglou, 2003; Ng et al., 2006; Wiebe and Riloff, 2005; Wilson et al., 2009; Liu, 2010; Pang and Lee 2008a).

Subjectivity classification aims is to classify an entity (sentence, question, or document) as subjective or objective. Previous researches typically resorted to opinion lexicons and have shown that the opinion words are important features for subjectivity detection.

In order to identify subjectivity at sentence level, Yu and Hatzivassiloglou (2003) used various features, including the number of opinion words, number of POS tags, and polarity, etc. They reported a high accuracy of 91% for this task. The features that they used are shown in the first row of Table 2.1.

Wiebe and Riloff (2005) showed that a rule-based subjective classifier that simply categorizes a sentence as opinion if it contains at least two strong opinion words can achieve a high precision of 90.4% but a low recall of 34.2%. In contrast, they showed that, a rule-based objective classifier that categorizes a sentence as factual based on the absence of strong opinion words in the sentence can achieve 82.4% precision and 30.7% recall. Some of the features that they used are the count of weak/strong opinion words in current, previous, and next sentences, appearance of pronouns or modals, etc. The rule-based classifier idea was also employed by other researchers for the different tasks of sentiment analysis (Stoyanov et al., 2005; Riloff et al., 2005; Wilson et al., 2009). The features that they used are shown in the second row of Table 2.1.

Ku et al. (2007) utilized opinion words for opinion question identification. They showed that the "*total number of opinion words in question*" and the "*question type*" (i.e. type of question in factual QA systems, e.g. *Yes/No*, *location*, etc.) are the most effective features in opinion question identification. They reported a high accuracy of 92.50% over 1289 opinion questions that were gathered from public opinion polls and other sources. The features that they used are shown in third row of Table 2.1.

Different from the above works, Li et al. (2008a; 2008b) utilized term unigram (TU) weighted by term frequency as feature of an NB classifier

Research work	**Accuracy**
yu et al., 2003- opinion sentence identification: unigrams-trigrams, pos tag, counts of opinion words, polarity of the head verb, the main subject and their immediate modifiers.	91.00%
wiebe and riloff, 2005 - unsupervised subjective/objective classification: syntactic template, count of opinion words, appearance of cardinal number, appearance of pronoun, appearance of modal.	73.80%
ku et al., 2007 - opinion question identification: question type (yes/no, location, etc.), number of opinion words, polarity of opinion words, absolute maximum opinion.	92.50%
li et al., 2008- opinion question identification: char 3 gram, word and pos n-grams ($n \leq 3$), text of question, text of best answer.	71.70%

Table 2.1: list of classification features used for subjectivity analysis

for the opinion question identification task. They showed that this feature is a strong baseline for opinion question identification in cQA data and outperforms all the other features like *characters*, *POS*, *n-grams*, *text of answers*, etc. The features that they used are shown in fourth row of Table 2.1.

Previous research also investigated the relation between subjectivity analysis and word sense disambiguation (Wiebe and Mihalcea, 2006; Gyamfi et al., 2009; Akkaya et al., 2009). They have argued that subjectivity is a property that can be assigned to word senses. They showed that the performance of a word sense disambiguation system can be improved using the subjectivity information and vice versa.

2.1.2.2 Sentiment Classification

Sentiment classification (or polarity detection) is the binary classification task of labeling a subjective entity (word, sentence, document) as expressing either an overall positive or an overall negative opinion (Pang and Lee, 2008a; Liu, 2010).

Words Level: One of the fundamental tasks in sentiment analysis is determining the polarity of words. For example, the words "*excellent*" and "*amazing*" are positive-bearing words, while words like "*poor*" and "*terrible*" are negative-bearing words. Opinion words are used for the majority of sentiment analysis tasks especially for opinion classification (Pang and Lee, 2008a; Liu, 2010).

Review and Sentence Level: Most prior works in the sentiment classification task has been done in the context of reviews (e.g., movie reviews, Amazon book reviews) and binary classification (positive and negative classes) (Pang and Lee, 2008a; Pang et al., 2002). Current research in sentiment classification task can be divided into three categories: classification based on sentiment phrases, classification based on text classification methods, and classification based on score functions (Liu, 2009; Pang and Lee, 2008a):

- **Classification based on sentiment phrases**: This approach uses the positive and negative words and phrases in the documents for classification. Here, the common practice is detecting phrases containing adjectives or adverbs (Turney, 2002).

- **Classification using text classification**: This approach employs common classification algorithms like Support Vector Machine (SVM), or K-Nearest Neighbor (KNN), etc and focuses on feature selection and reduction (Pang et al., 2002; Dave et al., 2003). The state of the art supervised algorithm in this area is reported in (Abbasi et al., 2008) on the movie review dataset (Pang et al., 2002). Abbasi et al. (2008) proposed an algorithm called Entropy Weighted Genetic Algorithm (EWGA) for this task. This method combines genetic algorithm and information gain heuristics for feature selection and reduction. The EWGA algorithm achieved state-of-the-art accuracy of 91% on the movie review dataset. The state-of-the-art semi-supervised method for this task is reported in (Dasgupta and Ng, 2009). They used a novel combination of active learning, transductive learning, and ensemble learning in the classification task and achieved around 76% accuracy. However, we note that the sentiment classification task is highly domain dependent and the accuracy of the algorithms differ across different domain (Pang and Lee, 2008a; Liu, 2009; Turney,2002).

- **Classification using score function**: This approach assigns a score to each word/phrase in the document and generates the overall score by summing up all the scores. The sign of the total score determines the document's class (Dave et al., 2003).

Pang et al. (2002) showed that a support vector machine (SVM) classifier with term unigram as its features and binary weights is a strong baseline for sentiment classification on the movie review dataset. They compared Naive Bayes, Maximum Entropy, and Support Vector Machines classifiers and showed that SVM outperform the other classification methods.

Mao and Lebanon (2006) investigated the problem of predicting local sentiment flow in documents for the purpose of polarity identification at the document (review) level. They modeled the sentiment flow in documents as a sequential model to represent the subjective documents. They used isotonic regression to predict the ordinal sequence of word sets. They assumed that the global sentiment of a document is a function of the local sentiment of its sentences. They showed that their method outperforms a plain bag-of-words representation in predicting global sentiment with a nearest neighbor classifier.

Becker and Aharonson (2010) showed that final sentences of reviews (instead of the whole review) can be used for polarity detection with no significant difference than using the whole reviews. This result shows that the users usually express their overall opinion toward the end of their reviews (especially for long reviews).

More recently, Yu et al. (2012) , Mohtarami et al. (2013a) , and Mohtarami et al. (2013b) proposed algorithms to mine hidden emotions from reviews. Their basic idea is to consider each review as a mixture of hidden emotions and proposed generative models to extract such emotions. Yu et al. (2012) used the emotions to predict product sale performance while Mohtarami et al. (2013a, 2013b) utilized the resultant emotion to answer indirect Yes / No questions as well as polarity detection.

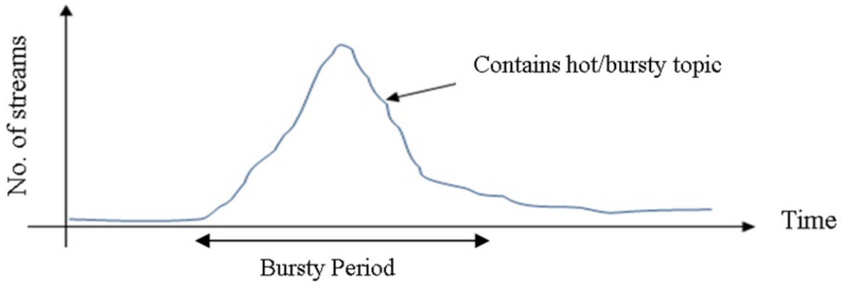


Figure 2.1: Sample bursty topic

2.2 Topic Analysis

2.2.1 Topic Detection and Tracking

Tweets, e-mails, and news are examples of document streams that arrive through time over topics. The intensity of streams is raised, when a particular topic appears. The intensity decreases as the the topic is disappearing. Previous researches have proposed approaches to identify the topics in document streams. Most of the Topic Detection and Tracking (TDT) techniques are based on the intuitive finding that the appearance of a topic in a document stream is signaled by a burst, with certain features rising sharply in frequency as the topic emerges. In addition, a bursty topic is a topic that is hot and appears in a specific time period (burst) as shown in Figure 2.1.

Kleinberg (2003)presented an approach for modeling such bursts. He proposed an infinite-state automation model in which bursts appear as state transitions. He considered that the gap in time between streams is distributed according to an exponential density function. The expected value of the gap has trade-off with the rate of stream arrivals. The bursty intervals can be discovered from the underlying state sequence. To detect bursty topics, Ihler et al. (2006) developed a framework for building a probabilistic model of time-varying counting process in which a superposition of both periodic time-varying and aperiodic processes were observed. To apply these methods to find bursty topics, the employed data stream must represent a single topic.

A possible method to detect all topics in streams is stream pivot clustering approach that involves two steps. First step is grouping similar topics together using clustering techniques to group similar streams together (e.g., K-Means). Second step is extracting the keywords or features of each topic using feature selection techniques (e.g., information gain etc).

In spite of stream pivot clustering approach, Fung et al. (2005) proposed a feature pivot clustering approach with three steps. The first step is identifying the bursty (hot) features. To achieve this aim, they modeled the distribution of a feature in a time window (day) by binomial distribution. The value of the binomial distribution of a feature (probability that the number of streams contain the feature) in any time window may significantly change. A significant change occurs by two reasons; very few documents suddenly contain the feature, or many documents suddenly contain the feature. The latter case can be indicated the bursty feature. The second step is grouping the bursty features into bursty topics. For this purpose, they employed Expected-Maximization (EM) technique to find a maximum probability that the features would be grouped together. The last step is determining the burst (hot time period) of the bursty topics using the highest average probability that the bursty features are appeared together. Weng and Lee (2011) proposed another feature pivot approach for topic detection using clustering of Wavelet-based signals. They attempted to analyze word-specific patterns in the time domain (temporal patterns) by applying wavelet analysis. The wavelet analysis provides precise measurements regarding when and how the frequency of the pattern changes over time. These patterns are utilized to filter away the trivial words, and then the remaining words which are the top bursty words are clustered to form topics. In summary, feature pivot clustering is based on distribution of features. However, stream pivot clustering is based on the content of the streams.

Another type of approach is employing the topic modeling techniques to extract hidden topics from streams and large document collections. Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999b), Non-Negative Matrix Factorization (Lin, 2007; Gu and Zhou, 2009; Ding, Li, and Peng, 2008), and Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan, 2003) are two examples of standard topic models. However, these topic models do not consider temporal information, that is, they do not consider topic changes over time.

PLSA aims to extract topics from large collections of text such that topics are interpretable and it is a method in which:

- documents are represented as numeric vectors in the space of words,
- the order of words is lost but the co-occurrences of words may still provide useful insights about the topical content of a collection of documents,
- each document is a probability distribution over topics , and
- each topic is a probability distribution over words

There are a few limitations that should be considered when deciding whether to use PLSA. Some of these are:

- In PLSA, the observed variable document is an index into some training set. Thus, there is no natural way for the model to handle previously unseen documents, and

- The number of parameters for PLSA grows linearly with the number of documents in the training set. The linear growth in parameters suggests that the model is prone to overfitting and empirically, overfitting is indeed a serious problem.

Various versions of PLSA have been proposed by previous research. For example, Chien and Wu (2008) extended MLE-style estimation of PLSA to MAP-style estimations; a hierarchical extension was proposed in (Hofmann, 1999a); Ding et al. (2008) showed the equivalent between PLSA and non-negative matrix factorization. A high order of proof was shown in (Peng, 2009).

Blei (2003) has proposed Latent Dirichlet Allocation (LDA) that is a generative probabilistic model of a corpus. LDA overcomes both of the PLSA problems by treating the topic mixture weights as a k-parameter hidden random variable. The parameters in a k-topic LDA model do not grow with the size of the training corpus.

The PLSA model assumes that each word of a training document comes from a randomly chosen topic. The topics are themselves drawn from a document-specific distribution over topics. However, LDA assumes that each word of both the observed and unseen documents is generated by a randomly chosen topic which is drawn from a distribution with a randomly chosen parameter.

In LDA model, the basic idea is that the documents are represented as random mixtures over latent topics, where a topic is characterized by a distribution over words. LDA is based on the exchangeability assumption and assumes that words are generated by topics and that those topics are infinitely exchangeable within a document. Infinitely exchangeable is defined based on Finettis Theorem that is described as follows:

- A finite set of random variables $\{x_1, ..., x_N\}$ is said to be exchangeable if the joint distribution is invariant to permutation. If π is a permutation of the integers from 1 to N:

$$p(x_1, ..., x_N) = p(x_{\pi(1)}, ..., x_{\pi(N)}) \tag{2.1}$$

- An infinite sequence of random is infinitely exchangeable if every finite subsequence is exchangeable.

Although the aforementioned standard topic models are strong to detect hidden topics from a collection, employing them without any customization is less effective specially for topi streams that are dynamically changing(e.g., tweets). Thus, a number of temporal topic models have been proposed to consider topic evolution over time. We study some of these methods in the next Section.

2.2.2 Topic Modeling on Tweet Streams

The social media portals like Twitter are the key live resources for mining topics of interest as they are heavily contributed by the crowds and hence are fast in propagating the news. For example, the live tweet streams have been used to address a wide variety of applications, from detecting emergencies like earthquakes (Sakaki, Okazaki, and Matsuo, 2010) and political election outcomes (Tumasjan et al., 2010) to topic mining and evolution (Saha and Sindhwani, 2012; Saha and Sindhwani, 2010; Kasiviswanathan et al., 2011; Hong and Davison, 2010), event detection (Weng and Lee, 2011; Mathioudakis and Koudas, 2010), and expert finding (Lappas, Liu, and Terzi, 2009; Smirnova, 2011). Furthermore, as mentioned before, models of burst and hot topic detection have been developed, from automation (Kleinberg, 2003) to temporal patterns (Weng and Lee, 2011; Yang and Leskovec, 2011).

Existing approaches on tweet streams take keywords and *hashtags*[1] as indicators of topics. While keyword based approaches work well on mining tweets about specific topics (Kotov, Zhai, and Sproat, 2011; Mathioudakis and Koudas, 2010), they are restricted to a set of keywords that are maintained manually. Considering the rapidly evolving nature of the social media content (Sahlgren and Karlgren, 2009), fixed keywords may fail discovering a large fraction of relevant information simply due to missing newly-introduced terms within topics. In addition, while the frequency of a term is a good indicator of its popularity, it may not be a useful measure to identify not so major topics like emerging ones. To tackle these issues, we propose to identify and utilize dynamic keywords as a more effective approach in discovering new topics and producing better coverage over the already-known ones.

Mining evolving and emerging topics in the social media content has become a hot research topic recently (Saha and Sindhwani, 2012; Wang, Agichtein, and Benzi, 2012; Kasiviswanathan et al., 2011; Takahashi, Tomioka, and Yamanishi, 2011; Kamath and Caverlee, 2011; Gohr et al., 2009) as the standard topic modeling approaches like LDA (Landauer, Foltz, and Laham, 1998) and LSA (Blei, Ng, and Jordan, 2003)

[1] Hashtags are keywords attached to the # symbol to categorize tweets based on their context.

are not directly applicable to streaming data as they are mainly designed for static collections. As we mentioned above, such models are not able to adaptively update the topics as massive amount of data streams in. As such, the above online approaches have been introduced to cluster the streaming data in a fast way. We study the most relevant approaches in this section:

Cataldi et al. (2010) proposed an approach to identify emergent keywords and utilized them to find emerging topics. They defined a term as emergent if it frequently occurs in the current time but not in the previous times. Wang et al. (2012) focused on individual users and introduced an LDA-based approach (called Temporal-LDA) that learns topic transition in a sequence of tweets posted by the same user and use it to predict the future distribution of the user's tweets.

Kasiviswanathan et al. (2011) and Saha and Sindhwani (2012) proposed to track the evolution of topics through time. Similar to our approach, they divided the streaming data into evolving and emerging topics. Kasiviswanathan et al. (2011) showed that a simple sparse coding algorithm with the non-negativity constraint is effective for topic modeling in the social media context. Saha and Sindhwani (2012) extended the above work by introducing the temporal continuity constraint. They showed that better topic modeling performance can be achieved, when the continuity between topics matrices in consecutive time stamps is taken into account.

While the above approaches are effective in mining topics in general, they ignore the user information of the tweets and are not designed for addressing the ambiguity issues for entities like organizations. Our work thus focuses on eliciting representative amount of ”relevant” data from different sources of knowledge for organizations and dealing with the ambiguous organizations.

Kamath and Caverlee (2011) proposed to mine *transient crowd*, a short-lived collection of people who actively *communicate* with each other through social messages like *reply* and *mention* of Twitter. They mined the transient crowd by first modeling the communication pattern between users in a graph context and then performing a traditional *minimum cut* clustering algorithm on small portions of the constructed graph. They introduced a locality concept to efficiently identify the transient crowds based on the small portions of the user graph. In contrast to (Kamath and Caverlee, 2011), in our definition, users can be part of the same community as long as they share interest on the same topic (we can call such communities as *interest communities*). As such, the minimum cut algorithm may not be effective to mine such interest communities as there may not be any direct conversation between the users in these communities.

2.3 User Community Detection

Graph clustering for community (or partition) detection has been studied for long time and several effective algorithms have been proposed: divisive algorithms detect inter-community links and remove them from the graph (Girvan and Newman, 2002; Newman and Girvan, 2004), agglomerative algorithms merge similar nodes and communities recursively (Pons and Latapy, 2006), and optimization methods are based on the maximization of an objective function (Newman, 2006b; Newman, 2006a). The quality of the discovered communities is often measured by the *modularity* measure. The modularity of a community/partition is a scalar value in $[-1, 1]$ that measures the density of links inside partitions as compared to links between partitions (Girvan and Newman, 2002; Newman, 2006b). In the case of weighted graphs, it is defined as follows (Newman, 2006b):

$$Q = \frac{1}{2m}\sum_{ij}[A_{ij} - \frac{k_i k_j}{2m}]\delta(c_i, c_j) \tag{2.2}$$

where A_{ij} represents the weight of the edge between nodes i and j, $k_i = sum_j A_{ij}$ is the sum of the weights of the edges attached to node i, c_i is the partition to which node i is assigned, the δ-function $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise, and $m = \frac{1}{2}\sum_j A_{ij}$. Other effective algorithms have also been proposed such as Metis (Karypis and Kumar, 1998) and Graclus (Dhillon, Guan, and Kulis, 2004).

The target of the previously proposed community detection algorithms is to partition the graph into communities of densely connected nodes such that nodes belonging to different communities being only sparsely connected. However, the target of our research is to identify the user community of a given organization as a whole which may comprise of loosely connected or even disconnected partitions. In other words, nodes belonging to different partitions may be from the same user community. In this sense, our community detection task is more related to label propagation over graphs (considering known-accounts as *seed* nodes).

Chapter 3

Unified Framework

In this Chapter, we give a general overview of a unified framework proposed in this thesis to address the issues we discussed in the introduction Chapter. We elaborate our proposed solutions for each of the challenges we discussed in the introduction Chapter:

3.1 Proposed Solutions

3.1.1 Sentiment Analysis on Short Text

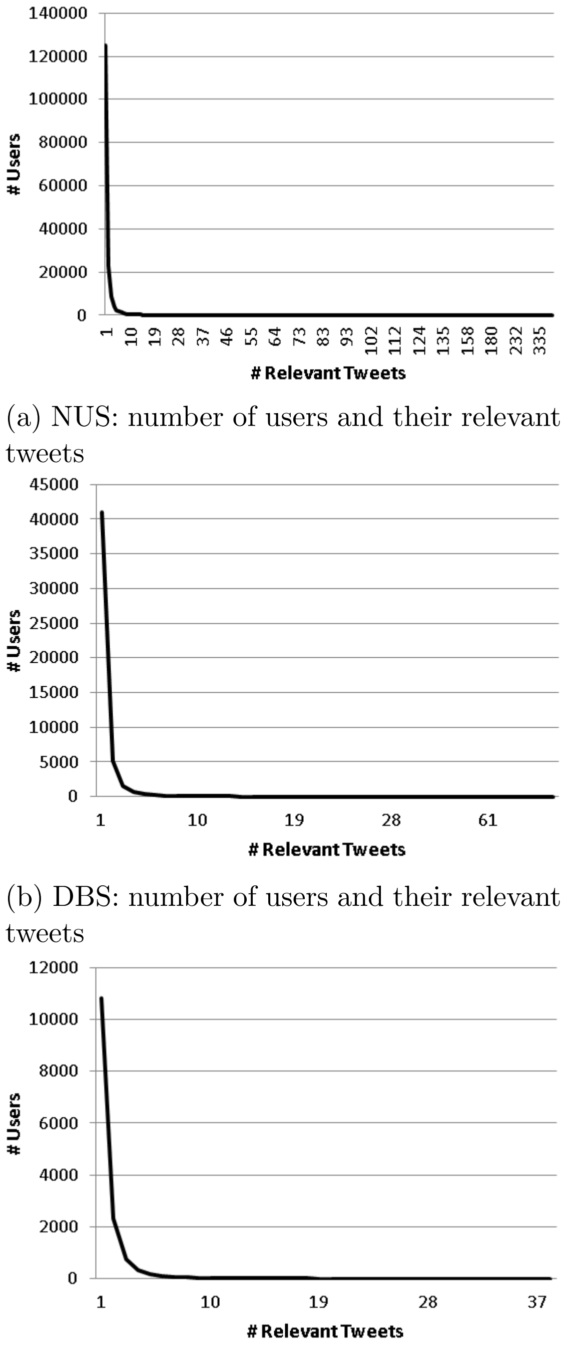
To address this challenge, we introduce a principled approach to constructing sentiment lexicons from user generated contents. In particular, we propose to make use of existing opinion words to extract slang and urban words/phrases from user generated contents. In contrast to previous approaches, our method not only learns the sentiment orientation of words from the existing opinion words but from other new opinion words. This approach is more feasible in the web context where the dictionary-based relations (such as synonym, antonym, or hyponym used by previous approaches) between most words are not available. We show that our approach is effective both in terms of the quality of the discovered new opinion words as well as its ability in inferring their sentiment orientation.

3.1.2 Intelligent Data Harvesting

We propose to address this issue by eliciting data from multiple sources of information: (a) *known accounts*, (b) *key-users*, and (c) *fixed keywords* of the organization. Here the *known accounts* refer to a few official accounts created on social media portals that broadcast news and announcements about the organization; while *key-users* are a dynamic list of active users of the target organization. Fixed-keywords are a list of keywords that specify the potentially relevant tweets of organizations in social media, the name of an organization and its acronym are fixed-keywords for the organization. We experimentally show that the above sources of information collectively elicit more relevant data for organizations as compared to the *fixed keywords* used by the common crawling methodologies.

Note that we expect the *key-users* (active users) of organizations to be usefull clues as we observed the power law correlation between the number of users and the number of relevant tweets for all the three organizations we study in this thesis (See Figure 3.1). We can identify such users based on several criteria like the activity level of the user in sending relevant micro-post about the organization, the number of followers the user has within the organization, and the dominance of the discussions he initiates about the organization.

In fact, our preliminary results (see Figure 3.1) indicate that a small number of users of an organization often produces the major portion of relevant content about the organization.



(a) NUS: number of users and their relevant tweets

(b) DBS: number of users and their relevant tweets

(c) StarHub: number of usersand their relevant tweets

Figure 3.1: Power law correlation between the number of users and their relevant tweets for three organization, namely NUS, DBS, and StarHub. The statistics are obtained from 1-year tweets posted for NUS, and 6-month tweets posted for DBS and StarHub organizations. The number of fixed keywords for NUS and DBS is around 10 keywords, while for StarHub it is only one keyword, the term StarHub itself. Here we only used fixed keywords to find the relevant data, however we believe such power law correlation will remain valid for the entire relevant data.

To discriminate the relevant contents generated by different crawlers, we propose to utilize the *context* of the target organization defined by the current relevant information (keywords and micro-posts) and the user community of the organizations. We design a classifier to predict the relevance of each incoming micro-post to the target organization based its context information, i.e. the content and user/author of the post. We show that this classifier can discriminate relevant micro-posts from irrelevant ones with high accuracy.

3.1.3 User Community Detection

To address this challenge, We consider users who posts relevant information about the target organization as its user community.

we propose to mine the user community of a given organization with respect to the social and topical relations among users. For this aim, we take into account the *temporal order* of micro-posts as the topic set can change through time. We utilize the following information to mine the user communities for organizations: 1) *known-accounts* of the organizations, 2) *social relations* among users, and 3) *topical similarities* among users. Users *follow* know-accounts to receive up-to-date news about the organizations of their interest. In the context of Twitter, social relations are referred to as *follower* and *friend* (*followee*) relationships between users. Together with the known-accounts, social relations are good signals for user community detection. However, not all the users follow the known-accounts of the organizations of their interest. In fact, the social relations among users may be too sparse to precisely discover communities of ambiguous organizations. Therefore, we propose to exploit hidden topics behind the user-generated contents to strengthen the community signals. Our motivation is that if users are interested in more related topics, they belong to the same community with a higher probability. As shown in Figure 6.2, by exploiting both social relations and topical similarities, the graph among users is converted to a more dense graph that makes community detection of ambiguous organizations possible.

3.1.4 Topic Discovery and Monitoring

To address this challenge, we cluster the stream of relevant micro-posts into *emerging* and *evolving* topics. The Emerging topics are the new topics that emerge and potentially become major in a short period of time, while the evolving ones are those that have been detected previously and are smoothly evolving through time. As time passes, the emerging topics become part of the evolving ones and other emerging topics are introduced. For the topic modeling purpose, we propose an *online* sparse coding approach with *temporal continuity* and *sparse matching* constraints. This approach better suits streaming data as it processes each input data only once and therefore is linear with respect to the number of micro-posts. Furthermore, we have a simple purging mechanism to detect the inactive topics to further improve the performance of topic modeling.

3.2 Overview of Approach

The overview of our approach is depicted in Figure 3.2. Given a target organization, we utilize several crawlers to continuously crawl the potentially relevant data about the organization from social media. The resultant data is given to a classifier to make a real-time judgment about their rele-

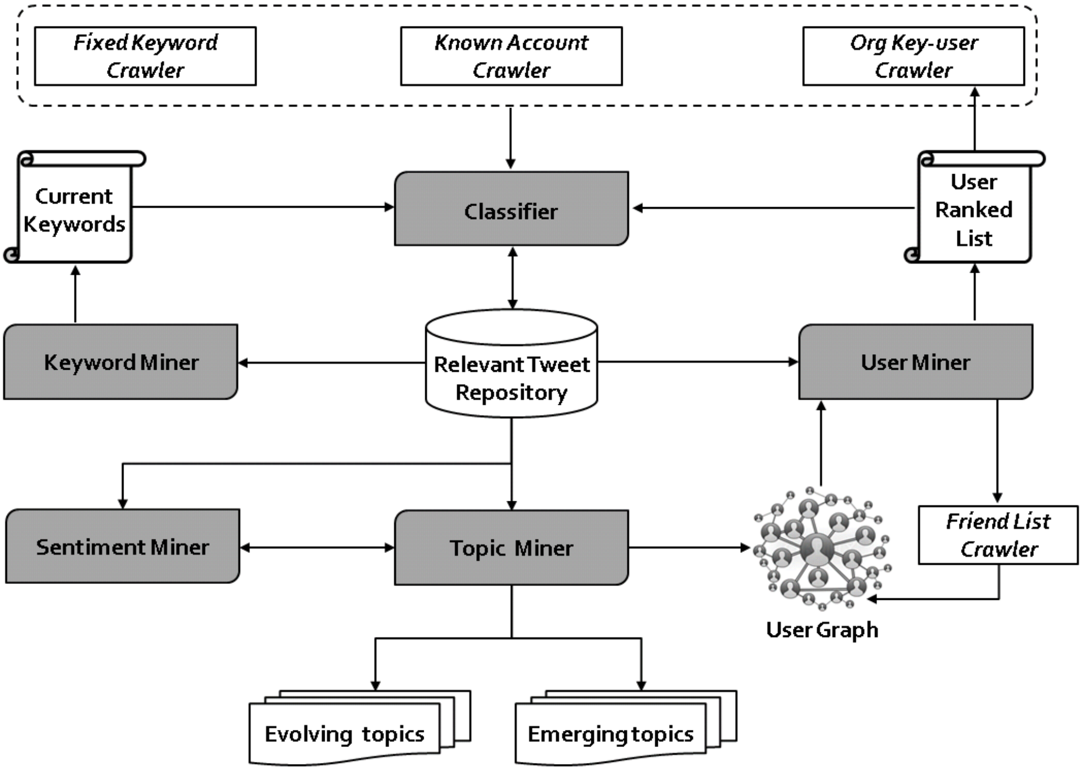


Figure 3.2: Unified framework for making sense of micro-posts for organizations

vance to the target organization. Our classifier makes use of the *context* of the organization (both content- and user-level information) provided by the *keyword miner* and *user miner* components respectively. The relevant data is then stored in the *relevant tweet repository* which will then be given to the *topic miner* and *sentiment miner* components to, respectively, extract the current emerging and evolving topics about the organization and the sentiments of such topics, see Figure 3.2. We provide detail information about each component below and discuss our approach for each component in the subsequent Chapters:

3.2.1 Streaming Data Crawlers

3.2.1.1 Fixed Keyword Crawler

Given the name of the target organization, most brand monitoring systems make use of a few manually selected *fixed keywords* that specify the organization in social media. Examples of fixed known keywords for a given organization are the name of the target organization or its products, the acronym of the organization etc. Such fixed keywords are given to the *fixed keyword* crawler to crawl the micro-posts that contain those keywords. For non-ambiguous organizations, all the data obtained by this crawler are relevant, however, for ambiguous organizations, this crawler may obtain a mix of relevant and irrelevant data about the organization that should be discriminated.

3.2.1.2 Known Account Crawler

Similar to *fixed keywords*, we can manually identify a few set of *known accounts* for the target organization (such as the *Optus* account in Figure 1.3). These are official accounts of the target organization created on social media portals that act as informers and always post relevant micro-posts about their organization. These accounts are given to the *known account* crawler to be observed.

3.2.1.3 Org Key-user Crawler

The *org key-user* crawler is provided with a dynamic list of key-users to be observed. We define key-users as those who frequently comment about the organization and participate in many related discussions. We elaborate our approach for mining key-users in Chapter 5.

3.2.1.4 User Friend Crawler

We also have a *user friend list* crawler that is used to construct the *user graph* of the target organization by crawling the social relationships between users who have posted relevant data about the organization. Note that we initially construct the graph with the known accounts of the organization and their followers as these followers are usually the organization users who want to be informed about the events and happenings about the organization. The user graph evolves over time as new users are identified.

3.2.2 Keyword Miner

The *keyword miner* component utilizes an active learning approach to extract temporally-relevant keywords for organization from the recently seen relevant data. These keywords are considered as *dynamic keywords* at each point of time and used by the *classification* component to determine the content-based relevance of the incoming micro-posts.

3.2.3 User Miner

The purpose of the *user miner* component is to identify the user community and key-users of the organization. The key-users are those active user who are involved in many discussions about the target organization. We monitor such users in order to obtain more relevant data about the organization. This component utilizes the user graph and user activity information to find key-users of the organization.

3.2.4 Classifier

The input data obtained by different crawlers are a mix of relevant and irrelevant data. For example, the data posted by the organization key-users are not always relevant to the target organization as the key-users may also send micro-posts about other subjects like their various life activities. The *classification* component utilizes the *context* information to label the input data as relevant or irrelevant.

3.2.5 Topic Miner

The *topic miner* component utilizes the relevant tweets to detect and keep track of topics for the target organization. We propose a novel adaptation of online sparse coding algorithms to learn the topics in an efficient way. This component divides the stream of relevant data into two sets of: (a) micro-posts with *evolving* (already known) topics, and (b) micro-posts with *emerging* topics. As time passes, the emerging topics become part of the evolving ones and other emerging topics are introduced. We show that this approach is efficient and more suitable for live streaming data as it is fast in learning the topics.

3.2.6 Sentiment Miner

This component determines the sentiment of the micro-posts in each topic. Here, we make use of automatically mined slang and urban opinion words to perform a highly quality sentiment classification on micro-posts.

3.3 Summary

In this Chapter we described our solutions to the challenges rises when we aim to make sense of micro-post for organizations. We described different components that in conjunction provide an effective solution for this problem.

Chapter 4

Mining Slang and Urban Opinion Words and Phrases

The first challenge in making sense of micro-posts for a given organization is to identify the public opinions on different topics about the organizations. Sentiment detection in user generated contents is challenging as opinions in UGC are usually expressed with slang and the so-called urban opinion words that are not available in standard sentiment dictionaries (e.g. "*topnotch*", and "*yuck*"). These subjectivity clues are useful for accurate sentiment classification. In this Chapter, we focus on the fundamental issue of constructing opinion lexicons from UGCs (e.g. tweet and reviews).

4.1 Introduction

Opinion lexicons contain opinion words with their polarity labels, either positive or negative. These lexicons are essential resources for different tasks of sentiment analysis such as opinion mining (Hu and Liu, 2004), opinion retrieval (Ounis et al., 2006), opinion question answering (Li et al., 2009), opinion questions identification (Li et al., 2008), and opinion summarization (Tomasoni and Huang, 2010).

We divide the task of opinion lexicon construction into two sub-tasks:

New Opinion Entity Detection: To the best of our knowledge, there is no principled approach to detect new opinion entities (words or phrases). Previous research either designed hand-crafted rules (Vasileios and Kathleen, 1997; Qiu et al., 2009; Kanayama and Nasukawa, 2006; Popescu and Etzioni, 2005) or used dictionaries and WordNet relations (Hu and Liu, 2004; Takamura, Inui, and Okumura, 2005; Hassan and Radev, 2010; Rao and Ravichandran, 2009; Kim and Hovy, 2004) for this purpose. Each of the above two approaches has its own advantages and disadvantages. For instance both rule-based and dictionary-based approaches are precise, but rules are hard to design and dictionaries have limited vocabulary (coverage) problem. We propose a principled approach to detect new opinion entities from UGC. Our approach effectively combines the above-mentioned methods in a unified framework and is able to detect non-standard entities such as urban opinion words, slang and misspellings etc.

Polarity Inference: The association between seed opinion words and new opinion entities provides a rich source of relationships. We model such relationships in a graph context to assign polarity to new opinion entities. Most of the previous methods only utilized labeled data (i.e. seeds) to predict such polarities, e.g. (Turney, 2002; Kanayama and Nasukawa, 2006), while our approach makes use of both labeled and unlabeled data to predict the polarity of new opinion words. Furthermore, previous approaches requires well-defined relations between words (e.g. synonym, antonym, or hyponym relations available in dictionaries like WordNet) to construct a high quality graph (Kamps et al., 2004; Esuli and Sebastiani, 2006; Rao and Ravichandran, 2009; Mohammad, Dunne, and Dorr, 2009), while we construct the graph from the raw data without being restricted to the above relations. Thus, our polarity inference method is more feasible in the Web context where the data contains many non-standard entities and the above dictionary-based relations are not available.

The rest of this Chapter is organized as follows: Section 4.2 gives an overview of our approach. Section 4.3 elaborates our method for mining new opinion entities and explains some linguistic considerations for this purpose. Section 4.4 describes our optimization framework for polarity inference. Section 4.5 reports the experimental settings and results on both polarity inference and sentiment classification tasks. Section 4.6 further study the effect of "*time*" on mining opinion words and, finally, Section 4.7 summarizes this Chapter.

4.2 Overview of Approach

We construct the opinion lexicon in two steps: (1) mining candidate opinion entities, and (2) inferring the polarity of the entities.

4.2.1 Mining Candidate Opinion Entities

We first extract a set of candidate opinion entities using seeds (the words with already-known polarity). Having two classes of positive and negative seeds, we extract entities (words or phrases) that frequently co-occur with one class (e.g. positive seeds) and rarely with its opposite class (e.g. negative seeds). We expect these entities to be rich in sentiment. We refer to such entities as Significant Entities (SEs) and consider them as candi-

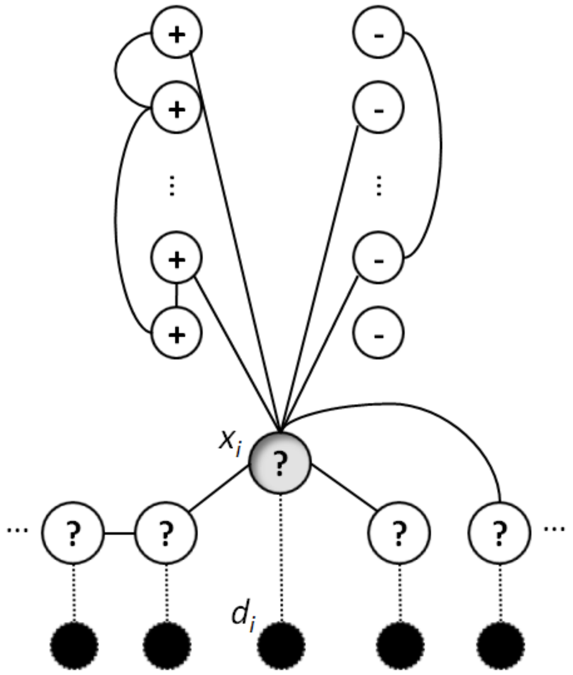


Figure 4.1: Polarity graph: '+' and '-' indicate positive and negative seeds respectively, '?' indicates SEs, and the black nodes are the initial polarity predictions for SEs.

date opinion entities. For instance "*cooool place*" and "*recommend*" are SE because they frequently co-occur with seeds like *fun*, *favorite* etc and rarely with *bad*, *terrible* etc. However, an entity like "*to go*" (semi) equally co-occurs with both positive and negative seeds and cannot be a SE.

4.2.2 Polarity Inference

In the next step, we construct a polarity graph from the seeds and the extracted SEs as depicted in Figure 4.1. In this Figure, the '+' and '-' nodes are labeled nodes (positive and negative seeds respectively), and the '?' nodes are SEs or unlabeled nodes. Each SE node is attached to a corresponding *d-node* (the black nodes) that contains an initial polarity prediction for the SE. The initial predictions are optional. We explain d-node prediction in Section 4.4. The solid edges in the graph reflect the polarity association between the nodes. The weight of such edges is computed as a function of the co-occurrence between their corresponding nodes. In our polarity graph, we restrict such edges to happen only between SEs and seeds, and any two seeds with the same polarity. This prevents the opposite seeds from directly propagating their labels through each other. Once the graph is constructed, the polarity inference problem is modeled as a semi-supervised learning task in the graph context where the labeled nodes are the seeds and the unlabeled nodes are SEs and the aim is to optimize the polarity of SEs based on the graph connectivity information. We treat the SEs with sufficiently high confidence as new opinion entities.

The above two Steps construct the polarity graph without using any dictionary-based relations between the nodes. As such, our method is more feasible in the Web context where such relations are generally not available among many non-standard entities.

4.3 Mining Significant Entities

In this Section, we first explain some linguistic considerations and then describe our method for mining SEs.

4.3.1 Linguistic Considerations

We first utilize three sources of easy-to-collect information as seeds. These sources provide high quality seeds that have high confidence (precision) but low coverage (recall) in sentiment:

- **General Purpose Opinion Lexicons**: We consider those opinion words as seeds that are either labeled as strong in the General Inquirer

(Stone and Hunt, 1963) or subjectivity lexicon (Wilson, Wiebe, and Hoffmann, 2005), or have positive or negative score of one in SentiWordNet (Esuli and Sebastiani, 2006). For SentiWordNet, we only consider the first sense of the words.

- **Linguistic Rules**: As we mentioned before, previous research designed linguistic rules to detect more opinion words. For example, the affixes "*dis*" and "*mis*" were used as evidences for opposite polarities (*honest* ↔ *dishonest*, *fortune* ↔ *misfortune*). We use the above seeds and linguistic rules of (Mohammad, Dunne, and Dorr, 2009) to find more high quality seeds.

- **WordNet Similarity**: We extract the synonym and antonym of each seed from WordNet and consider them as seeds too. The synonyms will be assigned the same polarity as their corresponding seeds, while the antonyms will receive the opposite polarity. We do not repeat this process because we want to ensure the high confidence (precision) of seeds.

The above sources provide an initial set of seeds. We only consider the seeds that occur more than once in our development corpus. These seeds will then be used to mine the significant entities.

Furthermore, we found that negations and disjunctive clauses are important factors to appropriately relate entities to seeds. For example, if a seed is negated, its context words tend to co-occur with the seed's antonym but not the seed itself. Sub-sentences of a disjunctive clause also have opposite polarities. We explain below the way we handle negations and disjunctive clauses in detail:

Negations: Negation words/phrases such as *not*, *none*, *cannot*, *barely*, *lack of*, and *never* etc reverse the sentiment of the seed words. Parser toolkits are useful resources to detect negations and their dependencies in the text. We consider the clause that contains a negation word as the scope of the negation. However, because of the weak grammar and the presence of high amount of short-form texts in UGCs, we designed some manual rules to better tackle the negation. We also consider cases that the negation word is not negating the seeds such as "*not only ... but also ...*", "*last but not least ...*" etc. In total we compiled 36 negation words and rules.

Disjunctive Clauses: We consider disjunctions like *but*, *though*, *although*, *despite*, *in spite of*, *except for*, *except that* etc to relate the entities and seeds. Consider an opinion sentence with two clauses connected by the disjunction "*but*", such as "CLAUSE1, but CLAUSE2" where CLAUSE1 contains the seed word s. These two clauses should have opposite sentiments because of the disjunction "but". Therefore, we can say that the entities of CLAUSE2 co-occur with the antonym of s instead of s itself. For example given the sentence: "*I think it's stylish to hang artworks on walls, but nowadays it's kind of tacky to hang up posters!*" with the word "*stylish*" as its seed, the entities in the clause after "*but*" such as "*tacky*", "*it's kind of tacky*", etc should be related to the antonym of "*stylish*". We also designed a few manual rules to better detect and handle disjunctive clauses. We utilize Stanford toolkit to extract clauses and split the text into sentences.

4.3.2 Significant Entity Extraction

As aforementioned, our assumption is that the entities that frequently co-occur with positive seeds and rarely (or never) with negative seeds are highly likely to be positive. Similarly, the entities that frequently co-occur with negative seeds and rarely (or never) with positive seeds are highly likely to be negative (Turney, 2002; Turney and Littman, 2003; Velikovich et al., 2010; Kaji and Kitsuregawa, 2007).

We use Point-wise Mutual Information (PMI) as the measure of co-occurrence. PMI between two words v and w is defined as follows:

$$PMI(v,w) = \log(\frac{P(v \text{ and } w)}{P(v)P(w)}) \tag{4.1}$$

where $P(v \text{ and } w)$ is the probability that the two words co-occur in the same context (e.g., a sentence or several consecutive sentences), and $P(v)$ and $P(w)$ are the probability of v and w occurring in the entire corpus respectively. PMI is a good measure to associate the words that frequently co-occur in the same context (Turney and Littman, 2003; Islam and Inkpen, 2008).

We define an entity as any word N-grams ($N = 1$, 2, or 3) extracted from UGC. Our aim is to use seeds to find SEs. For this purpose, for each seed s_i, we extract all the entities that occur in the context of s_i, compute their PMI with respect to s_i, and accumulate them in set $N_(s_i)$ as the set of neighboring entities of s_i.

We consider the sentence that contains s_i and its previous sentence as the context of s_i. It is necessary to consider a set of consecutive sentences as the context of the seed words for the following two reasons: (a) many new opinion entities do not co-occur with any seed word at the sentence level, and (b) the same opinion orientation is usually expressed in a few consecutive sentences (Kanayama and Nasukawa, 2006). So we can expect the same orientation among the entities of consecutive sentences. However, we limit the above requirement as follows: (a) if the previous sentence contains a seed with opposite polarity with s_i, we do not consider that sentence in the context of s_i; and (b) if the current sentence contains two seeds with opposite polarities, we only consider the previous sentence as the context of the seed that appears first.

We create the entity pool N from the sets of neighboring entities as follows:

$$N = \bigcup_{\forall i} N_{s_i} \tag{4.2}$$

We then compute an initial polarity score for each entity $e_k \in N$. This score is computed as a function of entity's co-occurrence with positive and negative seeds as follows:

$$InitPScore_(e_k) = \sum_{s_i \in Pos} PMI(s_i, e_k) - \sum_{s_j \in Neg} PMI(s_j, e_k) \tag{4.3}$$

where Pos is the set of positive seeds and Neg is the set of negative seeds. In Equation 4.3, we only consider positive PMI values because it reflects positive co-relation between entities and seeds. The above Equation measures the tendency of the entities towards positive or negative classes of seeds. In the above Equation, $|InitPScore_(e_k)|$ will be high for entities that are highly associated with only one of the positive or negative classes. We first normalize the $InitPScores$ and then sort the entities in descending order of the absolute values of their $InitPScores$. We then pick the Top K entities from this set and consider them as significant entities. These SEs are expected to be rich in sentiment.

4.4 Polarity Inference

The polarity inference problem can be described as follows: Assume that there exist n words $\{x_1, ..., x_n\}$ in the lexicon $\mathcal{X}$. Let the first l words $\mathcal{X}^l = \{x_1, ..., x_l\}$ be the labeled data (seeds) and the remaining words $\mathcal{X}^u = \{x_{l+1}, ..., x_n\}$ be the unlabeled data (the new opinion words). Let y_i indicates the label (polarity score) of x_i. The label of positive and negative seeds are +1 and -1 respectively, i.e. $y_i = +1$ for positive seeds and $y_i = -1$ for the negative seeds. The aim is to find a real-valued function $f : x \to \mathbb{R}$ that gives a polarity score $f(x)$ to each word x. The value of function f on the labeled data x_i is the same as its initial label y_i, i.e. $f(x_i) = y_i$ for $i = \{1, ..., l\}$. The problem is then predicting the polarity scores for the unlabeled nodes, i.e. $f(x_j)$, $j = \{l + 1, ..., n\}$.

The above problem can be best modeled as a semi-supervised learning task in the graph context where the connectivity information of the graph can be utilized to estimate the polarity scores for the unlabeled nodes. We first construct the polarity graph from the new opinion words and seeds, and then define the optimization criteria.

Let $G = (V, E)$ be an undirected edge-weighted graph defined on the dataset $\mathcal{X}$ with nodes V corresponding to the n entities of $\mathcal{X}$, and edges E that are weighted by an $n * n$ symmetric weight matrix $\mathbf{W}$. The weight of the edge $(v_i, v_j) \in E$, w_{ij}, indicates the polarity association between the nodes v_i and v_j and is obtained from $PMI(v_i, v_j)$ as defined in Equation 4.1. Formally, we construct G as follows:

- Any $x_j \in \mathcal{X}^u$ is connected to all the $\mathcal{X}^u$ and $\mathcal{X}^l$ nodes that have positive PMI with x_j, and

- Any $x_i \in \mathcal{X}^l$ is connected to all the $\mathcal{X}^l$ nodes that have the same polarity and positive PMI with x_i.

If there is no edge between two nodes its corresponding weight is deemed to be 0. Note that the PMI function is a symmetric function, i.e. $PMI(a, b) = PMI(b, a)$. The above configuration results in a large graph in which each unlabeled nodes (SE) is potentially connected to several labeled nodes and other unlabeled nodes through different edges/paths (see Figure 4.1).

Furthermore, we assume that, we have an initial polarity prediction (also called dongle node (Zhu, Ghahramani, and Lafferty, 2003)) for each unlabeled node, i.e. $f(x_i) = \hat{y}_i \; \forall i = l + 1...n$. Each d-node is connected to its corresponding unlabeled node with the edge weight of 1 and acts as prior knowledge for the semi-supervised learning framework. $\hat{y}_i$ is set to zero when there is no initial prediction. We explain how to estimate the value of d-nodes in Section 4.4.2.

4.4.1 Optimization Framework

The basic idea of the semi-supervised learning algorithms in the graph context is that the function $f(x)$ should be *smooth* with respect to the graph (Zhu, Ghahramani, and Lafferty, 2003; Wang and Zhang, 2006). $f(x)$ is not smooth with respect to the graph if there is a heavy edge with weight w_{ij} between two nodes x_i and x_j, and the difference between $f(x_i)$ and $f(x_j)$ is large, i.e. $w_{ij}(f(x_i) - f(x_j))^2$ is large. Therefore, the aim of the optimization is to minimize the above value over all the edges in the polarity graph.

Assuming that the d-nodes in Figure 4.1 are connected to their corresponding unlabeled nodes with the weight of 1, our aim is to minimize the following *energy* function:

$$\begin{aligned} E(f) = \gamma \sum_{x_i \in \mathcal{X}^u} (f(x_i) - \hat{y}_i)^2 + \\ (1-\gamma) \sum_{x_i \in \mathcal{X}^u} \Bigg(\sum_{x_j \in Adj^l(x_i)} \alpha w_{ij} (f(x_i) - f(x_j))^2 + \\ \sum_{x_j \in Adj^u(x_i)} (1-\alpha) w_{ij} (f(x_i) - f(x_j))^2 \Bigg) \end{aligned} \tag{4.4}$$

where $Adj^l(x_i)$ and $Adj^u(x_i)$ are the sets of x_i's adjacent labeled and unlabeled nodes respectively, the parameter $\gamma \in [0,1]$ represents the influence of each source of learning (dongle node vs. adjacent nodes) on the polarity of x_i, and the coefficient $\alpha \in [0,1]$ controls the effect of labeled and unlabeled nodes on the polarity of x_i. Equation (4.4) represents the requirements that for each unlabeled node $x_i \in X^u$, we want $f(x_i)$ to be consistent with its d-node, and its neighbors. The smaller values of α increase the effect of the adjacent unlabeled nodes, while greater values of α decrease such effects. Since the paths from unlabeled nodes could potentially be noisy, we expect $\alpha \geq 0.5$ to produce better performance.

The optimization problem can be defined as follows:

$$\hat{f} = \arg\min_f E(f) \tag{4.5}$$

To find a closed-form solution to the above Equation we define an $n * n$ matrix $\mathbf{T}$ as follows:

$$T_{ij} = \begin{cases} 0, & i \in L, j \in L \\ \alpha(1-\gamma) w_{ij}, & i \in L, j \in U \\ \alpha(1-\gamma) w_{ij}, & i \in U, j \in L \\ 2(1-\alpha)(1-\gamma) w_{ij}, & i \in U, j \in U \end{cases} \tag{4.6}$$

where $L = 1...l$ and $U = l+1...n$ are the labeled and unlabeled node indices respectively. Let $\mathbf{D}$ be a diagonal matrix derived from $\mathbf{T}$ as follows:

$$D_{ii} = \sum_{j=1}^{n} T_{ij} \tag{4.7}$$

Let $\mathbf{\Omega} = \mathbf{D} - \mathbf{T}$ be the $n*n$ graph Laplacian matrix (Luxburg, 2007), $\mathbf{f} = [f(x_1), ..., f(x_n)]^T$, and $y = [y_1, ..., y_l, \hat{y_{l+1}}, ..., \hat{y_n}]^T$ where $f(x_i) = y_i$ for the labeled nodes ($i = 1...l$), and $\hat{y_j}$ is the value of the dongle nodes for the unlabeled nodes ($j = l + 1...n$). We can then rewrite Equation (4.4) as follows where $\mathbf{I}$ is the $n * n$ identity matrix (see Appendix A for the derivations):

$$E(f) = \gamma(\mathbf{f} - \mathbf{y})^T \mathbf{I}(\mathbf{f} - \mathbf{y}) + \mathbf{f}^T \Omega \mathbf{f} \tag{4.8}$$

The minimum energy function $\hat{\mathbf{f}}$ of the above quadratic function can be obtained as follows:

$$\frac{\partial E(f)}{\partial f} = 0 \Rightarrow \hat{\mathbf{f}} = \gamma(\gamma \mathbf{I} + \mathbf{\Omega})^{-1} \mathbf{y} \tag{4.9}$$

Because $\mathbf{f}^T \mathbf{\Omega} \mathbf{f} > 0$, $\mathbf{\Omega}$ is a symmetric and positive semi-definite matrix and consequently the above solution is the unique answer to our optimization problem. We normalize this vector into $[-1, 1]$ range.

4.4.2 Polarity Prediction for Dongle Nodes

Given an unlabeled node, $x_i \in \mathcal{X}^u$, we utilize two methods to predict the polarity value of its corresponding d-node, d_i. The first intuitive method is based on Equation 4.3 that computes an initial polarity prediction for the unlabeled nodes. We refer to this prediction as CO in the experiments.

As the second prediction method, we make use of the idea proposed in (Velikovich et al., 2010). In particular, we compute a positive and a negative score for each unlabeled node $x_i \in X^u$. The positive score is computed as the sum over the maximum weighted path from every positive labeled node to x_i. Similarly, the negative score is computed as the sum over the maximum weighted path from every negative labeled node to x_i. The value of the corresponding d-node is then computed as the difference between the two positive and negative scores. Mathematically, for each node $x_i \in \mathcal{X}^u$, we compute the value of its d-node as follows:

$$d_i = \frac{1}{Z}(\sum_{x_j \in \mathcal{P}} S_{ij} - \varphi \sum_{x_k \in \mathcal{N}} S_{ik}) \tag{4.10}$$

where Pos and Neg are the positive and negative labeled nodes in X^l respectively, Z is a normalization term, S_{ij} is the value of the maximum weighted path from x_i to x_j, and φ is a constant value that accounts for the difference in the overall mass of positive and negative flow in the graph, and is computed as follows:

$$\varphi = \frac{\sum_{x_i \in \mathcal{X}^u} \sum_{x_j \in \mathcal{P}} S_{ij}}{\sum_{x_i \in \mathcal{X}^u} \sum_{x_j \in \mathcal{N}} S_{ij}} \tag{4.11}$$

Equation 4.10 assigns high positive (negative) values to an unlabeled node that is connected to multiple positive (negative) labeled nodes through short yet highly weighted paths. If x_i has higher positive score than negative score, then its initial guess will be positive, i.e. $d_i > 0$, and $d_i < 0$ otherwise. We refer to this prediction as GP in the experiments.

Our optimization framework improves upon these two baselines by: (1) imposing the smoothness restriction over the polarity graph (see Section 4.4.1), and (2) preventing label propagation through seeds with opposite polarities.

4.5 Experiments

In this Section we evaluate our approach from two perspectives:

Polarity Inference: We first evaluate the ability of the optimization framework in inferring the polarity of opinion words (polarity inference). We utilize the seed opinion words for this purpose. We assume that part of the seed dataset is unlabeled and evaluate the performance of our optimization framework in predicting the correct label of such seeds.

Sentiment Classification: We then evaluate the quality of the extracted new opinion entities. For this purpose, similar to (Velikovich et al., 2010; Turney, 2002), we consider a word-matching-based review classification task as the measure of evaluation. We expect opinion entities with higher quality result in higher performance of review classification

4.5.1 Data and Settings

Due to unavailability of large scale twitter ground-truth datasets, we resort to a restaurant review datasets for evaluation (note that in review datasets each review has a rating star, e.g 1-5 star, that can be used as the label of the review). This dataset was crawled from *newyork.citysearch.com*[1]. In this dataset, each review has a rating star scaling from 1 (highly negative) to 5 (highly positive). We used a balanced set of positive and negative reviews for the evaluation purpose (7K on positive and 7K on negative reviews).

We also used the newly released Yahoo! Webscope dataset[2] as the development dataset for mining opinion entities. We considered each question thread as an individual document and performed the experiments on

[1] Link to download: http://www.cs.cmu.edu/ mehrbod/RR/.

[2] http://webscope.sandbox.yahoo.com/

the "*Food*"(restaurant) domain. The "*Food*" domain of this collection contains 244K documents and 0.5M sentences. We use these documents to detect SEs and extract co-occurrence information.

In addition, from the seed words that we compiled in Section 4.3.1, we only kept the seeds that occur more than once in our development corpus. In this way, we obtained more than 2,500 seeds (almost balanced on positive and negative categories).

All the experiments in the subsequent Sections were performed through 10-fold cross validation and the two-tailed paired t-test $p < 0.01$ was used for significance testing. Throughout this Section, we use the asterisk mark (*) to indicate significant improvement over the best performing baseline.

4.5.2 Polarity Inference Performance

We use the seed dataset as the ground-truth to evaluate the performance of our optimization framework in polarity inference. For this purpose, we consider part of the seed dataset as the test data (unlabeled nodes) and the rest of the seeds as training data (labeled nodes), and evaluate the performance of the optimization framework in predicting the polarity of the unlabeled nodes. We use the following measures for the evaluation:

$$\begin{aligned} Precision &= \frac{N_{correct}}{N_{tagged}} \\ Recall &= \frac{N_{correct}}{N_{unlabeled}} \\ F1 &= \frac{2 * Precision * Recall}{Precision + Recall} \end{aligned} \tag{4.12}$$

where $N_{correct}$ is the number of unlabeled nodes that are assigned the correct polarities (either positive or negative), N_{tagged} is the number of unlabeled nodes that are assigned non-zero scores, and $N_{unlabeled}$ is the total number of unlabeled nodes.

We use 80% of the seed dataset for training and the rest for testing. In addition, we use 10% of the training set to tune the parameter α. For this purpose, we employ a greed search in the [0.1, 1] range with the greed step of 0.1. We analyze the effect of this parameter on the performance of polarity inference in the next Section. In addition, We treat the dongle nodes as other nodes in the graph and empirically set the value of γ to 0.5[3].

Table 4.1 presents the results. CO indicates the results when we use the co-occurrence information (Equation 4.3) to predict the polarity labels of test data. As Table 4.1 shows, CO produces a low F1 performance of 47.89%. This poor performance is due to the fact that CO ignores the co-occurrence among the unlabeled data. However, we should mention that the performance of CO highly depends on the amount of raw text provided for computing the co-occurrence information.

As Table 4.1 shows, GP produces a higher F1 performance than CO (66.27% vs. 47.89%). This difference is significant and stems from GP's utilization of both edges (direct co-occurrence) and paths (indirect co-occurrence) of the polarity graph. We consider GP and CO as the baselines.

The results of the optimization framework are shown in the last three rows of Table 4.1. OPT indicates the result when there is no initial predictions for the unlabeled nodes, i.e. $\hat{y}_i = 0$ for $i = l + 1 \ldots n$. As it is shown, it outperforms the CO and GP methods significantly and produces a F1 performance of 67.19%. OPT, in contrast to CO or GP, optimizes the

[3] We also experimented with some other values of γ and observed that giving more weight to adjacent nodes improves the performance when there is no prediction available for the d-nodes.

Table 4.1: Performance of polarity assignment for different methods

Method	**Precision**	**Recall**	**F1**
CO	47.89	47.89	47.89
GP	66.27	66.27	66.27
OPT, $\alpha : .5$	69.45	65.06	67.19*
OPT-CO, $\alpha : .7$	65.59	61.45	63.45
OPT-GP, $\alpha : .7$	**71.38**	**66.87**	**69.05***

polarity of unlabeled nodes by imposing the smoothness restriction on the polarity graph. As Table 4.1 shows, the value of α is 0.5 for OPT. This suggests that giving the same contribution to both labeled and unlabeled nodes produces a higher significant performance than both CO and GP when no initial prediction is available.

OPT-CO indicates the result when we use CO as the initial predictions for the unlabeled nodes. As it is shown in Table 4.1, this prediction decreases the F1 performance of OPT from 67.19% to 63.45%. This reduction is expected because the optimization framework has to optimize toward the polarity of both adjacent and d-nodes. Since CO produces poor performance in predicting the polarity of d-nodes, adding it to OPT reduces OPTS's performance.

Finally, OPT-GP gives the result when we use GP (see Equation 4.10) as the initial predictions for the unlabeled nodes. It outperforms both CO and GP by 21.16% and 2.78% in F1 score and the improvements are significant. OPT-GP also outperforms OPT by 1.86%. This result suggests that when we have better initial predictions, the performance of the optimization framework increases. Here the value of parameter α is set to 0.7 which emphasizes the important role of the labeled data (seeds) in the learning process. We study the effect of this parameter in the next Section.

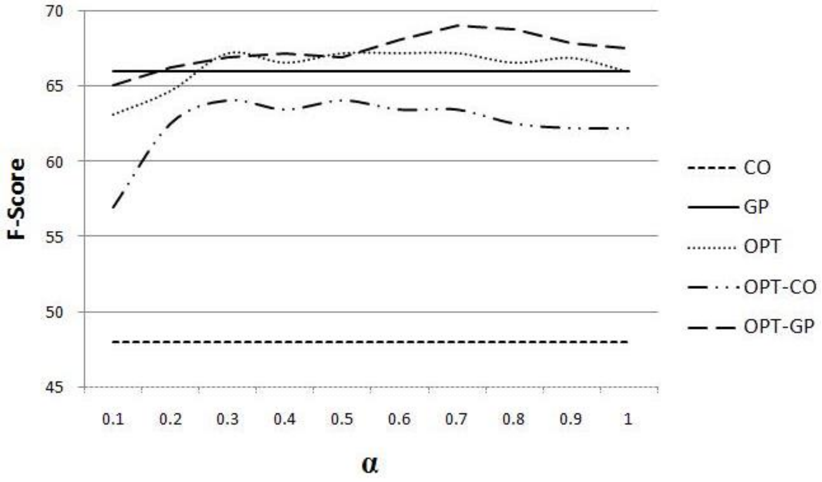


Figure 4.2: The effect of α on polarity inference.

4.5.2.1 Parameter Analysis

In this Section we study the effect of parameter on our optimization framework. As we mentioned before, this parameter has been tuned over 10% of the training data by a greed search in the [0.1, 1] range. We plot the F1 performance of different approaches (discussed in Table 4.1) on the test set with respect to parameter α. Note that in case of OPT the value of the d-nodes is 0 and therefore the parameter α has to be greater than 0, otherwise $f(x_i) = 0$ for $i = l + 1...n$ (See Equation (4.4)).

The results are shown in Figure 4.2. As it is clear, the best performance is obtained when we use the optimization framework in conjunction with the GP predictions, OPT-GP, and the worst performance belongs to CO. In addition, both OPT and OPT-GP perform better than both baselines for any $\alpha \geq 0.3$.

As expected, learning the predictions from GP, i.e. OPT-GP, improves the performance of OPT for all the values of α except when $\alpha = 0.3$ and $\alpha = 0.5$ where the performance of OPT is slightly higher than OPT-GP. This small reduction could be because of the noise in the unlabeled data. Figure 4.2 also shows that OPT and OPT-GP outperform OPT-CO independent from the value of α.

As expected, the smaller values of α produce lower performances. This shows that the labeled data play a crucial role in the learning process. However, Figure 4.2 shows that, to a lesser extent, learning from unlabeled data is also important. This is because when the optimization framework only learns from the labeled data, i.e. when $\alpha = 1$, the performance of both OPT and OPT-GP decreases. This indicates the importance of the unlabeled data in the learning process.

4.5.3 Sentiment Classification Performance

The aim of SC is to assign a polarity label (positive or negative) to any given review. We expect that the performance of SC to be higher when we use an opinion lexicon with higher quality.

As the ground truth, we treated all the reviews with 1 or 2 stars as negative reviews, and the reviews with 4 or 5 stars as positive reviews. We obtained a total number of 14K reviews (balance on positive and negative classes) from the review dataset. To perform the SC experiments, we learned new opinion entities (SEs) from the cQA dataset and tested their SC performance on the 14K reviews.

We performed the word-matching-based SC as follows: given a review, the sentiment score of the review was computed as the sum of the polarity scores of the SEs that appear in the review. An overall positive sentiment score indicates a positive review; and negative otherwise. We also considered negations and disjunctive clauses as we explained in Section 4.3.1. Here we do not use any classifier in order to emphasize the

Table 4.2: Sentiment classification performance on positive reviews

Lexicon	**Precision**	**Recall**	**F1**	**imp**
Seed_Lexicon	69.59	96.68	80.93	-
cQA_OPT	71.85	96.58	82.40*	+1.47
cQA_OPT-GP	72.34	96.59	**82.72***	+1.79

Table 4.3: Sentiment classification performance on negative reviews

Lexicon	**Precision**	**Recall**	**F1**	**imp**
Seed_Lexicon	97.36	42.88	59.54	-
cQA_OPT	96.57	56.16	71.02*	+11.5
cQA_OPT-GP	96.45	58.80	**73.06***	+13.5

quality of the lexicons. A higher performance can be obtained if we use an appropriate classifier.

We constructed separate polarity graphs for each value of N-Gram (N=1, 2, 3) and learned the Top 1000 SEs that have sufficiently high confidence, i.e. $|f(x_i)| \geq 0.5$, for each set. We then stored all these SEs into a lexicon to perform SC. Here, we only perform the experiments with the OPT and OPT-GP methods as they are the best performing methods based on the results of the previous Section. All the other parameters are set as reported in the previous Section, i.e. $\gamma = 0.5$ and $\alpha = 0.5$ for OPT, and $\gamma = 0.5$ and $\alpha = 0.7$ for OPT-GP.

Tables 4.2, 4.3, 4.4 show the performance of SC using different opinion lexicons and for different types of reviews (positive, negative, and all reviews respectively). The Seed_Lexicon only contains the seeds, while the other lexicons, namely cQA_OPT and cQA_OPT-GP, contain the combi-

Table 4.4: Sentiment classification performance on all the reviews

Lexicon	**Precision**	**Recall**	**F1**	**imp**
Seed_Lexicon	76.28	69.78	72.89	-
cQA_OPT	79.32	76.37	77.82*	+4.93
cQA_OPT-GP	**79.90**	**77.70**	**78.78***	+5.89

nation of seeds and SEs (mined from the cQA dataset) where OPT and OPT-GP were used for polarity inference respectively. The "*imp*" column shows the amount of F1 improvement over the Seed_Lexicon.

As Tables 4.2 and 4.3 show Seed_Lexicon produces a high F1 performance of 80.93% for the positive class, but a poor F1 performance of 59.54% for the negative class. We expected the Seed_Lexicon to have high precision but low recall for SC. But this is only the case for the negative class.

To find the reason, we count the number of times that seeds occur in positive and negative reviews and it turns out that the positive seeds occur more frequently than negative ones. This affects the performance of our word-matching-based sentiment classifier. Table 4.5 shows the statistics. The "*w negation*" column means we take into account the negation words/rules as well, i.e. a negated positive (negative) seed increases the count of negative (positive) seeds. The "*w/o negation*" column reports the statistics without considering negation rules/words.

As Table 4.5 shows, the occurrence of positive seeds is much greater than the negative seeds in the positive reviews (7.10 and 7.65 times greater than with and without considering negation words/rules respectively)[4]. As such, the word-matching-based sentiment classifier is able to correctly label many of the positive reviews as positive. This justifies the high recall of Seed_Lexicon for the positive class (96.68%). On the other hand, we found that the occurrence of positive seeds is slightly higher than the negative seeds in the negative reviews (1.04 and 1.29 times greater respectively). This seems to indicate that people tend not to use many negative words

[4] with negation, the occurrences are 4,777 and 13,040 in positive and negative reviews respectively.

Table 4.5: Occurrences of seeds in reviews

	Pos Reviews		**Neg Reviews**	
	w negation	w/o negation	w negation	w/o negation
Pos Seeds	48,704	49,135	26,183	29,007
Neg Seeds	6,855	6,424	25,234	22,410
Pos/Neg Ratio	7.10	7.65	1.04	1.29

even in negative reviews. This causes some of the negative reviews to be wrongly labeled as positive by the word-matching-based sentiment classifier. This in turn results in the relatively low precision of the Seed_Lexicon for the positive class (69.59%).

As shown in Table 4.5, the occurrence of positive seeds is slightly greater than the negative seeds in negative reviews (1.29 times greater). At the same time the occurrence of negation words/rules in negative reviews is greater than positive reviews. The above two indicators show that, in the negative reviews, users usually use negated positive seeds to express their negative opinions. This is consistent with the positive encouragement principle of critique, i.e. the shortcomings can be pointed out in a positive manner.

Tables 4.2 and 4.3 show that the cQA_OPT and cQA_OPT-GP lexicons result in significant improvement over Seed_Lexicon for both positive and negative classes, with a greater improvement on the negative class.

Table 4.4 shows the overall SC performance on all the reviews. The results show that both cQA lexicons significantly outperform the Seed_Lexicon. Overall cQA_OPT results in 4.93% improvement over the Seed_Lexicon (77.82% vs. 72.89%) while the cQA_OPT-GP result in 5.89% improvement over the Seed_Lexicon (78.78% vs. 72.89%).

4.6 Further Analysis on New Opinion Words

In this Section we further investigate the effect of "*time*" in mining sentiment terminology. In particular, we show that the current non-time-based polarity inference approaches may assign opposite polarity to the same opinion word at different times. We show that the polarity scores computed at different times can be efficiently combined to compute a globally correct polarity score for each opinion word. To the best of our knowledge, this thesis is the first work that investigates "*time*" as an important factor in mining sentiment terminology.

The method proposed in the previous Section utilizes synthetic and co-occurrence patterns to mine new opinion words (see (Turney and Littman, 2003; Amiri and Chua, 2012)). Here, we show that "*time*" is another important factor for mining opinion words in the sense that: (a) new opinion words emerge at different times as UGC is growing, (b) the current methods based on synthetic and co-occurrence patterns often estimate different polarities for the same opinion word at different times, and (c) though rarely happen, opinion words may change their sentiment orientation through time. For example, the term "*awesome*" meant "*terrifying*" in the past, but nowadays it means "*amazing*".

Figure 4.3 illustrates the polarity scores of several new opinion words estimated by the popular non-time-based Turney and Littman's (2003) method at different times (the time granularity is six months). As Figure 4.3 shows, for each word, the polarity scores are often wrongly estimated at different times and vary through time. This is because of the varying co-occurrence patterns observed at different times.

To tackle the above challenges, we propose a novel polarity infer-

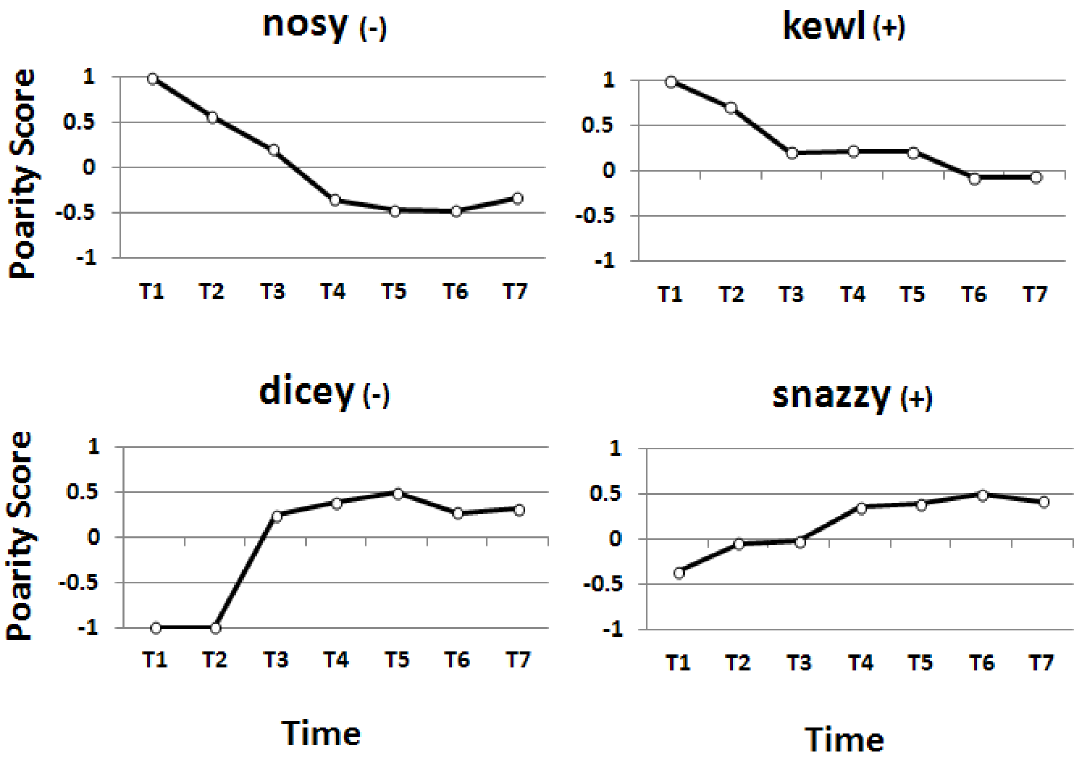


Figure 4.3: Polarity scores computed by the Turney and Littman's (2003) method at different times.

ence technique to infer *time accumulated* polarity scores for the new opinion words. We consider the polarity scores obtained at different times as polarity *evidences* and combine them to compute the time accumulated polarity scores. For this purpose, we use the Dempster-Shafer combination theory (Dempster, 1968; G., 1976) which is known to be strong with respect to flawed evidences. We show that this consideration leads to more accurate polarity inference.

Furthermore, although the method we proposed in the previous Section to detect new opinion words is precise (as it utilizes many linguistic features), it has high computational cost due to heavy usage of parser. To account for this, here we propose a much faster method with the same underlying approach as our previous method to find new opinion words. Our method utilizes the *interchangeability* characteristic of words to detect new opinion words.

4.6.1 Interchangeability

In this Section, we present a context-aware approach to mine new opinion words through time. We propose to find the interchangeable words that are distributionally similar with seeds (words with already known polarity) and consider them as candidate new opinion words. We define the interchangeability between two words as follows:

Definition 1: Two words are interchangeable, if they have:

1. low co-occurrence (see 4.14), and

2. high overlap in their left and right neighboring words

Due to the intuitive definition of interchangeability, the co-occurrence between two interchangeable words is expected to be low. For example, since "*suggest*" and "*recommend*" are interchangeable, we usually use one of them in a sentence to give a suggestion. Furthermore, we here separately deal with the left and right neighboring words to discard the effect of the words that occur on the opposite sides of the target words in measuring their interchangeability.

To find interchangeable words with seeds, we assume that the time-span T_i includes all the reviews written in the time interval $[t_{i-1}, t_i]$. Let T_i, $i \leq j$, be the *source* time-span and T_j be the *target* time-span. The words of T_j that are interchangeable with at least one seed of T_i are candidate new opinion words. Given two words a^i and b^i from the same time-span T_i, we first define the side-oriented PMI between them as follows:

$$\begin{aligned} &PMI^l(a^i, b^i) = \\ &\qquad \log\left(\frac{Count^i(a^i \textit{ occur on left side of } b^i)M^i}{Count^i(a^i)Count^i(b^i)}\right) \\ &PMI^r(a^i, b^i) = \\ &\qquad \log\left(\frac{Count^i(a^i \textit{ occur on right side of } b^i)M^i}{Count^i(a^i)Count^i(b^i)}\right) \end{aligned} \tag{4.13}$$

where $Count^i(x)$ is the number of sentences that contain x at time-span T_i, and M^i is the number of sentences at T_i.

In addition, given the word a^i from the time-span T_i, we refer to its left (right) significant neighboring words (SNWs) as the words of T_i that (a) occur on the left (right) side of a^i, and (b) have positive PMI^l (PMI^r) values with respect to a^i. For each word, we only consider its top z left (right) SNWs that have the highest PMI^l (PMI^r) values with respect to the word.

Let v^i be a *seed* word from T_i, $i \leq j$, and w^j be a target word from T_j. We define $S^l_{v^i w^j}$ and $S^r_{v^i w^j}$ as the common left and right SNWs of v^i and w^j respectively and compute the *context similarity* between the two words as follows:

$$\begin{aligned} Sim(v^i, w^j) = \tfrac{1}{z} \sum_{O \in \{l,r\}} \sum_{u \in S^O_{v^i w^j}} \\ [(PMI^O(v^i, u^i))^\zeta + (PMI^O(u^j, w^j))^\zeta] \end{aligned} \tag{4.14}$$

where O indicates left or right, u is a common (left or right) SNW of both v^i and w^j, and ζ is a constant. Equation 4.14 computes the similarity between two words by aggregating the PMI values of their common left and right SNWs. It assigns high similarity scores to the words that either (a) frequently co-occur, or (b) rarely co-occur but have high semantic association, such as “*recommend*” and “*suggest*”. According to Definition

1, we are only interested in the latter case, so, we discard the words that frequently co-occur. For this purpose, we use side-oriented PMI as the measure of co-occurrence and compute the interchangeability score between two words as follows:

$$Int(v^i, w^j) = \frac{Sim(v^i, w^j)}{c + \sum\limits_{O \in \{l,r\}} PMI^O(v^i, w^i) + PMI^O(v^j, w^j)} \tag{4.15}$$

where c is a small constant. We construct an interchangeability pool, $\mathcal{P}_{ij}$, for each source-target time-pair, (T_i, T_j) $\forall i \leq j$, as follows:

$$\mathcal{P}_{ij} = \{w_1^j, w_2^j, ...\} \tag{4.16}$$

where each $w_k^j \in \mathcal{P}_{ij}$ is a candidate new opinion word of T_j that is interchangeable with at least one seed of T_i.

4.6.2 Non-Time-Based Polarity Inference

We utilize a non-time-based approach to first assign a polarity score to each candidate new opinion word w_k^j that appears in an interchangeability pool. In particular, for each w_k^j, we use all the reviews up to time T_j to compute the polarity score of w_k^j obtained at time T_j. This will be considered as a polarity evidence for the word w in the future.

We use the optimization framework proposed in the Section 4.4 to compute the polarity scores at different times. Here, we consider each candidate opinion word $w_k^j \in \mathcal{P}_{ij}$ as an unlabeled node that, at the end of this process, will be assigned a polarity score $f(w_k^j)$ by the optimization framework. We refer to this value as a polarity evidence for the word w obtained at T_j and show it by $Pol(w_k^j)$.

4.6.3 Time Accumulated Polarity Inference

As we elaborated before, non-time-based polarity inference methods may assign different and even opposite polarity scores to a given opinion word at different times. This is mainly because such methods rely on the noisy co-occurrence patterns obtained at one particular time. To tackle this issue, we compute a time accumulated polarity score for each candidate opinion word using its polarity scores obtained at different times. For this purpose, we utilized the Dempster-Shafer combination theory as it is strong with respect to the flawed evidences. We first formulate this problem into a Dempster-Shafer combination problem:

In the Dempster-Shafer theory (Dempster, 1968; G., 1976) there exist a set of mutually exclusive alternatives which is called the *frame of discriminant* Θ. For example, for opinion words, Θ can be defined as follows:

$$\Theta = \{positive, negative\} \tag{4.17}$$

The Dempster-Shafer theory assigns a *belief* value to each element of the power set of Θ. Formally, the function $m : 2^{\Theta} \rightarrow [0, 1]$ is called *basic probability assignment* (BPA), if it has the following properties:

$$m(\phi) = 0, \sum_{A \in 2^{\Theta}} m(A) = 1 \tag{4.18}$$

where $m(A)$ indicates the belief value that the proposition $A \in 2^{\Theta}$ is true for an observation (i.e. a word here). Obviously, the belief values of the power set members should add up to 1.

BPAs can be inferred from various evidences using the combination rules of the Dempster-Shafer theory. For example, as the first evidence, let the polarity score of the word w at time-span T_1, i.e. $Pol(w^1)$, be a positive value $0 \leq s \leq 1$. The BPAs for this evidence can be defined in Dempster-Shafer terms as follows:

$$\begin{aligned}
&m_{w^1}(\phi) = 0\\
&m_{w^1}(positive) = s\\
&m_{w^1}(negative) = 0\\
&m_{w^1}(positive\ \ or\ \ negative) = 1 - s
\end{aligned} \tag{4.19}$$

Similarly, as the second evidence, let the polarity score of the word w at time-span T_2 be a negative value $-1 \leq r < 0$. The BPAs for this evidence can be defined as follows:

$$\begin{aligned}
&m_{w^2}(\phi) = 0\\
&m_{w^2}(positive) = 0\\
&m_{w^2}(negative) = |r|\\
&m_{w^2}(positive\ \ or\ \ negative) = 1 - |r|
\end{aligned} \tag{4.20}$$

Note that, according to the Dempster-Shafer theory, the first evidence only supports the positivity of w and does not say anything about its negativity. So the value $1-m_{w^1}(positive)$ reflects the amount of uncertainty that we have about the status of w at time T_1, i.e. $m_{w^1}(positive\ \ or\ \ negative)$. In other words, if the first evidence is flawed, w could still be either positive or negative. The same is true for the second evidence. The uncertainty state of the Dempster-Shafer theory is the major characteristic that differentiates this theory from other theories like Bayesian probability theory.

Given the above two (or more) evidences about the polarity of w, the Dempster-Shafer *combination rule* can be used to obtain the combined evidence about the polarity of w up to time T_2, i.e. $m_{w@2}(A), \forall A \in 2^{\Theta}$. The value of $m_{w@2}(A)$ is computed by combining w's polarity evidences at times T_1 and T_2: $\{m_{w^1}(.),\, m_{w^2}(.)\}$. The Dempster-Shafer rule for combining the above two evidences is as follows:

$$m_{w@2}(A) = \frac{\sum\limits_{X \cap Y = A} m_{w^1}(X) * m_{w^2}(Y)}{1 - \sum\limits_{X \cap Y = \phi} m_{w^1}(X) * m_{w^2}(Y)} \tag{4.21}$$

The above Equation measures the amount of agreement between the two evidences. The denominator is the normalization factor that ensures that $m_{w@2}(A)$ is a BPA. As the numerator shows, the combination rule focuses only on those proposition that both evidences support.

The generalized Dempster-Shafer combination rule for combining j evidences can be defined as follows: Let $m_{w@j}(A)$ indicates the combined evidence about the polarity of w up to time T_j. The value of $m_{w@j}(A)$ can be computed by combining w's polarity evidences obtained at times $T_1, ..., T_j$, i.e. $\{m_{w^1}(.),\, ...,\, m_{w^j}(.)\}$. The Dempster-Shafer rule for combining these j evidences is as follows:

$$m_{w@j}(A) = \frac{\sum\limits_{\cap X_i = A} \prod\limits_{1 \le i \le j} m_{w^i}(X_i)}{1 - \sum\limits_{\cap X_i = \phi} \prod\limits_{1 \le i \le j} m_{w^i}(X_i)} \tag{4.22}$$

We use the above belief values, $m_{w@j}(A), \forall A \in 2^{\Theta}$, to compute the time accumulated polarity score of w^j up to time-span T_j, $Pol(w@j)$, as follows:

$$\begin{aligned} Pol(w@j) = I * \max[&m_{w@j}(positive), \\ &m_{w@j}(negative), \\ &m_{w@j}(positive\ \ or\ \ negative)] \end{aligned} \tag{4.23}$$

where

$$I = \begin{cases} +1, & \text{if } m_{w@j}(positive) = max \\ -1, & \text{if } m_{w@j}(negative) = max \\ 0, & \text{if } m_{w@j}(positive\ or\ negative) = max \end{cases} \tag{4.24}$$

The value of $m_{w@j}(positive\ or\ negative)$ indicates the amount of uncertainty that we have about the polarity of w at T_j. Therefore, when this value is maximum, we avoid tagging w as a positive or negative opinion word at T_j and let the future time-spans determine its polarity. We consider any candidate opinion word with a non-zero $Pol(w@j)$ as a new opinion word.

This formulation can tolerate the noise of the polarity scores obtained at different times from the co-occurrence patterns. It can also capture the changes in the polarity of the words based on the observed evidences.

4.6.4 Experiments

We first explain the datasets we used in the experiments and some parameter settings. We then evaluate our approach based on (a) the quality of new opinion words, (b) the performance of our approach in polarity inference, and (c) the utility of the new opinion word in sentiment classification of reviews.

4.6.4.1 Data and Settings

We made use of the three opinion lexicons used in Section 4.5 to supply the seeds. We used a large dataset of `Amazon.com` reviews gathered by Jindal

	T_{10}	T_{11}	T_{12}	T_{13}
top 5, positive				
1	mettle	bros	offerred	healings
2	topnotch	excellance	worshiped	sticklers
3	amassed	muss	soulfully	ubers
4	reigning	earthshattering	excellance	exsperiance
5	fab	soulfully	ubers	dimmu
top 5, negative				
1	irks	gutteral	targetted	plagerized
2	groaner	molested	regretably	dumbledore
3	doomy	derailed	rackets	worsened
4	umph	errie	sqeaky	gimmie
5	maggots	dodged	ozzfest	lamer

Table 4.6: Top 5 detected words in the latest four time-spans.

and Liu (2008) to perform our experiments. This dataset contains more than 5.8M reviews dated from Jan 1996 to May 2006. We only performed the experiments on the reviews from Jan 2000 to May 2006 because there are very few reviews available before 2000 in this dataset. We divided this data into 13 time-spans at 6-monthly time intervals (except for reviews from 2006 that only cover five months, Jan to May). For sentiment classification of reviews, we balanced the data on the positive and negative reviews.

In Equation 4.14, we set the parameter ζ to 3, as suggested by (Islam and Inkpen, 2008), and z to the average sentence length in the above corpus. All the parameters of the optimization framework are set to the values of the best performing system as reported in Section 4.5.

In all the subsequent experiments, we used the two-tailed paired t-test $p < 0.01$ for significance testing.

4.6.4.2 Quality of New Opinion Words

Table 4.6 shows the top five positive and negative words learned by our method for the latest four time-spans. As it is shown, some misspelled seeds like *excellance*, *errie* and *regretably* etc as well as urban words like *fab* (*fabulous*), *topnotch* (*excellent*) and *lamer* (*stupid person*) etc have been accurately detected.

We also quantitatively evaluated the quality of the discovered new opinion words based on the percentage of such words that are indeed opinion. For this purpose, we manually annotated them as opinion or non-opinion. Hence, the quality of our method for finding new opinion words can be measured as follows:

$$Quality = \frac{N_{tagged}}{N_{total}} \tag{4.25}$$

where N_{tagged} is the number of words labeled as correct new opinion words and N_{total} is the total number of opinion words found at each time-span. Note that there could be overlap between the new opinion words found at different time-spans.

Table 4.7 shows the results. The average quality is 68.76%. The annotation shows that our method accurately detected many misspelled seeds as opinion words. In addition, the extracted non-opinion words were mainly the words that frequently co-occurred with one type of seeds (e.g. negative) and rarely co-occurred with the other type. These words were assigned high polarity scores by the optimization framework and consequently labeled as opinion by our system. Such words, though non-opinion, are good polarity indicators. For example, the word "*Dumbledore*" was labeled as negative by our system as it co-occur with many negative seeds but not with positive

Time	$\mathbf{N_{total}}$	$\mathbf{N_{tagged}}$	**Quality**
T_1	292	214	73.29
T_2	1000	695	69.50
T_3	1710	1150	67.25
T_4	2361	1580	66.92
T_5	3031	2042	67.37
T_6	3507	2392	68.21
T_7	4097	2830	69.07
T_8	4709	3239	68.78
T_9	5303	3653	68.89
T_{10}	5911	4156	70.31
T_{11}	6560	4571	69.68
T_{12}	7238	4980	68.80
T_{13}	7746	5096	65.79
Average	-	-	**68.76**

Table 4.7: Quality of the new opinion words based on manual annotation.

ones. We noticed that this word refers to a character in the "Harry Potter" series who received many negative opinions against his positive role in the movie.

4.6.4.3 Performance of Polarity Inference

For this evaluation, we considered part of the seed dataset as the test data and the rest as training data, and evaluated how the time accumulated polarity improves the polarity of the test seeds computed at different times. We only considered seeds that occur more than 10 times in our review corpus (i.e. 2500+ seeds) and conducted 5-fold cross validation over the seed dataset based on the following evaluation measures:

$$
\begin{aligned}
Precision &= \frac{N_{correct}}{N_{labeled}} \\
Recall &= \frac{N_{correct}}{N_{seed}} \\
F1 &= \frac{2 * Precision * Recall}{Precision + Recall}
\end{aligned}
\qquad (4.26)
$$

where $N_{correct}$ is the number of test seeds that were assigned correct polarity (either positive or negative), $N_{labeled}$ is the number of test seeds that were assigned non-zero scores, and N_{seed} is the total number of test seeds.

Figure 4.4 shows the results. At each time T_j, *Spcf* indicates the polarity inference performance of the optimization framework, Equation 5.4, and *Acm* indicates the performance of the time accumulated polarity computed from the combination of all the polarity evidences obtained up to time T_j, Equation 4.23.

The performances of *Acm* and *Spcf* are the same at the beginning as the polarity score at T_1 is the only available evidence. As Figure 4.4 shows, the performance of *Acm* increases through time with greater improvements in the latter times. This is because of the availability of more polarity evidences about the test seeds for *Acm* as time passes. However, the performance of *Spcf* depends on the co-occurrence patterns obtained at each time and as Figure 4.4 shows varies greatly through time. *Acm* significantly outperforms the *Spcf* method by 5.8% on average in F1 score. The difference between the two methods is significant for all T_i, for $i \geq 5$.

4.6.4.4 Performance of Sentiment Classification

In this Section, we study how the learning of new opinion words through time affect the performance of sentiment classification (SC) of reviews. For this purpose, similar to (Choi and Cardie, 2009), we designed a word-

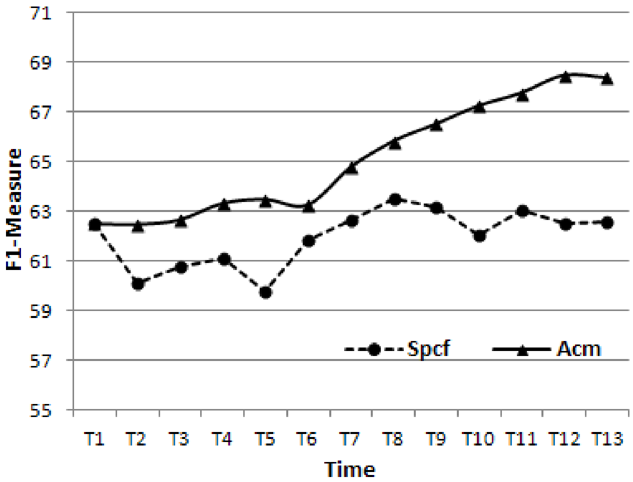


Figure 4.4: Polarity inference through time.

matching-based sentiment classifier. Note that, we do not use any popular classifier (like SVM or Naive Bayes) here in order to emphasize that the performance improvements come mainly from the quality of the new opinion words. However, we used the same set of manually created rules introduced in Section 4.5 to handle negations. We expect the performance of our word-matching-based SC to be better when we use opinion words with higher quality.

Given a review, the word-matching-based sentiment classifier computes a sentiment score for the review by summing up the polarity scores of the opinion words that appear in the review. A positive sentiment score indicates a positive review, and a negative one indicates a negative review (Choi and Cardie, 2009).

Figure 4.5 shows the performance of SC using seeds and new opinion words. *Seeds* as the baseline indicates the SC performance when we only use seeds to classify reviews of each time-span, while *Seeds+NOW_OPT* and *Seeds+NOW_AC* respectively show the SC performance when we use both seeds and all the new opinion words we learned up to time T_i to classify the reviews of the same time-span T_i. The difference between the

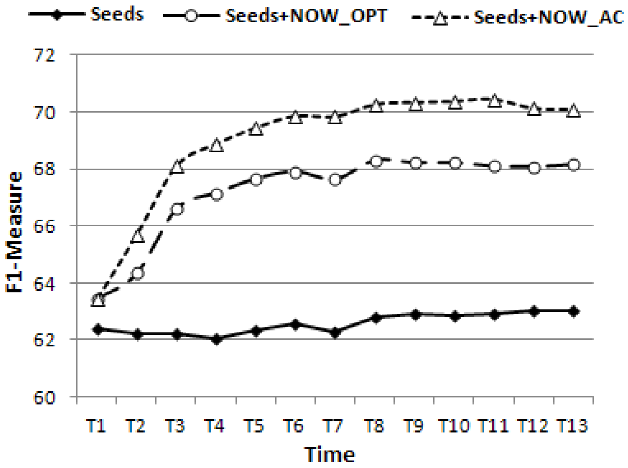


Figure 4.5: Effect of polarity inference on sentiment classification.

two methods is that *Seeds+NOW_OPT* uses the most recent polarity score of each new opinion word (obtained from Equation 5.4) to perform SC, whereas *Seeds+NOW_AC* utilizes the time accumulated polarity score for this purpose, Equation 4.23. The results show that both *Seeds+NOW_OPT* and *Seeds+NOW_AC* significantly outperform *Seeds* for all T_i, $i \geq 2$. This reflects the utility of the new opinion words found for SC. In addition, *Seeds+NOW_AC* significantly outperforms *Seeds+NOW_OPT* for all T_i, $i \geq 5$. This shows the effectiveness of the time accumulated polarity scores obtained by Dempster-Shafer combination rule.

We also studied the effect of learning more recent new opinion words on the performance of SC. For this purpose, at each time, we used seeds and the current opinion words to perform SC on the current and future reviews. Figure 4.6 shows the results. Each $SC - Ti$ indicates the performance of SC when we use both seeds and the new opinion words that we learned up to time T_i to perform SC on the current and future reviews, i.e. the reviews of T_k, $\forall k \geq i$. Here, we use the time accumulated polarity scores.

The results show that the SC performance improves as time passes. In other words, each $SC - Ti$ improves the SC performance over the earlier

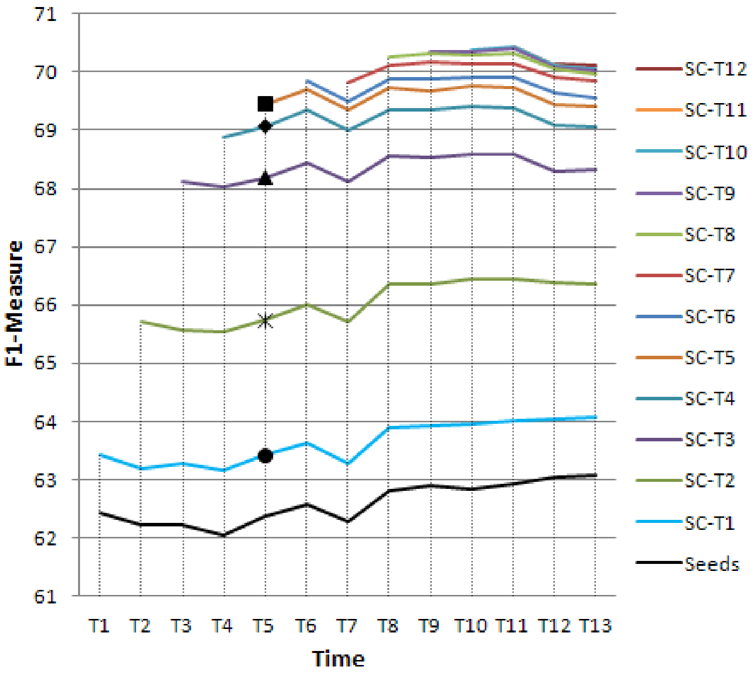


Figure 4.6: Performance of sentiment classification through time (best seen in color).

$SC - Tk$, $\forall k \leq i$. For example consider the time T_5. As highlighted in Figure 4.6, the performance of SC using the new opinion words we learned up to time T_5, i.e. $SC - T5$, is greater than the performance of SC using the new opinion words we learned at earlier times, i.e. $SC - T1$ to $SC - T4$. In other words, the improvement is greater when the sentiment classifier utilizes more recent new opinion words. This is because, in the more recent times, the classifier receives a greater number of new opinion words with more accurate polarity scores due to the existence of more polarity evidences.

4.7 Summary

In this Chapter, we focused on mining slang and urban opinion words and phrases from user generated contents (UGCs). Such opinion entities are useful for different tasks of sentiment analysis like sentiment classification and review mining. We proposed to utilize the opinion words with already known polarities (seeds) to extract a set of candidate opinion entities (or significant entities) from UGC. We then formulated the polarity inference task as a semi-supervised learning task in the graph context where the seeds and significant entities were modeled as the graph nodes. The graph connectivity information was then used to infer the polarity of significant entities. Our method is able to utilize both labeled and unlabeled data to learn the polarity of the entities and do not require dictionary-based relations (such as synonym, antonym, or hyponym) to construct the graph. We experimentally showed that our approach is effective in detecting new opinion entities and inferring their polarities. We also showed that learning from both labeled and unlabeled data play a crucial role in inferring the polarity of candidate opinion entities. For further analysis, we focused on time as another important factor for sentiment terminology mining. We proposed the *interchangeability* concept to find high quality new opinion words through time. We then utilized Dempster-Shafer combination theory to obtain a time accumulated polarity for each new opinion words through time. The time accumulated polarity was obtained by combining the available evidences about the polarity of the words. We showed that the time accumulated polarity better reflects the polarity of the opinion words than the polarity obtained at each particular time. We experimentally showed that mining more recent new opinion words result in a greater improvement in the performance of sentiment classification.

Chapter 5

Intelligent Data Harvesting and Temporal Topic Modeling

In this Chapter we explain our approaches for harvesting relevant contents about a given organization and modeling its topics through time. We also elaborate the performances of the proposed algorithms. In this Chapter, we may use the terms micro-post, tweet, streaming data interchangeably.

5.1 Intelligent Data Harvesting

5.1.1 Mining Dynamic Keyword

The *keyword miner* component (see Figure 3.2) extracts the dynamic keywords about the target organization from the recently seen micro-posts at each point of time. The dynamic keywords are then utilized by our classifier to judge the relevance of incoming micro-post to the organization.

At each point of time, we define the dynamic keywords as the keywords that represent the current discussions about the target organization. To identify such keywords, suppose we have two sets of *foreground* ($\mathcal{S}^t_{for}$)

and *background* ($\mathcal{S}^t_{bak}$) tweets at each point of time t. Let $\mathcal{S}^t_{for}$ includes the recently-seen *relevant* tweets posted in a short time window of length T, i.e. $[t-T, t]$, while $\mathcal{S}^t_{bak}$ includes the *irrelevant* tweets identified in the same time window, $[t-T, t]$. In addition, let $\mathcal{W}^t = \{w_1, w_2, ...\}$ be the vocabulary set obtained from $\mathcal{S}^t_{for}$. We define the dynamic keywords as a subset of $\mathcal{W}^t$ words that best represent the current relevant discussions about the organization. Our aim is to extract such keywords from $\mathcal{W}^t$.

For this purpose, we identify the terms of $\mathcal{W}^t$ that have different distributions in $\mathcal{S}^t_{for}$ and $\mathcal{S}^t_{bak}$. A significant difference between the two distributions of a term $w_i \in \mathcal{W}^t$ in $\mathcal{S}^t_{for}$ and $\mathcal{S}^t_{bak}$ signals that w_i better represents one of these sets, either the foreground (relevant) or the background (irrelevant) set. In terms of statistical, given the two distributions of $w_i \in \mathcal{W}^t$ in $\mathcal{S}^t_{for}$ and $\mathcal{S}^t_{bak}$, can we disprove, to a certain level of significance, the null hypothesis that the two distributions are drawn from the same distribution function? Disproving the null hypothesis for a term signifies that the term has different importance in the two foreground and background sets. Thus, those significantly important terms that have rising frequency in $\mathcal{S}^t_{for}$ can potentially represent the dynamic keywords.

There are different approaches to compare two distributions (Mood and Graybill, 1963; Strang, 1986). Here we utilize the chi-squared test as its calculation is fast and suitable for rapidly evolving social media content. To derive this value for each $w_i \in \mathcal{W}^t$, we use the following Equation:

$$\chi^2_i = \begin{cases} \dfrac{(f_i - b_i)^2}{b_i} + \dfrac{[(100 - f_i) - (100 - b_i)]^2}{100} & \text{if } f_i > b_i \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

where f_i and b_i are the normalized term frequency values of w_i in the foreground and background sets respectively and are computed as follows:

$$f_i = 100 * \frac{w_i^{for}}{\sum\limits_{\forall i} w_i^{for}} \tag{5.2}$$

$$b_i = 100 * \frac{w_i^{bak}}{\sum\limits_{\forall i} w_i^{bak}} \tag{5.3}$$

where w_i^{for} and w_i^{bak} is the term frequency of w_i in $\mathcal{S}_{for}^t$ and $\mathcal{S}_{bak}^t$ respectively. Equation 5.1 assigns higher weights to the terms that frequently occur in $\mathcal{S}_{for}^t$, but rarely occur in $\mathcal{S}_{bak}^t$. Thus, Equation 5.1 only takes into account the words $w_i \in \mathcal{W}^t$ with $f_i > b_i$ and assign zero weight to those with $f_i \leq b_i$.

We rank the terms based on their χ^2 values and consider those with χ^2 value greater than ϵ (where $\epsilon = 2.706$ which corresponds to $p = 0.10$ significant level of t-test) as the *dynamic keywords*.

It may happen that a term in $\mathcal{S}_{for}^t$ has a term frequency of zero in $\mathcal{S}_{bak}^t$ that results in division by zero in Equation 5.1. We adapt *add-one* smoothing method (i.e. increasing the term frequencies by 1) to prevent division by zero. Furthermore, we only take into account the words of $\mathcal{S}_{for}^t$ that have a term frequency greater than a predefined threshold[1]. This is to prevent the domination of the low frequent terms.

5.1.2 Mining Organization Users

Given the user graph of the organization, the *user miner* component ranks users with respect to the target organization. A good ranking algorithm should rank the more *active* and *influential* users of the organizations in the higher ranks, while, in case of ambiguous organizations, discard the users of the other organizations.

[1] This threshold is set to 10 in our experiments

We define an active user of an organization as the one who sends many relevant micro-posts about the organization, has many followers within the organization, and initiates major discussions about the organization. The combination of these measures can be used to rank the users of the target organization with high accuracy. Note that we only consider the number of followers within the organization. This is because the *total* number of followers is only a good measure to identify generally-influential users with large profiles (Bakshy and Hofman, 2011; Lee et al., 2010; Kwak et al., 2010). However, such users may have little influence with respect to the target organization.

Based on the above discussion, a *key-user* of an organization is an active user who regularly tweets about the organization (number of relevant micro-posts), has many followers within the organization (number of followers), and initiates major discussions (number of re-tweets) about the organization.

Let G^t be the user graph of the target organization at time t (See Figure 3.2) and $\mathcal{U}^t = \{u_1, ..., u_m\}$ be the set of nodes in G^t. We compute the score for each user $u_i \in \mathcal{U}^t$ based on the following Equation at time t:

$$W^t_{u_i} = sign(r^t_{u_i} - I^t_{u_i}) \log \left(\tau \frac{|r^t_{u_i} - I^t_{u_i}|}{\sum\limits_j r^t_{u_j} + I^t_{u_j}} * \varphi \frac{f^t_{u_i} + 1}{\sum\limits_j f^t_{u_j}} * \omega \frac{q^t_{u_i} + 1}{\sum\limits_j q^t_{u_j}} \right) \tag{5.4}$$

where $r^t_{u_i}$ is the total number of *relevant* tweets posted by u_i up to time t, $I^t_{u_i}$ is used in case of ambiguous organizations and indicates the total number of irrelevant tweets that contain the acronym of the target organization posted by u_i up to time t (in case of non-ambiguous organization $I^t_{u_i}$ is 0 at any time). This parameter penalizes users of the other organizations that share the same acronym with the target organization. The variable $f^t_{u_i}$ is the total number of u_i's followers who exist in $\mathcal{U}^t$, $q^t_{u_i}$ is the total number of u_i's relevant tweets that have been re-tweeted by other users up to time t, $sign(.)$ is the sign function, and τ, φ and ω are weighting parameters such that $\tau + \varphi + \omega = 1$[2].

The above Equation ranks the user based on the aforementioned three criteria. The users are ranked based on their influence scores and the top K users are considered as the key-users of the organization at time t. These users are passed to the *org key-user* crawler to be monitored.

5.1.3 Relevant Tweet Detection

As we mentioned before, one of the challenges in mining the sense of organizations in social media is real-time discrimination of relevant and irrelevant micro-posts for potentially ambiguous organizations as large data streams in through time.

As we discussed before, in case of ambiguous organization, the content information alone may simply relate micro-posts and consequently their topics to wrong organizations. For example, consider two university-based ambiguous organizations such as *National Union of Students* (NUS) and *National University of Singapore* (NUS). These organizations share many similar terminology in general and therefore the content information alone may not be an effective mean to discriminate their relevant data especially when we notice that the micro-posts are usually short and provide little information for discrimination. As discussed before, user information can help the classification task for ambiguous organizations.

We propose a high quality classifier by combining the content (i.e.

[2] We empirically set these parameters as follows in our experiments: $\tau = 0.50$, $\varphi = 0.25$, and $\omega = 0.25$.

dynamic keywords and micro-posts) and user information of the input data. We show that this classifier can discriminate relevant micro-posts from irrelevant ones with high accuracy.

5.1.3.1 Learning Content-Based Classifier

Our aim here is to assign a relevance score to each input data based on its content similarity with the current discussions about the organization. For this purpose, at each point of time, we utilize the dynamic keywords (mined in Section 5.1) as such keywords are good indicators of the current discussions about the organization.

Formally, let $\mathcal{W}^t = \{w_1, ..., w_m\}$ of arbitrary size m contains the dynamic keywords at time t. Also, as before, let ($\mathcal{S}^t_{for}$) be the set of recently-seen relevant tweets over the time window $[t - T, t]$ and ($\mathcal{S}^t_{bak}$) be the irrelevant tweets in the same time-span $[t - T, t]$ where t is the current time. We utilize $\mathcal{W}^t$ as the classification features and $\mathcal{S}^t_{for} \cup \mathcal{S}^t_{bak}$ as training data to discriminate the input streaming data into relevant and irrelevant sets. The dynamic keywords provide a fast way to prune the huge amount of irrelevant input data as they stream in.

We take a binary weighting schema to weight the features for each input tweet. That is, given a tweet, we create its m-dimensional feature vector using $\mathcal{W}^t$ as follows: the ith entry of the feature vector is set to 1, if the tweet contains w_i, and 0 otherwise. Any input data with a zero feature vector is regarded as irrelevant by default. At the end of this process, each test tweet will be assigned a relevance score which represents the content-based relevance score of the tweet.

As the classification approach, we do experiments with SVM classifier which is an effective classifier on textual data. We utilize the implementation of the Weka toolkit (Hall et al., 2009) with default parameters for this purpose. As the classification baseline, we consider Unigram and Bigram features obtained form the combination of $\mathcal{S}^t_{for}$ and $\mathcal{S}^t_{bak}$ tweets.

Note that we preprocess the input data based on some heuristic rules before the classification. For example the tweets posted by the known accounts of the target organization that contain a fixed keyword are treated as relevant data. Also the tweets of length smaller than three words are ignored as such tweets have low content information.

Furthermore, we consider the dynamic keywords as the only classification features while we include the irrelevant tweets, $\mathcal{S}^t_{bak}$, to the training set. This helps our classifier to also learn the sets of terms/features that may represent the irrelevant data even though the individual features are all extracted from the relevant data.

5.1.3.2 Combining Content and User Information

Given the context (i.e. content and user information), we can make a final judgment about the relevance of the tweet to the target organization. However, because of our system design (see Figure 3.2), we only need to utilize the user information for the data we obtained from the *fixed keyword* crawler. This is because the data crawled from the other two crawlers (*known account* and *org key-user* crawlers) come from the users who already have high relevance scores to the target organization and therefore we just need to ensure the relevance of their content.

Formally, given a test tweet s^t_i at time t obtained from the *fixed keyword* crawler, we determine the final score of the tweet by the linear combination of its content and user score as follows:

Algorithm 5.1. Classification at time t

```
Input: $\mathbf{Q}^t$: input test data,
       $T$: learning time interval,
       $\alpha$: learning parameter.

Output: $\mathbf{L}$: classification result.

learn SVM classifier with labeled data seen in $[t-T,t]$
for each $s_i^t \in \mathbf{Q}^t$ do
    use the classifier to compute $C_{s_i}$
    if $s_i^t$ contains a fixed keyword
        compute $W_{u_i}^t$
        $L_i = \alpha * C_{s_i} + (1-\alpha) * W_{u_i}^t$
    else
        $L_i = C_{s_i}$
end for
```

$$L_i = \alpha * C_{s_i} + (1-\alpha) * W_{u_i}^t \tag{5.5}$$

whereas for a tweet obtained from the other crawlers we determine its final score by solely considering its content relevance score as follows:

$$L_i = C_{s_i} \tag{5.6}$$

where $C_{s_i} \in [-1,1]$ indicates the content-based relevance score of s_i^t and $W_{u_i}^t \in [-1,1]$ indicates the relevance score of u_i as the author of s_i^t (see Equation (5.4)). The parameter α controls the contribution of each of the above scores in labeling the tweet. We learn this parameter using our development data. We expect α to be smaller than 0.5 because if a tweet contains a fixed keyword, the user relevance score is a very important measure to judge the relevance of the tweet.

Any incoming tweet with $L_i > 0$ is considered as relevant, and the rest as irrelevant. The relevant tweet will be added to the relevant tweet repository which will be then utilized in the next iterations. Algorithm 5.1

illustrates our online classifier. We analyze the effect of the length of the time interval T on the classification performance.

5.2 Mining Evolving and Emerging Topics

The *topic miner* component utilizes relevant tweets to mine the evolving and emerging topics for the target organization. As depicted in Figure 3.2, these tweets are taken from the *relevant tweet repository.* We propose an online sparse coding algorithm to incrementally learn the topics for the target organization through time.

5.2.1 Streaming Input Data

Assume that, at each point of time t, we receive a set of relevant tweets $\mathbf{S}^t = \{s_1, s_2, ..., s_{n^t}\} \in \mathbb{R}^{m*n^t}$ where n^t is the number of relevant tweets at time t and m is the size of vocabulary. We represent each $s_i \in \mathbb{R}^m$ as a term vector of length m weighted by the standard Term Frequency (TF) and Inverted Document Frequency (IDF) as follows:

$$s_{ij} = \frac{\log(TF(i,j)) * \log(IDF(j))}{C} \tag{5.7}$$

where C is the normalization factor, $TF(i,j)$ indicates the frequency of the term w_j in s_i, and $IDF(j)$ indicates the inverted document frequency of w_j.

We should note that, as new posts are received and new terms are introduced, the vocabulary size (m) increases. However, here for simplicity, we assume a global vocabulary containing a total number of m terms. The extension to the case where the vocabulary size increases can be simply

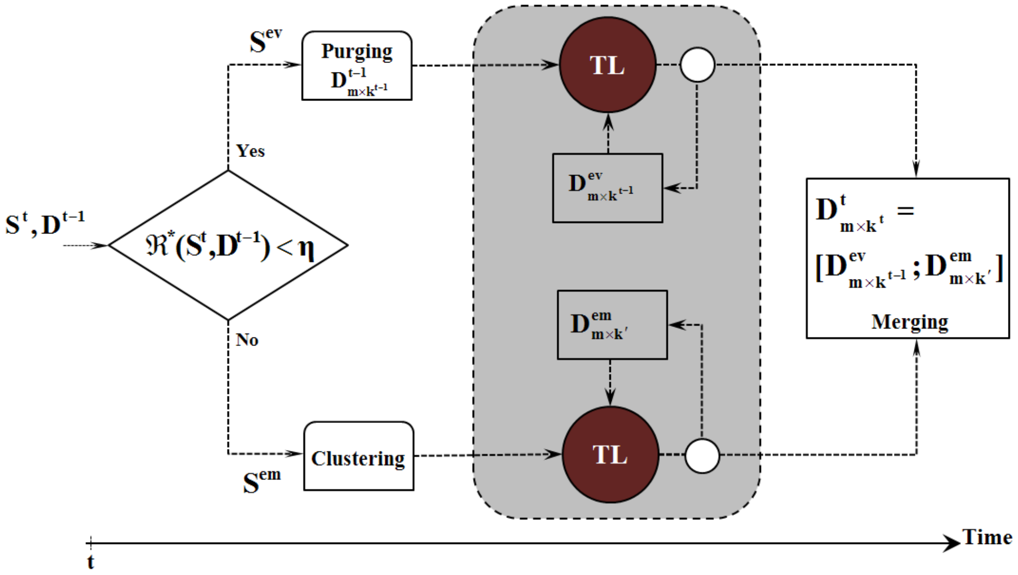

S^{ev}
Purging
$D^{t-1}_{m \times k^{t-1}}$
TL
Yes
$D^{ev}_{m \times k^{t-1}}$
S^t, D^{t-1}
$\Re^*(S^t, D^{t-1}) < \eta$
No
$D^{em}_{m \times k'}$
Clustering
S^{em}
TL
$D^t_{m \times k^t} =$
$[D^{ev}_{m \times k^{t-1}}; D^{em}_{m \times k'}]$
Merging
t
Time

Figure 5.1: Learning evolving and emerging topics at time t; the circles represent the topic learning (TL) process

handled by adjusting the size of the related matrices by automatic zero-padding.

5.2.2 Live Topic Learning

As aforementioned, our topic modeling problem is to identify the topics as the relevant tweets stream in. At each point of time, such tweets can be either matched with the already known topics or can potentially represent new emerging topics for the organization.

Let the non-negative matrix $\mathbf{D}^{t-1} \in \mathbb{R}^{m*k^{t-1}}$ represents the k^{t-1} topics found up to time $t-1$ for the target organization and $\mathbf{S}^t \in \mathbb{R}^{m*n}$ indicates the relevant incoming tweets at time t. Given $\mathbf{D}^{t-1}$ and $\mathbf{S}^t$, the problem is to determine the topic matrix at time t, i.e. $\mathbf{D}^t \in \mathbb{R}^{m*k^t}$. This matrix comprises of the smooth evolution of the k^{t-1} previously known topics (*evolving* topics) as well as the new topics identified at time t (*emerging* topics).

Let $\mathbf{S}^{ev} \in \mathbb{R}^{m*n^{ev}}$ indicates the tweets of $\mathbf{S}^t$ that can be *matched*, to a certain level of significance, with a topic in $\mathbf{D}^{t-1}$, and $\mathbf{S}^{em} \in \mathbb{R}^{m*n^{em}}$ be the rest of $\mathbf{S}^t$s' tweets (these tweets can potentially form the *emerging* topics) where $n^t = n^{ev} + n^{em}$. We explain the way we decompose $\mathbf{S}^t$ into these two matrices in the next section.

As depicted in Figure 5.1, given $\mathbf{S}^{ev}$, $\mathbf{S}^{em}$, and $\mathbf{D}^{t-1}$, we need to solve the following two sub-problems to obtain $\mathbf{D}^t$: (a) how to learn the evolving topics using $\mathbf{S}^{ev}$ and $\mathbf{D}^{t-1}$ (we indicate the evolving topics by $\mathbf{D}^{ev} \in \mathbb{R}^{m*k^{t-1}}$), and (b) how to learn the new emerging topics using $\mathbf{S}^{em}$ (we indicate the emerging topics by $\mathbf{D}^{em} \in \mathbb{R}^{m*k'}$). The topic matrix $\mathbf{D}^t \in \mathbb{R}^{m*k^t}$ where $k^t = k^{t-1} + k'$ can then be achieved by vertical concatenation of $\mathbf{D}^{ev}$ and $\mathbf{D}^{em}$.

We consider the following two constraints learn the topic matrix $\mathbf{D}^t$:

- *Temporal Continuity* constraint: This requirement constraints $\mathbf{D}^{ev}$ to be a smooth evolution of $\mathbf{D}^{t-1}$, and

- *Sparse Matching* constraint: This constraint indicates that each tweet s_i can only represent a “few” topics.

This first constraint is to prevent dramatic changes in the evolving topics in two consecutive time stamps, whereas the second constraint is due to the limited length of the tweets. Similar idea of considering a single topic for short texts has been used before (Gruber, Weiss, and Rosen-Zvi, 2007; Zhao et al., 2011). In fact, tweets are limited to 140 characters; this space is too short to be used for writing about several topics.

Based on the above requirements, the evolving topic matrix $\mathbf{D}^{ev} \in \mathbb{R}^{m*k^{t-1}}$ can be learned by minimizing the following optimization problem (Wang, Li, and Knig, 2011; Liu, Latecki, and Yan, 2010; Mairal et al., 2009; Gu and Zhou, 2009):

$$\begin{aligned}&(\mathbf{D}^{ev}, \mathbf{X}^{ev}) = \arg\min_{\mathbf{D},\mathbf{X}} \parallel \mathbf{S}^{ev} - \mathbf{D}\mathbf{X} \parallel_F^2 + \mu \parallel \mathbf{D} - \mathbf{D}^{t-1} \parallel_F^2 + \lambda \parallel \mathbf{X} \parallel_1 \\ &s.t.: \ \mathbf{X} \geq 0, \, \mathbf{D} \geq 0, \ \parallel d_j \parallel_2^2 \leq 1 \ \forall j \in \{1...k^{t-1}\}\end{aligned} \tag{5.8}$$

where $\mathbf{X}^{ev} \in \mathbb{R}^{k^{t-1} * n^{ev}}$ is the weight matrix and $\lambda \in [0,1]$ and $\mu \in [0,1]$ are the learning parameters. The first term in the above Equation is the reconstruction error, while the second and the third terms represent the two aforementioned constraints respectively. The above topic learning process optimizes the matrix $\mathbf{D}^{ev}$ with respect to $\mathbf{D}^{t-1}$ and $\mathbf{S}^{ev}$. Note that, in this step, no new topic is introduced.

It is well known that the ℓ_1 regularization produces sparse weight matrix, $(\mathbf{X})$, and is robust to irrelevant features (Mairal et al., 2009; Wang, Li, and Knig, 2011). Here, for interpretability, we put the positivity constraints on $\mathbf{X}$ and $\mathbf{D}$ and normalize each column of $\mathbf{D}$ so that it resembles a probability distribution of terms over the corresponding topics.

In contrast to the evolving topics, the emerging topics are totally new and there is no prior information about the number of emerging topics. Therefore, we utilize the standard NMF algorithm to find an initial set of clusters from $\mathbf{S}^{em}$. We then find the optimum value for $\mathbf{D}^{em}$ as follows:

$$\begin{aligned}&(\mathbf{D}^{em}, \mathbf{X}^{em}) = \arg\min_{\mathbf{D},\mathbf{X}} \parallel \mathbf{S}^{em} - \mathbf{D}\mathbf{X} \parallel_F^2 + \lambda \parallel \mathbf{X} \parallel_1 \\ &s.t.: \ \mathbf{X} \geq 0, \, \mathbf{D} \geq 0, \ \parallel d_j \parallel_2^2 \leq 1 \ \forall j \in \{1...k'\}\end{aligned} \tag{5.9}$$

where $\mathbf{X}^{em} \in \mathbb{R}^{k^{t-1} * n^{em}}$ is the weight matrix.

Figure 5.1 depicts the overall procedure of learning topics at each point of time. Note that the above two processes (learning $\mathbf{D}^{ev}$ and $\mathbf{D}^{em}$)

can be performed in parallel to speed up the overall learning process. The purging process in Figure 5.1 will be explained in Section 5.4.4.

5.2.3 Decomposition of Streaming Data

Given the input matrix $\mathbf{S}^t$ and the topic matrix $\mathbf{D}^{t-1}$, we need to decompose $\mathbf{S}^t$ into $\mathbf{S}^{ev}$ and $\mathbf{S}^{em}$ matrices. For this, we find the best representation of each $s_i \in \mathbf{S}^t$ in terms of $\mathbf{D}^{t-1}$ as follows:

$$\begin{aligned} &x_i = \arg\min_x \parallel s_i - \mathbf{D}^{t-1}x \parallel_2^2 + \lambda \parallel x \parallel_1 \\ &s.t.\!: x \geq 0 \end{aligned} \tag{5.10}$$

The resultant vector $x_i \in \mathbb{R}^{k^{t-1}}$ indicates the already known topics that best represent the input vector s_i. Using this vector, we compute the representation error of s_i on $\mathbf{D}^{t-1}$ (what we call *residual* error) as follows:

$$\mathcal{R}^*(s_i, \mathbf{D}^{t-1}) = \parallel s_i - \mathbf{D}^{t-1}x_i \parallel_2^2 + \lambda \parallel x_i \parallel_1 \tag{5.11}$$

Based on the value of the residual error, the matrix $\mathbf{S}^t$ can be decomposed into the two matrices as follows:

- $\mathbf{S}^{ev}$: contains all $s_i \in \mathbf{S}^t$ with a residual error equal to or smaller than a chosen threshold η, and
- $\mathbf{S}^{em}$: includes other inputs, i.e. all $s_j \in \mathbf{S}^t$ with a residual error greater than η.

5.2.4 Purging Trivial Topic

As time passes, some topics may become old and no more discussions arrive about them. Such topics can be safely removed from the topic matrix $\mathbf{D}^t$. There are different approaches to accomplish this. For example one can directly *remove* the non-active topics or *replace* them with a randomly selected input data (Mairal et al., 2009). We here apply the first approach as it better suits our need for keeping the size of the learned topics manageable.

To do so, for each topic, we store the most recent time that the topic is selected as the *dominant* topic for an input tweet. This time is used as a measure to purge the topics. The dominant topic of each s_i is the topic that has the greatest matching score with s_i as compared to all the other topics, i.e. the topic d_j such that $j = \arg\max_j x_{ij}$ where x_i is obtained from Equation 5.10. We should note that the matching score between each d_j and each s_i is determined by the ijth entry of the weight matrix $\mathbf{X}$, i.e. x_{ij}, see Equations 5.8 and 5.9.

In our setting, all the topics that have not been selected as a dominant topic for a reasonably large amount of time (e.g. past 24 hours) are considered as non-active and are removed from $\mathbf{D}^t$.

5.2.5 Optimization Algorithms

In this Section, we explain a fast online approach to solve the optimization problem of Equation 5.8 (the same approach can be used to solve Equation 5.9). This optimization problem is in general non-convex, but, it has been shown that, if one of the variables, either $\mathbf{D}$ or $\mathbf{X}$ is known, optimization with respect to the other variable will be convex (Mairal et al., 2009; Liu, Latecki, and Yan, 2010). Therefore, a general solution is to iteratively optimize the objective function by alternatively optimizing with respect to $\mathbf{D}$ and $\mathbf{X}$ while holding the other fixed.

If $\mathbf{D}$ is fixed, i.e. we set it to the value of its previous time stamp,

$\mathbf{D}$=$\mathbf{D}^{t-1}$, then the problem is equivalent to an ℓ_1-regularized least square problem and can be efficiently solved by least angle regression (LARS) method (Efron et al., 2004; Fraley and Hesterberg, 2009) or alternating direction method (Yang and Zhang, 2011). However, when $\mathbf{X}$ is fixed, the problem is a least square problem with quadratic constraints. There are different approaches to solve this problem such as the projected gradient solvers (Lin, 2007). However, such techniques access the whole dataset in each iteration and consequently cannot process large data in an online fashion. To overcome this problem, we adapt an advanced version of the projected gradient approach that has recently been proposed by (Wang, Li, and Knig, 2011). It is an effective online approach that processes each input data (or a small subset of data) only once. This is particularly important in the context of social media where the input data can potentially be large at each time.

If $\mathbf{D}$ is fixed, then Equation 5.8 will be converted to the following problem (for simplicity in notation and exposition, we assume $\mathbf{D} = \mathbf{D}^{ev}$, $\mathbf{S} = \mathbf{S}^{ev}$, and $\mathbf{X} = \mathbf{X}^{ev}$):

$$\begin{aligned} &\mathbf{X} = \arg\min_{\mathbf{X}} \parallel \mathbf{S} - \mathbf{D}^{t-1}\mathbf{X} \parallel_F^2 + \lambda \parallel \mathbf{X} \parallel_1 \\ &s.t.\text{:}\ \mathbf{X} \geq 0. \end{aligned} \tag{5.12}$$

The above Equation finds the optimal value of $\mathbf{X}$ and can be solved by least angle regression (LARS) method. We note that as x_is are independent, they can be optimized in parallel. However, if $\mathbf{X}$ is fixed, i.e. obtained from the above Equation, then Equation 5.8 will be converted to the following problem:

$$\begin{aligned} &\mathbf{D} = \arg\min_{\mathbf{D}} \parallel \mathbf{S} - \mathbf{D}\mathbf{X} \parallel_F^2 + \lambda \parallel \mathbf{X} \parallel_1 + \mu \parallel \mathbf{D} - \mathbf{D}^{t-1} \parallel_F^2 \\ &s.t.\ \mathbf{D} \geq 0,\ \parallel d_j \parallel_2^2 \leq 1\ \forall j \in \{1...k^{t-1}\} \end{aligned} \tag{5.13}$$

Given $\mathbf{S}$, $\mathbf{X}$, and $\mathbf{D}^{t-1}$, let us define a loss function $\mathcal{L}(\mathbf{D})$ as follows:

$$\mathcal{L}(\mathbf{D}) = \parallel \mathbf{S} - \mathbf{D}\mathbf{X} \parallel_F^2 + \lambda \parallel \mathbf{X} \parallel_1 + \mu \parallel \mathbf{D} - \mathbf{D}^{t-1} \parallel_F^2 \tag{5.14}$$

The projected gradient approach (Lin, 2007) solves Equation 5.13 by iteratively obtaining the projected gradients using the following updating rule:

$$\mathbf{D}_{i+1} = P\Big[\mathbf{D}_i - \alpha_i \nabla_{\mathbf{D}} \mathcal{L}\big(\mathbf{D}\big)_{[\mathbf{D}_i, \mathbf{X}]}\Big] \tag{5.15}$$

where $\mathbf{D}_i$ indicates $\mathbf{D}$ at *iteration* i, the parameter α_i is the step size, and $\nabla_{\mathbf{D}} \mathcal{L}(\mathbf{D})_{[\mathbf{D}_i, \mathbf{X}]}$ is the gradient of $\mathcal{L}(\mathbf{D})$ with respect to $\mathbf{D}$, see Equation 5.16, evaluated on $\mathbf{D}_i$ and $\mathbf{X}$, and $P[.]$ is a projection function defined for the non-negativity constraint, Equation 5.17:

$$\nabla_{\mathbf{D}} \mathcal{L}(\mathbf{D}) = 2\mathbf{S}\mathbf{X}^T + \mathbf{D}\mathbf{X}\mathbf{X}^T + 2\mu(\mathbf{D} - \mathbf{D}^{t-1}) \tag{5.16}$$

$$P[z] = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.17}$$

The disadvantage of the above approach is that it is slow and needs the parameter α to be carefully chosen to obtain good results. To resolve these issues, Wang et al. (2011) proposed to use the second order information, the Hessian matrix, to make the updating rule in Equation 5.15

Algorithm 5.2. Computing $\mathbf{D}^t$ and $\mathbf{X}^t$ at time t, see TL in Figure 4

Input: $\mathbf{S}^t$, $\mathbf{D}^{t-1}$, itr: number of iterations

Output: $\mathbf{D}^t$, $\mathbf{X}^t$

```
Compute $\mathbf{X}^t$ using $\mathbf{S}^t$ and $\mathbf{D}^{t-1}$ → Equation (5.12)
$\mathbf{D}^t_0$=$\mathbf{D}^{t-1}$
for i=1 : itr do
        compute $\nabla_{\mathbf{D}}\mathcal{L}(\mathbf{D^t_{i-1}})$ → Equation (5.16)
        $\mathbf{U} = \nabla_{\mathbf{D}}\mathcal{L}(\mathbf{D^t_{i-1}})diag^{-1}\big(\mathcal{H}[\mathcal{L}(\mathbf{D})]_{[\mathbf{X^t}]}\big) + \mathbf{D^t_{i-1}}$ → Equation (5.18)
        $\mathbf{D^t_i} = max(\mathbf{0}, \mathbf{U})$
end for
```

parameter free with faster convergence. Following the same approach, we utilize the Hessian matrix to obtain the final updating rule as follows:

$$\mathbf{D}_{i+1} = P\Big[\mathbf{D}_i - \nabla_{\mathbf{D}}\mathcal{L}\big(\mathbf{D}\big)_{[\mathbf{D}_i,\mathbf{X}]}\mathcal{H}^{-1}\big[\mathcal{L}(\mathbf{D})\big]_{[\mathbf{X}]}\Big] \tag{5.18}$$

where Hessian matrix of $\mathcal{L}(\mathbf{D})$ is defined as follows:

$$\mathcal{H}[\mathcal{L}(\mathbf{D})] = \mathbf{X}\mathbf{X}^T + 2\mu\mathbf{I}_k \tag{5.19}$$

and $\mathcal{H}^{-1}\big[\mathcal{L}(\mathbf{D})\big]_{[\mathbf{X}]}$ is the inverse of the Hessian matrix evaluated on $\mathbf{X}$. Since the exact calculation of the inverse of the Hessian matrix is time-consuming for large number topics, we approximate the Hessian matrix by its diagonal line based on the diagonal approximation method as suggested by (Wang, Li, and Knig, 2011). Algorithm 5.2 summarizes the detail procedure of computing $\mathbf{D}^t$ and $\mathbf{X}^t$ given $\mathbf{S}^t$ and $\mathbf{D}^{t-1}$.

In can be shown that the time and space complexity of the proposed algorithm is O($n * itr$) and O($n * m$) where n is the number of input data, m is the vocabulary size, and itr is the number of iterations in Algorithm 5.2. For more information, please see (Mairal et al., 2010; Wang, Li, and Knig, 2011; Mairal et al., 2009).

5.3 Evaluation Methodology

The purpose of our evaluation is to assess how the proposed approach makes a real-time judgment on the relevant keywords, micro-posts, and topics about a given organizations. We evaluate our approach from two perspectives: (a) the performance of our approach in identifying the relevant data, and (b) modeling the topics about the organization as live data streams in through time.

5.3.1 Evaluation Metrics for Classification

We evaluate the performance of our classifier based on the traditional IR evaluation metrics, namely *Precision*, *Recall* and *F1-score* metrics (Manning, Raghavan, and Schtze, 2008). In particular, we employ the the following measures to evaluate the performance of our classifier for the positive (relevant) and negative (irrelevant) classes:

$$\begin{aligned} Precision^{+} &= \frac{N_{correct^{+}}}{N_{labeled^{+}}} \\ Recall^{+} &= \frac{N_{correct^{+}}}{N_{total^{+}}} \\ F1^{+} &= \frac{2(Precision^{+})(Recal^{+})}{(Precision^{+})+(Recal^{+})} \end{aligned} \tag{5.20}$$

$$\begin{aligned} Precision^{-} &= \frac{N_{correct^{-}}}{N_{labeled^{-}}} \\ Recall^{-} &= \frac{N_{correct^{-}}}{N_{total^{-}}} \\ F1^{-} &= \frac{2(Precision^{-})(Recal^{-})}{(Precision^{-})+(Recal^{-})} \end{aligned} \tag{5.21}$$

$$Avg - F1 = \frac{(F1^{+})+(F1^{-})}{2} \tag{5.22}$$

where $N_{correct^+}$ is the number of micro-posts that were assigned correct relevant label, $N_{labeled^+}$ is the number of micro-posts that were labeled as relevant, and N_{total^+} is the total number of relevant micro-posts (the same definition applies for the irrelevant class). $F1^+$ and $F1^-$ are the classification performances for the relevant and irrelevant classes respectively and therefore $Avg - F1$ indicates the average classification performance in terms of F1-score.

5.3.2 Evaluation Metrics for Topic Learning

We consider two evaluation metrics to assess the performance of our *topic miner* component, namely *topic detection accuracy*, and *miss-rate at first detection*. The first measure evaluates the topic detection performance in terms of precision and recall, whereas the second measure evaluates the amount of information (number of tweets) that has been *missed* before the first automatic detection of each topic. The second measure is important as we need a small miss-rate for earlier prediction of emerging topics. Here, we formally define these two evaluation measures.

Assume that the set $\mathcal{I} = \{I_i, I_2, ..., I_n\}$ is our topic ground-truth where each I_j represents a topic and $\phi(I_j)$ indicates the set of tweets that are related to the topic I_j. Furthermore, let o_{I_j} and l_{I_j} be the time that the first and last tweet of I_j were posted respectively (we call these two times the *origin* and the *last* time for I_j respectively, thus, $[o_{I_j}, l_{I_j}]$ shows the life time of I_j). Let $d_i \in \mathbf{D}^t$ be a topic that was detected at time t such that $o_{I_j} \leq t \leq l_{I_j}$. We define the closeness between d_i and I_j as follows:

$$\begin{aligned}Precision^{ij} &= \frac{|\phi(d_i) \cap \phi(I_j)|}{|d_i|} \\ Recall^{ij} &= \frac{|\phi(d_i) \cap \phi(I_j)|}{|I_j|} \\ F1^{ij} &= \frac{2Precision^{ij}Recal^{ij}}{Precision^{ij} + Recal^{ij}}\end{aligned} \tag{5.23}$$

where $|.|$ indicates the cardinality of the corresponding set (number of tweets). The value of $F1^{ij}$ shows the similarity between the two topics, i.e. $F1^{ij} = 1$ iff the two topics contain exactly the same set of tweets, and $F1^{ij} = 0$ iff they are disjoint. Topic $d_i \in \mathbf{D}^t$ that produces the maximum value of $F1^{ij}$ for I_j is considered as the most similar topic to I_j (i.e the *best* match).

The overall performance of topic detection for the topic set $\mathcal{I}$ can then be determined as follows:

$$F1 = \frac{\sum\limits_{\forall j=1, i=\arg\max_k F1^{kj}}^{n} F1^{ij}}{n} \tag{5.24}$$

As for the second evaluation measure, *miss-rate at first detection*, the fraction of $\phi(I_j)$ tweets posted before the *origin* time of d_i (that is the best match of I_j) are considered as the missed tweets and their percentage determines the value of miss rate (MR) for I_j. Formally, the miss rate for I_j is determined with respect to $d_i \in \mathbf{D}^t$ and is defined as follows:

$$MR^j = \frac{|s : s \in \phi(I_j)\ \&\ timestamp(s) < b_i|}{|I_j|} \tag{5.25}$$

where b_i is the origin time of d_i. The overall miss-rate for the topic set $\mathcal{I}$ is then obtained as follows:

$$MR = \frac{\sum_{j=1}^{n} MR^j}{n} \tag{5.26}$$

A good topic miner should have a high topic detection performance, $F1$, and a small miss rate, MR.

5.4 Experiments

We first explain the data and settings we used in this Chapter and then present the results of our experiments.

5.4.1 Data and Settings

We considered three organizations in this study. The three organizations are namely *National University of Singapore*[3] (NUS), *Development Bank of Singapore*[4] (DBS), and *StarHub* company[5] (StarHub). NUS is an ambiguous organization as it shares its acronym with *National Union of Students* in UK [6] and Australia[7] and NU Skin company[8] etc. Similarly, DBS is ambiguous as it shares its acronym with many organizations like *Dublin Business School*[9],*Doha British School*[10], and concepts like *Defensive Backs* etc, while the third organization (StarHub) is not ambiguous.

Our crawlers utilized the streaming API of twitter to crawl the corresponding data. We manually identified around 10 fixed keywords for each of the ambiguous organizations NUS and DBS (including their acronyms)

[3] http://www.nus.edu.sg/

[4] http://www.dbs.com.sg/

[5] http://www.starhub.com/

[6] http://www.nus.org.uk/

[7] http://www.unistudent.com.au/site/

[8] http://www.nuskin.com/

[9] http://www.dbs.ie/

[10] www.dohabritishschool.com/

Table 5.1: Statistics and crawling period for three organizations, NUS, DBS, and StarHub

Org	**fixed kw**	**known acc**	**key-user**
NUS 1/1/2012-12/30/2012	142K	10K	2.3M
DBS 6/1/2012-12/30/2012	6.6K	5.5K	0.5M
StarHub 6/1/2012-12/30/2012	9.7K	5.9K	2.3M

and only one fixed keyword, the term “starhub” itself, for StarHub. Furthermore, we manually identified the known accounts for each organization (around 5 to 30 accounts for each organization). Table 5.1 shows the number of tweets obtained from each of the crawlers and the crawling period for the three organizations. It is clear that the key-users generate a large portion of these data.

5.4.1.1 Ground Truth and Settings for Classification

We created a ground-truth of tweets as relevant or irrelevant for each of the three organizations. For this purpose, we considered all the tweets crawled in a time-window of 10 continuous days for each organization and employed a semi-automatic approach to label them as relevant or irrelevant to the target organization.

To ease the annotation task, we first extracted all the hashtags from the tweet set of each organization. We manually labeled these hashtags as relevant or irrelevant to the target organizations[11]. We then constructed a set of labeled tweets using: (a) all the tweets that contained at least one of the labeled hashtags and (b) all the tweets posted by known accounts of the organizations that contained at least one fixed keyword. We learned

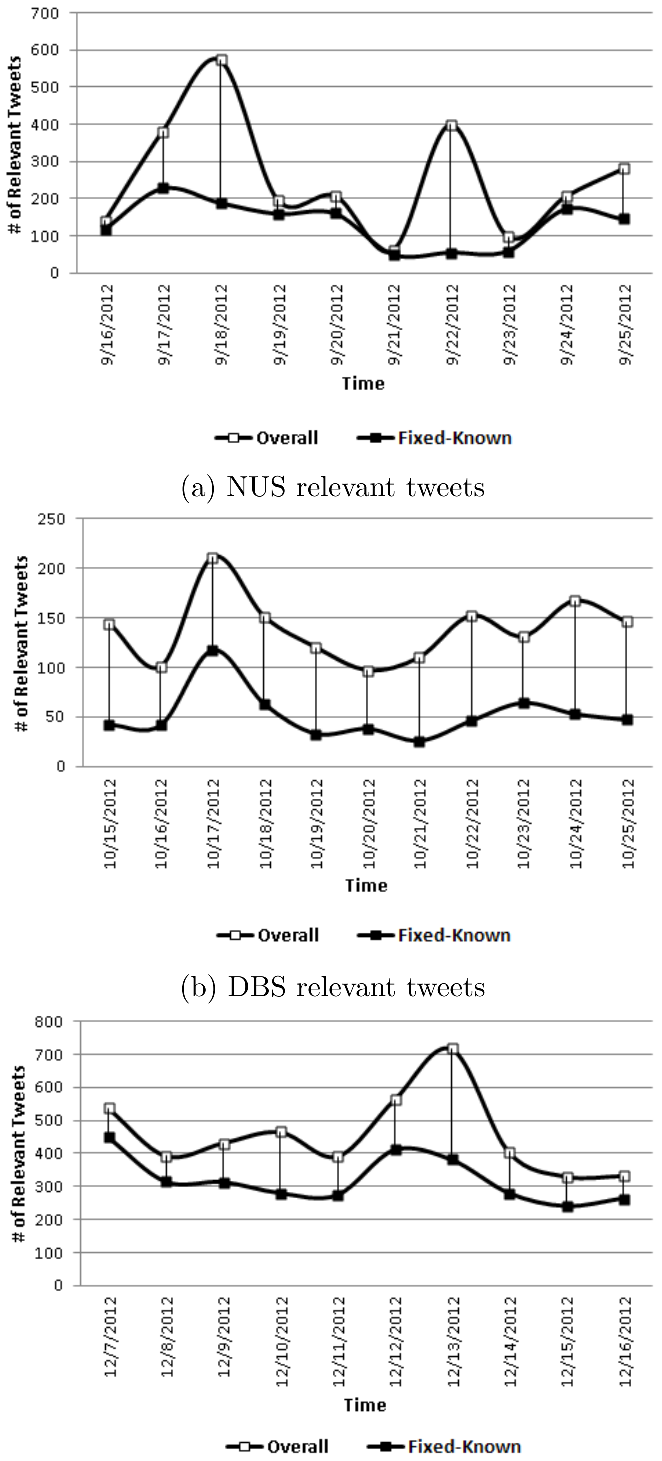
[11] We ignored all the general hashtags like “#news”, “#travel”, “#job”, etc

an SVM classifier (Hall et al., 2009) using this training set and utilized it to label the rest of the tweets crawled in the time-window of 10 continuous days. We utilized *term Unigrams* and *user profile information* such as *user's location* and *timezone* as classification features. In case of low confidence in the classification results, we judged the tweets based on manual annotation. Overall, we obtained 2.5K, 1.5K, and 4.5K relevant tweets for NUS, DBS, and StarHub respectively[12].

Figures 5.2 shows the distribution of the relevant tweets in the resultant ground-truth for the three organizations. "Fixed-Known" indicates the number of relevant tweets obtained by the *fixed keyword* or *known account* crawlers for the organization, while "overall" indicates the total number of relevant tweets obtained by all the three crawlers. As it is clear, there are many relevant tweets crawled by the *key-user* crawler. Such tweets can greatly improve the performance of online topic miner algorithms by providing more content information about the topics. We should also note that there is a high overlap between the data obtained by the *fixed keyword* and *known account* crawlers. This is to be expected as the tweets posted by the *known accounts* of organizations are mainly official news about the organization and usually contain the fixed keywords.

For parameter setting, we use the first three days of the ground-truth as development data to learn the parameters T and α. We then employed the resultant values to evaluate the classification performance on the other seven days.

[12] We only considered English tweets and ignored all the tweets with less than three terms because such tweets are usually context-less with no useful information.



(a) NUS relevant tweets

(b) DBS relevant tweets

(c) StarHub relevant tweets

Figure 5.2: The distribution of relevant tweets for three organizations namely NUS, DBS, and StarHub.

5.4.1.2 Ground Truth and Settings for Topic Modeling

Similar to the above approach, we conducted a semi-automatic method to construct our topic dataset. For this purpose, we manually identified 45 topics for the three organizations (15 for each organization). For each topic, we identified the *hashtags* and all the keywords and key-phrases that uniquely identify the topic. Then, for each topic, we found the tweets that are posted within the topic life time and contain at least one topical keyword or key-phrase. We treat these tweets as the relevant tweets to that topic. Table 5.2 shows a sample of such topics. Our topic dataset covers different events about the organizations and range from small topics of around 50 tweets per topic to topics with more than 1000 tweets.

For parameter setting, we used part of the topic dataset (5 first topics of each organization) as development data to tune the learning parameter μ for each organization. We then utilized the resultant μ values to perform the evaluation on the rest of the topics for the target organizations. We also study the effect of this parameter on the performance of our approach. In addition, for parameter setting, we set $\lambda = \frac{1.2}{\sqrt{m}}$, a classical normalization factor (Bickel, Alexandre, and Tsybakov,), in all the experiments where m is the number of terms. We also empirically set the threshold for residual error to $\eta = 0.3$.

5.4.2 Experimental Results

In this section, we report detail experiments on the performance of our classifier and topic miner components.

Table 5.2: Some sample topics/events of organizations along with their occurrence information

Organization	Topic/Event	Period
NUS	fire in engineering	08/10/12 - 08/30/12
	nus open house	03/09/12 - 03/24/12
	flagday	08/05/12 - 08/16/12
DBS	sudden jump in dbs profit	11/01/12 - 11/04/12
	dbs grant for social enterprises	10/29/12 - 10/31/12
	paypass facility	11/01/12 - 11/06/12
StarHub	poor outdoor coverage fine	12/06/12 - 12/10/12
	leeteuk sistar on starhub	12/16/12 - 12/18/12
	lunch of central comedy asia	10/31/12 - 11/08/12

5.4.2.1 Classification Performance

For the classification experiments, we employed the SVM implementation of the Weka toolkit as our content-based classifier. To simulate live data streaming, we ran our online model over one month data (the month that includes the ground truth data) for each organization, while we restricted the evaluation to the tweets in our ground truth dataset.

As mentioned before, we used the first three days of the ground-truth to learn T and α and employed the obtained values to evaluate the classification performance on the other seven days. Table 5.3 the classification performance measured by the average F1 score discussed in Section 5.6.1. The *fixed-kw*, *Unigram*, and *Bigram* rows show the classification performance when we used fixed keywords, term Unigrms, and term Bigrams as classification features respectively (we consider them as baselines). *Dynamic-kw* show the classification performance when we used dynamic keywords as classification features, i.e. the results obtained from Equation (6), while *Dynamic-kw + User* represents the performance when we used dynamic keywords in conjunction with user information, Equation (5). Note that, in all the settings, if an input tweet did not contain any classification fea-

Table 5.3: Classification performance in terms of $Avg - F1$ with different types of features and input data

	NUS	**DBS**	**StarHub**	**Average**
Fixed-kw	47.82, T:3	41.67, T:4	49.63, T:4	46.37
Unigram	63.24, T:6	84.76, T:3	88.92, T:7	78.97
Bigram	62.30, T:6	84.89, T:7	88.92, T:7	78.70
Dynamic-kw	65.15*, T:5	85.29*, T:4	88.94*, T:3	79.79
Dynamic-kw + User	**81.08***, T:5, a:0.3	**89.64***, T:3, a:0.4	**89.82***, T:2	**86.85**

ture, we treated it as irrelevant. In addition, in all the experiments the *two-tailed paired* t-test with $p < 0.01$ was used for significance testing. We use the asterisk mark (*) to indicate significant improvement over the best performing baseline.

We list the insights we obtained from the results as follows:

- In all the settings, using only fixed keywords leads to poor classification performance: this was expected as fixed keywords can only capture part of the relevant data and result in very low recall.

- The Unigram model, though simple, greatly improves the classification performance as compared to the fixed keywords. This is because it utilizes more context information. We note that the improvement for DBS and StarHub is greater. This could be related to the very specific domain of these organizations that helps the Unigram model to easily prune the noise from the test data.

- Bigram model does not improve the classification performance over the Unigram model for NUS, but slightly improves the performance for DBS and StarHub. We also observed that the Bigram model is not effective for tweets with fixed keywords: this is because the fixed keywords alone are readily good classification features, while the Bigram model reduces the weight or importance of these features by combining them with other keywords to form Bigrams. Note that, for the Bigram model, the value of T (the training interval) is higher than other models. In other words, the Bigram model needs to incorporate more past data to produce good results. This may not be desired as it forces higher processing time.

- The dynamic keywords alone significantly improve the classification performance over the best performing baseline. This indicates that our keyword mining algorithm is able to effectively discriminate current relevant keywords from irrelevant ones for each organization. We should also note that since the dynamic keyword model has fewer number of features (as compared to the total number of terms or Unigrams), the classification is performed very fast which is desirable in online settings.

- Adding user information significantly improves the classification performance. This is because we utilize the entire user activity with respect to each organization, see Equation 5.5, to judge its input data.

- The value of α is smaller than 0.5 for both NUS and DBS: this was expected because the parameter α only affects tweets with fixed keywords (see Algorithm 5.1) and for such tweets the weight of the user score, i.e. $1 - \alpha$, is expected to be high.

- The classification performance is invariant to the parameter α in case of StarHub: as we mentioned above, the parameter α only affects tweets with fixed keywords. Such tweets are considered as relevant for non-ambiguous organizations by default (see Figure 5.3(c)).

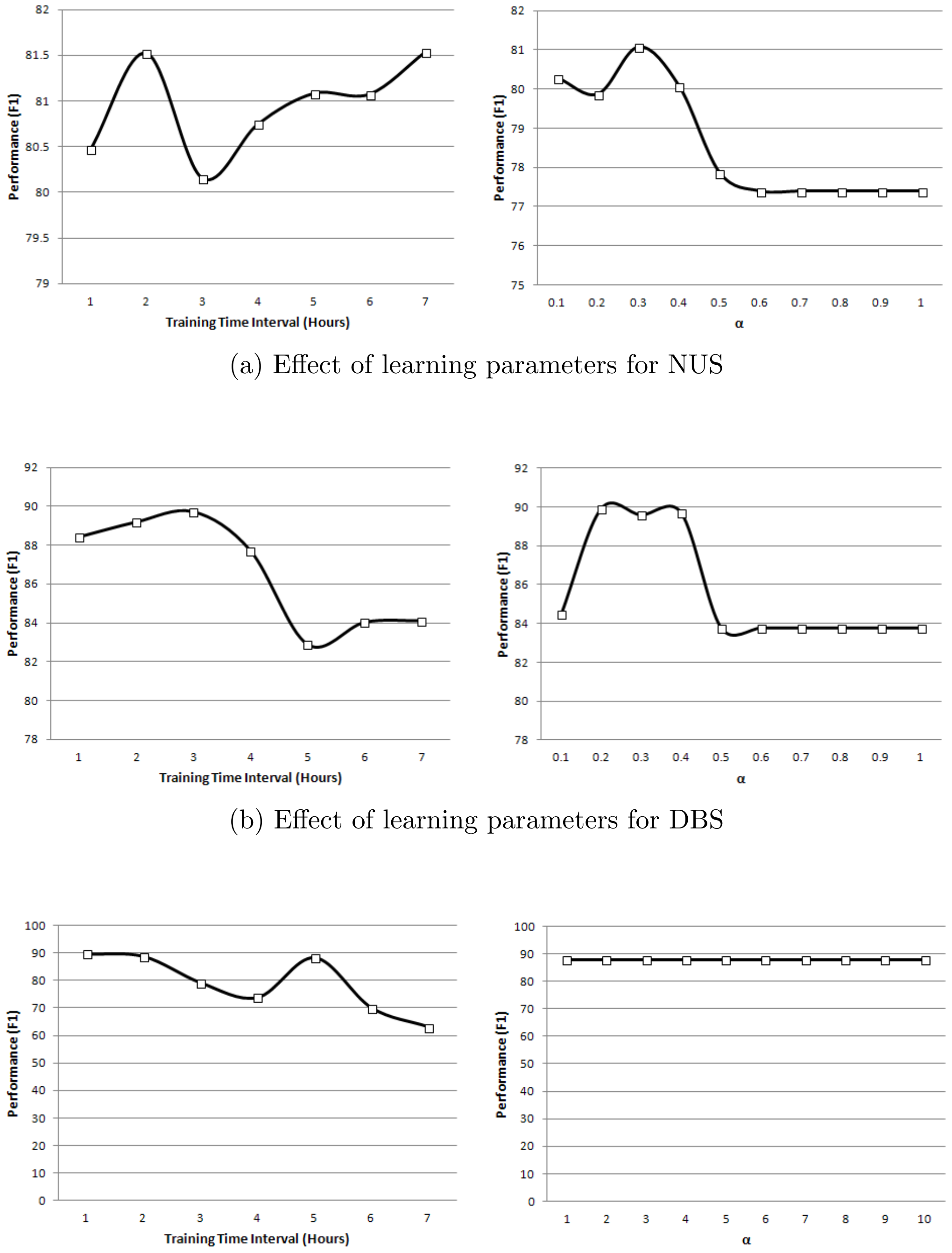
Figures 5.3 shows the effect of the learning parameters T and α on our model, *Dynamic-kw + User*, evaluated over the entire ground truth dataset. In each case, we fixed one of the parameters and investigated the effect of the other one. For the fixed parameter, we used the value obtained from the development set (Table 5.3). We restricted the interval time to 7 hours, i.e. $1 \leq T \leq 7$, and the parameter α to $0.1 \leq \alpha \leq 1$ with learning steps set to 1 hour and 0.1 for T and α respectively.

Here are some insights we learned from this Figures:

- As the Figure shows, greater time intervals (T) *slightly* increases the classification performance for NUS but causes great reduction in the classification performance for DBS and StarHub: We believe the lifetime of the topics happening about the organization can affect the classification performance. If the topics have a long lifetime, increasing the time interval T may not reduce the performance as the old topics are still active, whereas for topics with short lifetime, increasing T dramatically reduces the performance as the old discussions are not active anymore and thus the dynamic keywords extracted from such topics are not useful features to classify the current input data.

- As Figure 5.3 (a) and (b) show, for NUS and DBS as ambiguous organizations, smaller values of α (i.e. giving less weight to the content relevance score and higher weight to the user score for the tweets with fixed keywords) leads to better performance. This result indicates the important role of user scores to classify tweets with fixed keywords.

- As mentioned above, the classification performance for non-ambiguous organizations like StarHub is invariant to the parameter α but will be affected by the learning time interval.



(a) Effect of learning parameters for NUS

(b) Effect of learning parameters for DBS

(c) Effect of learning parameters for StarHub

Figure 5.3: Effect of the learning parameters T and α of the classification performance for the three organizations.

5.4.2.2 Topic Modeling Performance

We evaluate the performance of the topic miner component in this Section. To simulate live streaming data we apply our online topic modeling algorithm over the entire dataset for each organization and only restrict the evaluation to the topic dataset.

We first tune the learning parameter μ for each organization using our development dataset [13]. We then employ the resultant μ to evaluate the topic modeling performances of different approaches on the test topics. In these experiments, we consider the basic Non-Negative Matrix Factorization (NMF) algorithm without the *temporal continuity* and *sparse matching* constraints as the baseline (i.e. for the baseline, we set $\lambda = 0$ and $\mu = 0$ in our optimization framework to obtain the baseline performance).

Table 5.4 shows the evaluation results for *topic detection* in terms of F1 performance measured by Equation (22). The *Overall* column shows the performance when we perform the evaluation over all the relevant input data for the topic modeling purpose, while the *Known* column shows the corresponding performance when we only use the relevant tweets obtained from *fixed keyword* or *known account* crawlers.

Here, we list the insights we obtained from these results:

- In almost all the case (except DBS baseline), the *overall* data results in a higher performance as compared to known data: this improvement is because of our intelligent data harvesting approach. In other words, the results show that there are many relevant tweets that are not covered by the fixed keywords and known accounts of organizations. The average performance of baseline and our optimization

[13] as mentioned before, we set $\lambda = \frac{1.2}{\sqrt{m}}$ and $\eta = 0.3$.

Table 5.4: Topic detection F1 performance with known and overall input, the higher values show better performances

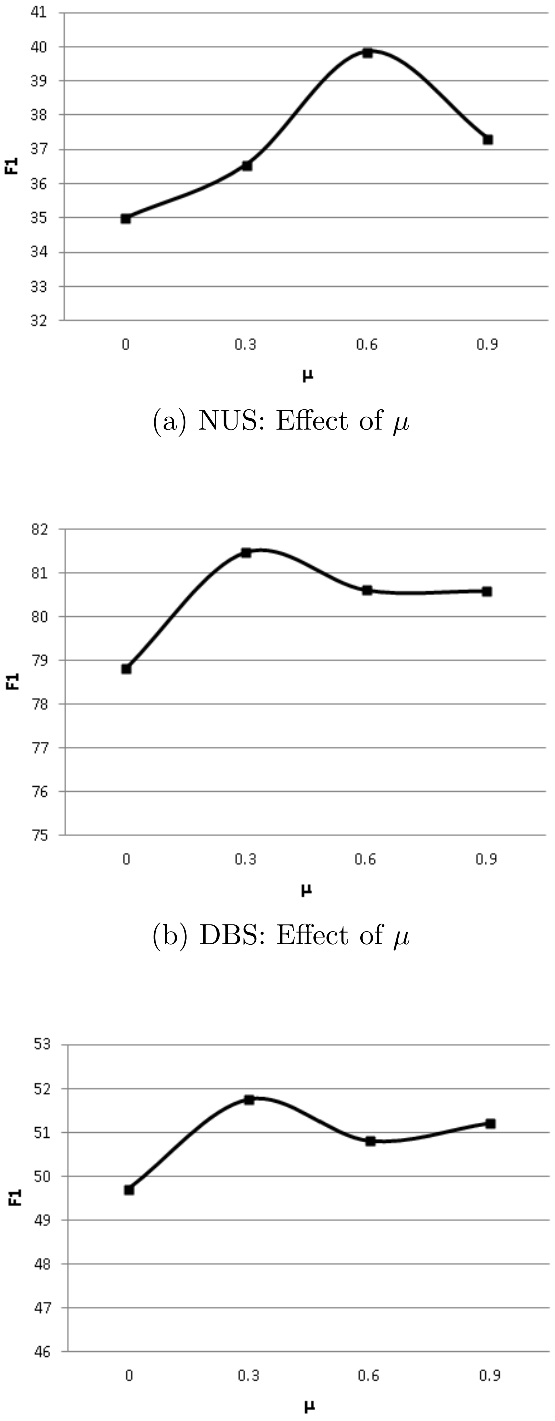
	Baseline (NMF)		**Optimization Framework**	
Organization	**fixed-known**	**overall**	**fixed-known**	**overall**
NUS	39.30	**40.03**	37.56, $\mu: 0.3$	39.82, μ:0.3
DBS	64.67	61.42	65.63, $\mu: 0.3$	**80.64**, μ:0.4
StarHub	42.88	46.84	43.07, $\mu: 0.3$	**51.78**, μ:0.3
Average	48.95	49.43	48.75	**57.41**

framework increases from 48.95% to 49.43% and 48.75% to 57.41% respectively by utilizing these relevant tweets.

- Our optimization framework outperforms the baseline for DBS and StarHub while its performance for NUS is comparable with the baseline. The average improvement over the baseline is 7.98%, i.e. from 49.43% to 57.41%, when we utilize the overall input data for topic modeling.

- We note that the lower F1 performance for NUS and SatrHub (as compared to DBS) could be related to the longer lifetime of NUS' and StarHub's topics than DBS's topics in our dataset. The long topics may reduce the topic modeling performance because such topics may be divided into sub-topics by different algorithms (mainly because of shifts in topics through time). This is while we only have one best match for each topic.

Comparing the average performances of the baseline and our optimization framework, we conclude that topic detection and tracking is more effective if we use the sparsity and temporal continuity constraints for topic mining. In fact, the temporal continuity constraint helps the system to utilize the past information about topics to make a better judgment about the current topics. Figure 5.4 shows the effect of this constraint on the over-



(a) NUS: Effect of μ

(b) DBS: Effect of μ

(c) StarHub: Effect of μ

Figure 5.4: Effect of the temporal continuity constraint on the performance of topic modeling for three organizations. We perform the experiments with $\mu = 0, 0.3, 0.6, 0.9$.

Table 5.5: Miss rate results for fixed keywords and overall input, the lower values show better performances

	Baseline (NMF)		**Optimization Framework**	
Organization	**fixed-known**	**overall**	**fixed-known**	**overall**
NUS	36.06	37.83	28.17, $\mu = 0.3$	**26.78**, $\mu = 0.3$
DBS	15.88	**15.87**	15.88, $\mu = 0.3$	16.96, $\mu = 0.4$
StarHub	40.27	27.62	40.27, $\mu = 0.3$	**23.11**, $\mu = 0.3$
Average	30.74	27.11	28.11, $\mu = 0.3$	**22.28**

all topic modeling performance. The high performance of topic modeling when $\mu = 0$ shows the effect of the sparsity constraint controlled by λ.

Table 5.5 shows the evaluation results for the *miss-rate at first detection* metric measured by Equation 5.26. The lower values of miss-rate indicate that the topic modeling algorithm is able to identify the emerging topics earlier. As Table 5.5 shows, the average miss-rate is lower when we use the overall data instead of only tweets obtained by the *fixed keyword* or *known account* crawlers. This suggests that we can detect emerging topic earlier, if we make use of more (relevant) tweets. We thus conclude that our *key-user* crawler is an effective resource for early prediction of emerging topics about organizations. The results show that our approach outperforms the baseline by 4.83% reduction in the average miss-rate (from 27.11% to 22.28%).

5.5 Summary

In this Chapter we proposed a principled online approach to harvesting a more representative distribution of relevant contents about organizations by mining their current keywords and key-users. We showed that content and user information can be utilized effectively to identify relevant data for organizations. The results show that the key-users of organizations are useful resources to elicit more relevant data about organizations from social media which in turn leads to a more accurate topic modeling for the organization as well as earlier detection of its emerging topics. Furthermore, we found that users and content information in conjunction are the key factors in judging the relevance of micro-posts to organizations specially for ambiguous ones.

We also proposed an effective online non-negative matrix factorization approach for mining organization topics through time. We found that the performance of topic detection is higher when the topics are allowed to evolve under the *temporal continuity* constraint that restricts dramatic changes in the topic sets of two consecutive time points. We also show that the *sparsity* constraint that restrict tweets to match with only a few topics is another effective constraint in modeling the evolution of topics through time.

Chapter 6

Mining User Communities for Organizations

It is critical for *user-centric* organizations and businesses to identify their user community and influential users from social media to acquire actionable insights from the relevant content that they produce. In this Chapter, we focus on the task of community detection for organizations. User community detection is a challenging task because of the *polysemy* problem in the social media context. To tackle this issue, we utilize topical information to strengthen community signals. In particular, we formulate the community detection task as a semi-supervised learning task in the graph context and introduce two different relations, namely: *social* and *topical* relations to discover user communities for organizations. We experimentally show the effectiveness of the proposed approaches for several organizations on streaming data obtained from Twitter.

6.1 Introduction

As discussed before, users may *follow* the known accounts of organizations to get up-to-date news etc. As such, the social relations among users (obtained from the *user graph*) are helpful clues to detect user community of the organizations. However, not all the users of an organization follow its official accounts. Therefore, we propose to utilize topical relations among users to strengthen the community signals (e.g. see Figures 6.1 and 6.2) for discriminating user communities of ambiguous organizations.

The rest of this Chapter is organized as follows: Section 6.2 presents an effective optimization framework designed for community detection for ambiguous organizations. Section 6.3 reports the experimental settings and results, and, finally, Section 6.4 summarizes the Chapter.

6.2 Community Detection for Organizations

We propose a graph propagation algorithm to mine user communities for ambiguous organizations. Let G^u be a *user graph* constructed from all the users who posted at least one micro-post that contain a fixed keyword of a given organization. The edges of G^u represent the social relations among these users (e.g. see Figure 6.1). Since such user graphs are barely connected (specially for large organizations), we utilize the topical relations between users to strengthen the community signals (see Figure 6.2). Such topical relations are expected to be effective as they can relate users of an organization to each other even though they are not socially connected. This results in a more dense graph that leads to more accurate community detection. We refer to the resultant graph as *context graph*, $G^c = (\mathcal{V}, \mathcal{E})$.

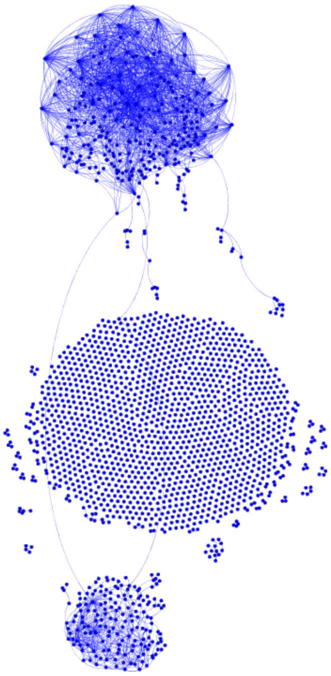


Figure 6.1: User Graph of NUS dataset. The blue nodes represent users and the edges represents *follower / friend* relationships.

In G^c, the nodes represent the users and the topics that they commented about, and the edges represent either the social relations among users or topical relations between users and topics.

Given G^c, we consider the known-accounts of the target organization as positive *seeds*, and the known-accounts of the other organizations that share the same acronym with the target organization as negative seeds. We discriminate the nodes of G^c as relevant or irrelevant to the target organization using its connectivity information. All the user nodes labeled as relevant form the user community of the target organization and all the topic nodes labeled as relevant represent the relevant topics of the target organization.

We attach a *dongle node* or *d-node* (Zhu, Ghahramani, and Lafferty, 2003) to each node of G^c (these nodes are not shown in Figure 6.2 for better clarity of the graph). The purpose of d-nodes is to accumulate the label

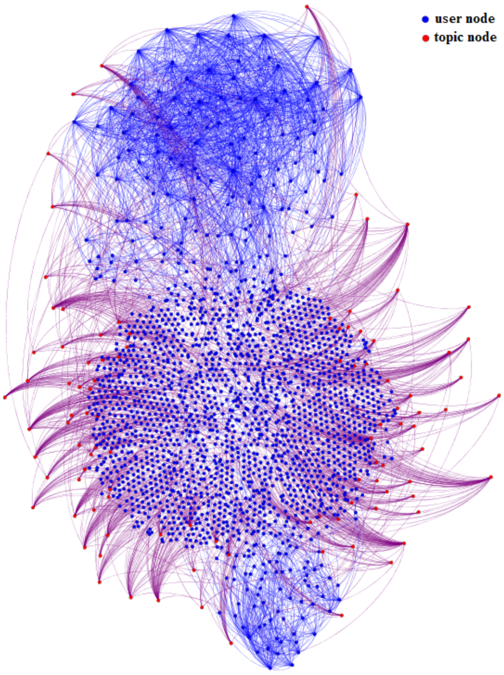


Figure 6.2: The corresponding Context Graph of NUS. The red nodes represent topics. The edges between users and topics represent *topical* similarities.

scores for their corresponding nodes through time so that we can utilize the past information for label propagation as new data streams in. Initially, the d-nodes are set to +1 for relevant nodes (e.g. the known-account of the target organization), -1 for irrelevant nodes (known-account of other organizations), and 0 for all the other nodes. See Section 6.2.1.1 for more information about d-nodes.

6.2.1 Label Propagation

The label propagation problem can be described as follows: Assume that there exist n nodes $\mathcal{X} = \{x_1, ..., x_n\}$ in G^c. Let the first l nodes $\mathcal{X}^l = \{x_1, ..., x_l\}$ be the labeled data (nodes with non-zero d-node value) and the remaining nodes $\mathcal{X}^u = \{x_{l+1}, ..., x_n\}$ be the unlabeled nodes. Let y_i indicates the label of x_i, i.e. $y_i = +1$ for relevant nodes and $y_i = -1$ for irrelevant nodes. The aim is to find a real-valued function $f : x \rightarrow \mathbb{R}$ that gives a score $f(x)$ to each node x. The value of function f on the labeled data x_i is the same as its initial label y_i, i.e. $f(x_i) = y_i$ for $i = \{1, ..., l\}$. The problem is then to predict the scores for the unlabeled nodes, i.e. $f(x_j)$, $j = \{l + 1, ..., n\}$.

The above problem can be modeled as a semi-supervised learning task in the graph context where the connectivity information of the graph can be utilized to estimate the scores for the unlabeled nodes. We first construct the context graph, and then solve the resultant optimization problem.

6.2.1.1 Context Graph Construction

Let $G^c = (\mathcal{V}, \mathcal{E})$ be an undirected edge-weighted graph where the node set $\mathcal{V}$ corresponds to the n elements of $\mathcal{X}$, and edges E are weighted by an $n * n$ symmetric weight matrix $\mathbf{W}$. We construct the context graph as follows: for any two users u_i and u_j, if one user follows another, we add an undirected edge with edge weight of 1, i.e. $w_{ij} = 1$, between the two users $(u_i, u_j) \in E$. Similarly, for any user node u_i and topic node u_j, if u_i has sent tweet(s) about the topic u_j, there will an undirected edge between the two nodes $(u_i, u_j) \in E$ (Appendix B explains our approach to compute the edge weights between user and topic nodes). If there is no edge between two nodes, the corresponding weight is set to 0. The above configuration results in a large graph in which each unlabeled nodes is potentially connected to several labeled nodes through different edges/paths (see Figure 6.2).

Furthermore, as discussed above, we assume a d-node containing an initial score for each unlabeled node, i.e. $f(x_i) = \hat{y_i}$ $\forall i = l + 1...n$. Each d-node is connected to its corresponding unlabeled node with the edge weight of 1 and acts as prior knowledge for the semi-supervised learning framework. An example of prior knowledge is the label of a topic either implicitly learned by a textual classifier or explicitly given by an annotator. $\hat{y}_i$ is set to zero when there is no such initial labels.

6.2.2 Optimization Framework

Assuming that the d-nodes are connected to their corresponding unlabeled nodes with the weight of 1, our aim is to obtain a *smooth graph* by minimizing the following *energy* function:

$$\begin{aligned} E(f) = \alpha \sum_{x_i \in \mathcal{X}^l} (f(x_i) - \hat{y}_i)^2 + \\ (1-\alpha) \sum_{x_i \in \mathcal{X}^u} \sum_{x_j \in Adj(x_i)} w_{ij}(f(x_i) - f(x_j))^2 \end{aligned} \tag{6.1}$$

where $Adj(x_i)$ represents the sets of x_i's adjacent nodes, the parameter $\alpha \in [0,1]$ indicates the influence of each source of learning (dongle node vs. adjacent nodes) on the score of x_i. Equation 6.1 represents the requirements that for each unlabeled node $x_i \in X^u$, we want $f(x_i)$ to be consistent with its d-node and its neighbors. The smaller values of α increase the effect of the adjacent nodes, while greater values of α decrease such effects.

The optimization problem can be defined as follows:

$$\hat{f} = \arg\min_f E(f) \tag{6.2}$$

To find a closed-form solution to the above Equation we define an

$n * n$ matrix $\mathbf{T}$ as follows:

$$T_{ij} = \begin{cases} 0, & i \in L, j \in L \\ \alpha w_{ij}, & i \in L, j \in U \\ \alpha w_{ij}, & i \in U, j \in L \\ 2(1-\alpha) w_{ij}, & i \in U, j \in U \end{cases} \tag{6.3}$$

where $L = 1...l$ and $U = l{+}1...n$ are the labeled and unlabeled node indices respectively. Let $\mathbf{D}$ be a diagonal matrix derived from $\mathbf{T}$ as follows:

$$D_{ii} = \sum_{j=1}^{n} T_{ij} \tag{6.4}$$

Let $\mathbf{\Omega} = \mathbf{D} - \mathbf{T}$ be the $n{*}n$ graph Laplacian matrix (Luxburg, 2007), $\mathbf{f} = [f(x_1), ..., f(x_n)]^T$, and $y = [y_1, ..., y_l, \hat{y_{l+1}}, ..., \hat{y_n}]^T$ where $f(x_i) = y_i$ for the labeled nodes ($i = 1...l$), and $\hat{y_j}$ is the value of the dongle nodes for the unlabeled nodes ($j = l + 1...n$). We can then rewrite Equation 6.1 as follows:

$$E(f) = (\mathbf{f} - \mathbf{y})^T \mathbf{I} (\mathbf{f} - \mathbf{y}) + \mathbf{f}^T \Omega \mathbf{f} \tag{6.5}$$

where $\mathbf{I}$ is the $n * n$ identity matrix. The minimum energy function $\hat{\mathbf{f}}$ of the above quadratic function can be obtained as follows:

$$\frac{\partial E(f)}{\partial f} = 0 \Rightarrow \hat{\mathbf{f}} = (\mathbf{I} + \mathbf{\Omega})^{-1} \mathbf{y} \tag{6.6}$$

Because $\mathbf{f}^T \mathbf{\Omega} \mathbf{f} > 0$, $\mathbf{\Omega}$ is a symmetric and positive semi-definite matrix and consequently the above solution is the unique answer to our optimization problem. However, since the exact calculation of the inverse matrix is time-consuming for large graphs, we approximate the inverse matrix by its diagonal line based on the *diagonal approximation* method (Wang, Li, and Knig, 2011).

The user community of the target organization will then be all the user nodes with positive label: $\{x_i \in \mathcal{X} : f(x_i) > 0 \text{ and } x_i \text{ is a user node}\}$. The resultant labels will be stored in d-nodes so that we can use them as training data in the next iteration (when new data streams in).

6.3 Experiments

6.3.1 Data and Setting

As mentioned before, NUS is shared among *National University of Singapore*, *National Union of Students*, and *NU Skin*$^{\text{TM}}$etc. Similarly, DBS is shared among *Development Bank of Singapore*, *Dublin Business School*, and *Doha British School* etc. To obtain data for these organizations, we use the streaming API of twitter with crawling queries formed from the full name of the above organizations and their acronym. Furthermore, we manually identify around 10 known-accounts for each organization. Figure 6.1 and 6.2 show the overall user and context graph for NUS and DBS respectively.

In this Chapter, we consider *National University of Singapore* (NUS-1), *National Union of Students* (NUS-2), *Development Bank of Singapore* (DBS-1), and *Dublin Business School* (DBS-2) as the target organizations. To create a ground-truth of user communities for the target organizations, we considered all the tweets posted in a time-window of 10 continuous days in the crawled datasets. We employed a semi-automatic approach to label the users with respect to the ambiguous organizations. In particular, we used features like *content of the tweets posted by users*, *tweets posted by known-account*, *user profile information* such as *user's location*, *timezone*, and *bio information*, *manually labeled topical keywords*, *manually labeled hashtags* in conjunction with an SVM classifier to determine the relevance of users to organizations. In case of low confidence in the classification results, we judge the user based on manual annotation. We obtained 567 users and around 2.1K tweets for NUS-1, and 1,323 users and 4.9K tweets for NUS-2 during 08/20/2012 to 08/30/2012. Similarly, We obtained 862 users and 2K tweets for DBS-1, and 1,293 users and 3K tweets for DBS-2 during 10/20/2012 to 10/30/2012.

For parameter setting, we learn α in Equation 6.2 from data.

6.3.1.1 Data Analysis

Tables 6.1 and 6.2 show different statistics about the user and context graphs constructed for the target organizations. The first two measures (rows 1 and 2) evaluate the number of components that can be extracted from the graphs using different criteria. *#Communities* shows the number of clusters that can be obtained based on the modularity maximization algorithm, Equation 6.1. As mentioned before, modularity measures the strength of division of the graph into clusters (groups). The *#Components* shows the number of connected components in each graph where a connected component is a sub-graph in which any two nodes can be connected through a path. As Table 6.1 and 6.2 show, the number of clusters and connected components in the users graphs is much higher than that in the context graphs. This indicates that the topical relations are good means to relate those users who are part of the same user community but are not socially connected.

The third and fourth measures in Table 6.1 and 6.2 evaluate the graphs based on the availability of paths between nodes. The statistics show

Table 6.1: User and context graphs of NUS-1 and NUS-2.

	NUS-1		**NUS-2**	
Measures	**user**	**context**	**user**	**context**
#Communities	408	15	951	34
#Components	404	7	942	17
#Shortest-paths	0.06M	1.1M	0.14M	2.6M
Avg Path length	1.40	1.65	3.26	3.85

Table 6.2: User and context graphs of DBS-1 and DBS-2.

	DBS-1		**DBS-2**	
Measures	**user**	**context**	**user**	**context**
#Communities	910	32	607	21
#Components	905	15	604	10
#Shortest-paths	0.66M	2.9M	0.44M	1.9M
Avg Path length	3.17	3.49	2.12	2.33

that the number of shortest paths between nodes greatly increases, once we add the topical relations to the user graphs. Furthermore, we compute the average graph distance between all pairs of nodes where the connected nodes have a graph distance of 1. As the results show, the context graphs have higher average path length than that of the user graphs. Such increase in the number and length of paths leads to more effective label propagation.

6.3.2 Performance of Community Detection

We consider the task of community detection for a given target organization as a classification task where we treat the target organization as the positive class and all the other organizations with the same acronym as the negative class.

We use the first 5 days of the ground truth to learn the parameter $\alpha \in (0, 1]$, see Equation 6.2. For this purpose, we employ a grid search with steps of 0.1. The evaluation results on the next 5 days are shown in Table 6.3. As the results show, the $F1$ performance of community mining over the

Table 6.3: Community detection performance on user and context graphs of target organizations.

		Precision	**Recall**	**F1**	α
context	NUS-1	61.32	70.94	65.78	0.2
	NUS-2	87.14	83.00	85.02	0.2
	DBS-1	64.10	49.26	55.71	0.2
	DBS-2	38.64	22.03	28.06	0.3
	AVG	**62.80**	**56.31**	**58.64**	-
user	NUS-1	80.85	25.54	38.82	0.2
	NUS-2	91.02	41.00	56.53	0.4
	DBS-1	82.20	25.60	39.04	0.3
	DBS-2	85.07	11.00	19.48	0.2
	AVG	**84.79**	**25.79**	**38.47**	-

context graphs is significantly higher than the corresponding performance on the user graphs for all the target organizations. The improvement stems from the topic nodes that connect users of the same community to each other. We note the high precision but very low recall for community detection over the user graphs. This was expected as many users in the user graph are loosely connected and not reachable from any known-accounts (except followers of the known-accounts). However, adding the topic nodes leads to propagation of the community labels to such nodes.

We also study the effect of the learning parameter α on the performance of community detection for the target organizations. Figures 6.3 and 6.4 show the results of F1 performance for NUS-1 and NUS-2, and DBS-1 and DBS-2 respectively. As the Figures show, a small value of α can lead to a good performance over the context graphs. This is because a small α gives higher weight to the adjacent nodes/users and as such leads to greater propagation of labels. Note that, in the community detection process, the d-node values we learn in each iteration are utilized for training in the next iteration. Greater values of alpha give more weights to the d-nodes obtained from previous iterations. However, a very high value

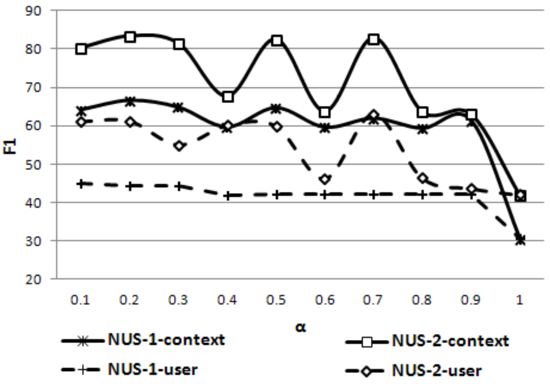


Figure 6.3: Effect of α on the F1 performance of community detection for NUS.

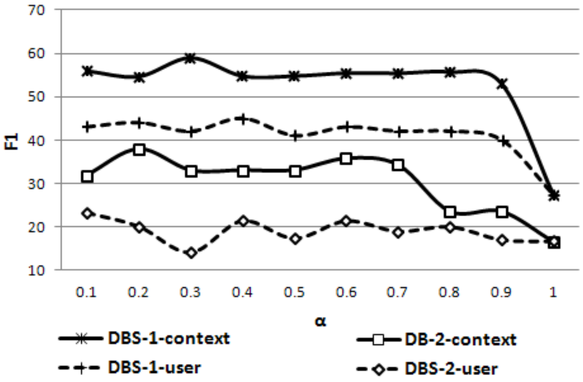


Figure 6.4: Effect of α on the F1 performance of community detection for DBS.

of α prevents label propagation. For example, in the extreme case, when $\alpha = 1$, no label information is propagated and as such all the users are treated as irrelevant to the target organization. This leads to a very poor performance.

6.4 Summary

We proposed efficient algorithms to mine user community of (ambiguous) organizations from social media. We defined the community of an organization as a group of users who post relevant contents about the organization in social media. We showed that topical relations among users can significantly improve the performance of community detection for ambiguous organizations.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we aimed to make sense of the social media contents for user-centric organizations. For this purpose, we first investigated the general problem of opinion word mining in user generated contents. We then studied the problems of intelligent data harvesting for organizations from social media, online learning of the evolving and emerging topics happening about the organizations, and finally mining user communities for organizations. These different aspects of knowledge helps organizations to get actionable insight from social media.

We proposed a general algorithm to sentiment analysis on short text such as micro-posts or online reviews. We introduced a principled approach to constructing sentiment lexicons from user generated contents. In particular, we utilized existing opinion words to extract slang and urban words/phrases from user generated contents. In contrast to previous approaches, our method not only learns the sentiment orientation of words from the existing opinion words but also from other unknown potential opinion words. This approach is more feasible in the web context where the dictionary-based relations (such as synonym, antonym, or hyponym used by previous approaches) between most words are not available. We show that our approach is effective both in terms of the quality of the discovered new opinion words as well as its ability in inferring their sentiment orientation. In addition, by further investigation, we found that *time* is an important factor for sentiment terminology mining. We showed that the time accumulated polarity better reflects the polarity of the opinion words than the polarity obtained at each particular time.

We showed that the common crawling methodology that makes use of a list of known keywords to crawl data cannot obtain a representative distribution of data about organizations from social media. Considering the power law correlation between the number of users and the number of relevant micro-posts for organizations, we proposed to identify and monitor the key-users of the organizations to harvest more relevant data about them. In particular we proposed to elicit data from multiple aspects of information, including (a) *known accounts*, (b) *key users*, and (c) *fixed keywords* of the organization. To address the relevance challenge, we proposed to utilize *context* of the target organization that is defined by the current relevant information (dynamic keywords and micro-posts) and the user community of the organizations. We designed a classifier to predict the relevance of each incoming micro-post to the target organization based on its current context information. We showed that this classifier can discriminate relevant micro-posts from irrelevant ones with a high accuracy. We also show that our data harvesting approach elicit more relevant data about organizations as compared to only *fixed keywords*.

To address the topic discovery and monitoring issue, we proposed to cluster the stream of relevant micro-posts into *emerging* and *evolving* topics through time. The Emerging topics were defined as the new topics that emerge and potentially become major in a short period of time, while the evolving ones were defined as those that have been detected previously and are smoothly evolving through time. For the modeling such behavior, we proposed an *online* sparse coding approach with *temporal continuity* and *sparse matching* constraints. We showed that, this approach better suits streaming data as it processes each input data only once and therefore is linear with respect to the number of micro-posts.

Furthermore, we proposed an effective algorithm to community detection for organizations. We showed that topical relations among users can significantly improve the performance of community detection for organizations.

We found that mining slang and urban opinion words and phrases can significantly improve the performance of sentiment classification on user generated contents, while learning the sentiment orientation of words through time leads to a more accurate polarity inference than learning the sentiment orientation without considering the time factor. Furthermore, we found that the combination of user and content information leads to effective discrimination of relevant micro-posts specially for the ambiguous organizations. However, fixed keywords alone are not effective features for this purpose. We also found that key-users are useful clues to elicit more relevant content about organizations from social media. We showed that monitoring key-users of organizations leads to: (a) higher performance of topic modeling algorithms and (b) earlier detection of emerging topics. Furthermore, we show that the performance of topic modeling further improves when topics are allowed to evolve under the *temporal continuity* and *sparsity* constraint. The temporal continuity constraint ensures no dramatic changes in the topic sets of two consecutive time points while the sparsity constraint restrict each micro-post to match with only a *few* topics. Finally, we showed that topical relation between users is an effective knowledge to detect the user community of organizations.

7.2 Future Work

One can envision several venues for future work. We categorize them into three aspects: organization, user, and content.

Regarding organization, can we define organization-specific models and metrics based on the the business category of the organization? For example the knowledge we discover for a *hotel* could be different from a *telecommunication* organization as for example the patterns of user interactions with these organizations are totally different. This knowledge helps to mine more insight from the data.

Regarding users, one can improve the performance of our community detection algorithm by learning a classifier based on the textual content of the streaming data and utilize this knowledge to initiate the d-node values for user and topic nodes. Regarding, influential user mining with respect to topics, one can develop techniques to control the OR-ness and AND-ness of the current topics in ranking the organization users. The *ordered weighted averaging* (OWA) operators are useful means for this purpose.

Furthermore, in this thesis, we treated the known accounts of organizations as other users. It would be interesting if we study how the activity of these accounts differs from other ordinary users. For example, how different is the content that they produce from the content that other users generate? Do such contents attract more user discussions and reach more audiences? This can help mining the social engagement of the organizations. In addition, can we design algorithms to discover more knowledge about the user community of the organizations and create a virtual profile for each user? Such profiles could contain information from age group, profession, and location of the users to the information about user satisfaction on different services of the organization. Given such user profiles, data mining algorithms can help to extract insight from the data.

Regarding the content, as natural language has its well-known ambiguity issue, it is always a research issue to identify the relevant data about organizations with high accuracy. However, as we showed in this thesis, a more accurate input data (i.e. more relevant data) leads to more accurate topic detection and earlier prediction of emerging topics. So, it will be an important discovery, if we can find other social media signals and sources of information to discriminate relevant from irrelevant data more accurately. Furthermore, can we design algorithms to predict the emergency of the emerging topic? What are the features that should be considered to help early prediction of emerging topics? These all lead to making a better sense of micro-posts in social media for organizations!

Papers arising from this thesis:

- Hadi Amiri, Chen Yan, Anqi Cui, Tat-Seng Chua. Mining the Sense of Organizations in Social Media: Leveraging Organization Users and Terminology. *Submitted to* ACM Transactions on Information Systems (TOIS).

- Hadi Amiri, Tat-Seng Chua. 2012. Mining Sentiment Terminology through Time. In Proceedings of CIKM '12. Maui, Hawaii, USA.

- Hadi Amiri, Tat-Seng Chua. 2012. Mining Slang and Urban Opinion Words and Phrases from cQA Services: An Optimization Approach. In Proceedings of WSDM '12. Seattle, WA, USA.

References

Aggarwal, C.C. 2011. *Social Network Data Analytics*. Springer.

Amiri, Hadi and Tat-Seng Chua. 2012. Mining slang and urban opinion words and phrases from cqa services: an optimization approach. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 193–202, New York, NY, USA. ACM.

Bakshy, Eytan and Jake M. Hofman. 2011. Identifying influencers on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, New York, NY, USA. ACM.

Bickel, J., Alexandre, and B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *Annals of Statistics.*

Blei, David M., Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.

Choi, Y. and C. Cardie. 2009. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

Dempster, A. P. 1968. A generalization of bayesian inference. In *Journal of the Royal Statistical Society*, B(30).

Dhillon, Inderjit S., Yuqiang Guan, and Brian Kulis. 2004. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of ACM SIGKDD.*

Ding, Chris, Tao Li, and Wei Peng. 2008. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Comput. Stat. Data Anal.*, 52(8):3913–3927, April.

Efron, Bradley, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. 2004. Least angle regression. *Annals of Statistics*, 32:407–499.

Esuli, Andrea and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*.

Fraley, Chris and Tim Hesterberg. 2009. Least angle regression and lasso for large datasets. *Stat. Anal. Data Min.*, 1(4):251–259, March.

G., Shafer. 1976. A mathematical theory of evidence. In *Princeton University Press*.

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

Gohr, André, Alexander Hinneburg, Rene Schult, and Myra Spiliopoulou. 2009. Topic evolution in a stream of documents. In *SDM*, pages 859–872. SIAM.

Gruber, Amit, Yair Weiss, and Michal Rosen-Zvi. 2007. Hidden topic markov models. *Journal of Machine Learning Research - Proceedings Track*, 2:163–170.

Gu, Quanquan and Jie Zhou. 2009. Local learning regularized nonnegative matrix factorization. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 1046–1051, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter

Reutemann, and Ian H. Witten. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

Hassan, Ahmed and Dragomir R. Radev. 2010. Identifying text polarity using random walks. In *Proceedings Association for Computational Linguistics (ACL).*

Hofmann, Thomas. 1999a. The cluster-abstraction model: unsupervised learning of topic hierarchies from text data. In *Proceedings of the 16th international joint conference on Artificial intelligence - Volume 2*, IJCAI'99, pages 682–687, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Hofmann, Thomas. 1999b. Probabilistic Latent Semantic Analysis. In *Proceedings of Uncertainty in Artificial Intelligence, UAI*, Stockholm.

Hong, Liangjie and Brian D. Davison. 2010. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88.

Hu, Minqing and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of ACM SIGKDD.*

Islam, Aminul and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Trans. Knowl. Discov. Data*, 2.

Kaji, Nobuhiro and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of HTML documents. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP).*

Kamath, Krishna Yeshwanth and James Caverlee. 2011. Transient crowd discovery on the real-time social web. In *Proceedings of the fourth*

ACM international conference on Web search and data mining, WSDM '11, pages 585–594, New York, NY, USA. ACM.

Kamps, Jaap, Maarten Marx, Robert J. Mokken, and Maarten De Rijke. 2004. Using wordnet to measure semantic orientation of adjectives. In *Proceedings of LREC*.

Kanayama, Hiroshi and Tetsuya Nasukawa. 2006. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Karypis, George and Vipin Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, December.

Kasiviswanathan, Shiva Prasad, Prem Melville, Arindam Banerjee, and Vikas Sindhwani. 2011. Emerging topic detection using dictionary learning. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 745–754, New York, NY, USA. ACM.

Kim, Soo-Min and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of COLING*.

Kleinberg, Jon. 2003. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7:373–397. 10.1023/A:1024940629314.

Kotov, Alexander, ChengXiang Zhai, and Richard Sproat. 2011. Mining named entities with temporally correlated bursts from multilingual web news streams. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 237–246, New York, NY, USA. ACM.

Ku, L.-W. and H.-H. Liang, Y.-T.and Chen. 2007. Question analysis and answer passage retrieval for opinion question answering systems. In *Proceedings of ROCLING*.

Kwak, Haewoon, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA. ACM.

Landauer, Thomas K., Peter W. Foltz, and Darrell Laham. 1998. An Introduction to Latent Semantic Analysis. *Discourse Processes*, (25):259–284.

Lappas, Theodoros, Kun Liu, and Evimaria Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 467–476.

Lee, Changhyun, Haewoon Kwak, Hosung Park, and Sue Moon. 2010. Finding influentials based on the temporal order of information adoption in twitter. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 1137–1138, New York, NY, USA. ACM.

Li, Baoli, Yandong Liu, and Eugene Agichtein. 2008. Cocqa: co-training over questions and answers with an application to predicting question subjectivity orientation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 937–946, Stroudsburg, PA, USA. Association for Computational Linguistics.

Li, Baoli, Yandong Liu, Ashwin Ram, Ernest V. Garcia, and Eugene

Agichtein. 2008. Exploring question subjectivity prediction in community qa. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 735–736, New York, NY, USA. ACM.

Li, Fangtao, Yang Tang, Minlie Huang, and Xiaoyan Zhu. 2009. Answering opinion questions with random walks on graphs. In *Proceedings of Association for Computational Linguistics (ACL).*

Lin, Chih-Jen. 2007. Projected gradipent methods for non-negative matrix factorization. Technical report, Neural Computation.

Liu, Hairong, Longin J. Latecki, and Shuicheng Yan. 2010. Robust Clustering as Ensembles of Affinity Relations. In *NIPS*.

Liu, Kun-Lin, Wu-Jun Li, and Minyi Guo. 2012. Emoticon smoothed language models for twitter sentiment analysis. In *AAAI'12*.

Luxburg, Ulrike. 2007. A tutorial on spectral clustering. *Statistics and Computing*, 17.

Mairal, Julien, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2009. Online dictionary learning for sparse coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 689–696, New York, NY, USA. ACM.

Mairal, Julien, Francis Bach, Jean Ponce, and Guillermo Sapiro. 2010. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, March.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schtze. 2008. *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA.

Mathioudakis, Michael and Nick Koudas. 2010. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 international conference on Management of data*, SIGMOD '10, pages 1155–1158, New York, NY, USA. ACM.

Mohammad, Saif, Cody Dunne, and Bonnie Dorr. 2009. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP).*

Mohtarami, Mitra, Man Lan, and Chew-Lim Tan. 2013a. From semantic to emotional space in probabilistic sense sentiment analysis. In *Proceedings of AAAI.*

Mohtarami, Mitra, Man Lan, and Chew-Lim Tan. 2013b. Probabilistic sense sentiment similarity through hidden emotions. In *Proceedings of Association for Computational Linguistics (ACL).*

Mood, A.M.F. and F.A. Graybill. 1963. *Introduction to the theory of statistics.* McGraw-Hill series in probability and statistics. McGraw-Hill.

Newman, M. E. and M. Girvan. 2004. Finding and evaluating community structure in networks. *Physical Review E*, 69.

Newman, M. E. J. 2006a. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74.

Newman, M. E. J. 2006b. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, June.

Ounis, Iadh, Maarten Rijke, Craig Macdonald, Gilad Mishne, and Ian Soboroff. 2006. Overview of the trec 2006 blog track.

Peng, Wei. 2009. Equivalence between nonnegative tensor factorization and tensorial probabilistic latent semantic analysis. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 668–669, New York, NY, USA. ACM.

Pons, Pascal and Matthieu Latapy. 2006. Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, 10(2):191–218.

Popescu, Ana-Maria and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of HLT-EMNLP*.

Qiu, Guang, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. In *Proceedings of IJCAI*.

Rao, Delip and Deepak Ravichandran. 2009. Semi-supervised polarity lexicon induction. In *Proceedings of the EACL*.

Saha, Ankan and Vikas Sindhwani. 2010. Dynamic nmfs with temporal regularization for online analysis of streaming text.

Saha, Ankan and Vikas Sindhwani. 2012. Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization. In *Proceedings of the fifth ACM international conference on Web search and data mining*, WSDM '12, pages 693–702, New York, NY, USA. ACM.

Sahlgren, Magnus and Jussi Karlgren. 2009. Terminology mining in social media. In *Proceedings of the 18th ACM conference on Information and knowledge management*, CIKM '09, pages 405–414, New York, NY, USA. ACM.

Sakaki, Takeshi, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 851–860, New York, NY, USA. ACM.

Smirnova, Elena. 2011. A model for expert finding in social networks. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, SIGIR '11, pages 1191–1192.

Stone, Philip J. and Earl B. Hunt. 1963. A computer approach to content analysis: studies using the general inquirer system. In *Proceedings of SPRING*.

Strang, Gilbert, editor. 1986. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press.

Takahashi, Toshimitsu, Ryota Tomioka, and Kenji Yamanishi. 2011. Discovering emerging topics in social streams via link anomaly detection. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 1230–1235, Washington, DC, USA. IEEE Computer Society.

Takamura, Hiroya, Takashi Inui, and Manabu Okumura. 2005. Extracting semantic orientations of words using spin model. In *Proceedings of Association for Computational Linguistics (ACL)*.

Tomasoni, Mattia and Minlie Huang. 2010. Metadata-aware measures for answer summarization in community question answering. In *Proceedings of Association for Computational Linguistics (ACL)*.

Tumasjan, A., T.O. Sprenger, P.G. Sandner, and I.M. Welpe. 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, Peter D. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of Association for Computational Linguistics (ACL).*

Turney, Peter D. and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21.

Vasileios, Hatzivassiloglou and McKeown Kathleen. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of Association for Computational Linguistics (ACL).*

Velikovich, Leonid, Sasha Blair-Goldensohn, Kerry Hannan, and Ryan McDonald. 2010. The viability of web-derived polarity lexicons. In *Proceedings of North American Chapter of the Association for Computational Linguistics.*

Wang, Fei, Ping Li, and Arnd Christian Knig. 2011. Efficient document clustering via online nonnegative matrix factorizations. In *SDM*, pages 908–919. SIAM / Omnipress.

Wang, Fei and Changshui Zhang. 2006. Label propagation through linear neighborhoods. In *Proceedings of ICML.*

Wang, Yu, Eugene Agichtein, and Michele Benzi. 2012. Tm-lda: efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 123–131, New York, NY, USA. ACM.

Weng, Jianshu and Francis Lee. 2011. Event detection in twitter. In

Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, *ICWSM*. The AAAI Press.

Wiebe, Janyce and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th international conference on Computational Linguistics and Intelligent Text Processing*, CICLing'05, pages 486–497, Berlin, Heidelberg. Springer-Verlag.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of HLT-EMNLP*.

Yang, Jaewon and Jure Leskovec. 2011. Patterns of temporal variation in online media. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 177–186, New York, NY, USA. ACM.

Yang, Junfeng and Yin Zhang. 2011. Alternating direction algorithms for $\ell1$-problems in compressive sensing. *SIAM J. Sci. Comput.*, 33(1):250–278, February.

Yu, Hong and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, EMNLP '03, pages 129–136, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yu, Xiaohui, Yang Liu, Jimmy Xiangji Huang, and Aijun An. 2012. Mining online reviews for predicting sales performance: A case study in the movie domain. *IEEE Transactions on Knowledge and Data Engineering*, 24(4):720–734.

Zhang, L., R. Ghosh, M. Dekhil, M. Hsu, and B. Liu. 2011. Combining lexiconbased and learning-based methods for twitter sentiment analysis. Technical report, Technical Report HPL-2011-89, HP, 21/06.

Zhao, Wayne Xin, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd European conference on Advances in information retrieval*, ECIR'11, pages 338–349, Berlin, Heidelberg. Springer-Verlag.

Zhu, Xiaojin, Zoubin Ghahramani, and John Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of ICML*.

Appendix A

Derivation for Energy Function

Since $f(x_i) = y_i \; \forall \; i = 1...l$, clearly:

$$\gamma(\mathbf{f} - \mathrm{y})^T \mathbf{I}(\mathbf{f} - \mathbf{y}) = \gamma \sum_{i=l+1}^{n} (f_i - \hat{y}_i)^2$$

Now we need to show that:

$$\mathbf{f}^T \mathbf{\Omega} \mathbf{f} = (1-\gamma) \sum_{i=l+1}^{n} \left(\sum_{j=1}^{l} \alpha w_{ij} (f_i - f_j)^2 + \sum_{j=l+1}^{n} (1-\alpha) w_{ij} (f_i - f_j)^2 \right)$$

By definition $\mathbf{\Omega} = \mathbf{D} - \mathbf{T}$, so we have:

$$\mathbf{f}^T \mathbf{\Omega} \mathbf{f} = \mathbf{f}^T \mathbf{D} \mathbf{f} - \mathbf{f}^T \mathbf{T} \mathbf{f} = \sum_{i=1}^{n} d_{ii} f_i^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij} f_i f_j$$

$$= \tfrac{1}{2} \left(\sum_{i=1}^{n} d_{ii} f_i^2 - 2 \sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij} f_i f_j + \sum_{j=1}^{n} d_{jj} f_j^2 \right)$$

$$= \tfrac{1}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} T_{ij} (f_i - f_j)^2 \right)$$

Considering the symmetric matrix $\mathbf{T}$ defined in Equation (11), we have:

$$\mathbf{f}^T \mathbf{\Omega} \mathbf{f} = \tfrac{1}{2} \left(\sum_{i=1}^{l} \sum_{j=l+1}^{n} T_{ij} (f_i - f_j)^2 \right) +$$

$$
\begin{aligned}
&\qquad \tfrac{1}{2}\left(\sum_{i=l+1}^{n}\sum_{j=1}^{l} T_{ij}(f_i - f_j)^2\right) + \\
&\qquad \tfrac{1}{2}\left(\sum_{i=l+1}^{n}\sum_{j=l+1}^{n} T_{ij}(f_i - f_j)^2\right) \\
&= \sum_{i=l+1}^{n}\sum_{j=1}^{l} T_{ij}(f_i - f_j)^2 + \\
&\qquad \tfrac{1}{2}\left(\sum_{i=l+1}^{n}\sum_{j=l+1}^{n} T_{ij}(f_i - f_j)^2\right) \\
&= \sum_{i=l+1}^{n}\sum_{j=1}^{l} \alpha(1-\gamma)w_{ij}(f_i - f_j)^2 + \\
&\qquad \tfrac{1}{2}\left(\sum_{i=l+1}^{n}\sum_{j=l+1}^{n} 2(1-\alpha)(1-\gamma)w_{ij}(f_i - f_j)^2\right) \\
&= (1-\gamma)\sum_{i=l+1}^{n}\left(\sum_{j=1}^{l} \alpha w_{ij}(f_i - f_j)^2 + \right. \\
&\qquad \left. \sum_{j=l+1}^{n}(1-\alpha)w_{ij}(f_i - f_j)^2\right) \blacksquare
\end{aligned}
$$

Appendix B

Live User-Topic Modeling

Given n input data $\mathbf{A}^t = \{a_1, ..., a_n\} \in \mathbb{R}^{m*n}$ at time t where m is the size of vocabulary, each a_i can be either matched with the already known topics or can potentially be part of a new emerging topic. As we discussed in Chapter 5, let the non-negative matrix $\mathbf{D}^{t-1} \in \mathbb{R}^{m*k^{t-1}}$ represents the k^{t-1} topics found up to time $t-1$. Given $\mathbf{D}^{t-1}$ and $\mathbf{A}^t$, we aim to find $\mathbf{D}^t \in \mathbb{R}^{m*k^t}$. This matrix comprises of the smooth evolution of the k^{t-1} previously known (*evolving*) topics as well as the new (*emerging*) topics identified. $\mathbf{D}^t$ can be learned by minimizing the following optimization problem (Wang, Li, and Knig, 2011; Mairal et al., 2009; Liu, Latecki, and Yan, 2010; Gu and Zhou, 2009):

$$\begin{aligned}(\mathbf{D}, \mathbf{X}) = \arg\min\nolimits_{\mathbf{D},\mathbf{X}} \parallel \mathbf{A}^t - \mathbf{D}\mathbf{X} \parallel_F^2 + \mu \parallel \mathbf{D} - \mathbf{D}^{t-1} \parallel_F^2 \\ +\lambda \parallel \mathbf{X} \parallel_1 \\ s.t. :\ \mathbf{X} \geq 0,\ \mathbf{D} \geq 0,\ \parallel d_j \parallel_2^2 \leq 1\ \forall j \in \{1...k^{t-1}\}\end{aligned} \tag{B.1}$$

where $\mathbf{X} \in \mathbb{R}^{k^{t-1}*n}$ is the weight matrix and $\lambda, \mu \in [0,1]$ are learning parameters. The emerging topics can similarly be learnt. The best representation of each input $a_i \in \mathbf{A}^t$ in terms of $\mathbf{D}^t$ can be obtained as follows:

$$
\begin{aligned}
&x_i = \arg\min_x \parallel a_i - \mathbf{D}^t x \parallel_2^2 + \lambda \parallel x \parallel_1 \\
&s.t.\text{: } x \geq 0
\end{aligned}
\tag{B.2}
$$

where vector $x_i \in \mathbb{R}^{k^t}$ indicates the topics that have been matched with the input data a_i. As such, the most probable topic for the input data a_i considering the weight vector x_i would be topic k where:

$$k = \arg\max_j x_{ij} \tag{B.3}$$

Let u_i be the user who posted the micro-post a_i, we associate u_i to the topic k with a weight of $w_{ik} = x_{ik}$. If a user comment more than once about a topic, we sum up the weights. Later we utilize these weights for our community mining purpose.