The Pennsylvania State University

The Graduate School

TOPIC ORIENTED EVOLUTION AND SENTIMENT ANALYSIS

A Dissertation in

Information Sciences and Technology

by

Bi Chen

© 2011 Bi Chen

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

May 2011

The dissertation of Bi Chen was reviewed and approved* by the following:

Dongwon Lee
Associate Professor of Information Sciences and Technology
Dissertation Advisor, Chair of Committee

Xiaolong (Luke) Zhang
Assistant Professor of Information Sciences and Technology

Prasenjit Mitra
Associate Professor of Information Sciences and Technology

Daniel Kifer
Assistant Professor (James L. Henderson Memorial Professorship)
of Computer Science and Engineering

Ji Fang
Research Scientist of Palo Alto Research Center
Special Member

David Hall
Professor of Information Sciences and Technology
Dean

*Signatures are on file in the Graduate School.

Abstract

Topic modeling techniques help people to understand what is talking about in a corpus, and dramatically improve humans work on academic or business productivity. Although these topic modeling techniques can usually handle topic information represented by word frequencies well, they cannot deal with other unstructured text information such as timestamps, sentiments, or opinions. However, real applications often have needs to explain topics using both structured and unstructured text information together. When texts have their timestamps, it is necessary to identify different topics at different timestamps and how they evolve overtime. When sentiments or opinions are included in texts, it is necessary to identify human's opinion on certain aspects. Toward these challenges, novel models are explored in this thesis to solve problems of topic evolution and aspect-level sentiment analysis.

Two novel models towards topic evolution as follows: (1) The first model, through exploiting social networks in blogsphere, can accurately predict what topics bloggers will talk about in future; and (2) The second model, by applying

citation networks in scientific literature, can identify new research topics and how research topics evolve overtime. Three novel models towards aspect-level sentiment analysis as follows: (3) The first model, by incorporating sentiment lexicons as prior knowledge with machine learning approaches such as Support Vector Machine, can significantly improve the accuracy of sentiment analysis; (4) The second model, through semi-supervised Chinese Restaurant Process, can identify new aspects as well as their sentiments; and (5) The third model, through discovering associations between topic and opinion words, can identify opinionists' standpoints on certain topics.

All Five models are rigorously validated using both real and synthetic experimental data. Experiments on these first proposed models are compared with our baselines, and the third model is compared with the state-of-the-art methods. The first model can predict future topics in blogsphere for next 4 weeks with high precision (0.94). The second model can construct the map of research topic evolution and measure topic influence with accuracy (0.65) comparable to human ratings (0.76). The third model can significantly improve the accuracy of sentiment analysis by 5% compared with the state of arts methods; The fourth model can find new aspects with high precision (0.82) and recall (0.78); The fifth model can find and visualize most controversial topics and extract opinion sentences to represent opinionists standpoints with high accuracy (0.97).

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Topic and Topic Modeling

Topic provides us a way to represent a large volume of unstructured texts. Why can topic do that? The reason is topic reveals the correlations of words. For example, when words like *cosmology, planets, galaxies, asteroids, astrophotography* appears with high frequency, we know the topic "astronomy" is talking about. Those correlations of words are captured by defining topic as a probability distribution over words [1, 2, 3]. Finding topics from documents is the task of topic modeling.

Topic can be modeled by supervised or unsupervised methods. In the supervised context, topics are explicit and have human labeled names. Taking the previous example, "astronomy" is the human labeled name for that topic. Supervised methods, like Native Beyes Classifier (NBC), Support Vector Machine (SVM) [4], can be used for finding correlations between human labeled topics and words. In the unsupervised context, topics are latent and no human labeled names. Although

there is no human labeled name, topics are self explainable. Taking the previous example again, a topic is only a distribution on words, where words like *cosmology, planets, galaxies, asteroids, astrophotography* having higher probability. We can guess this latent topic expresses the topic about "astronomy". Unsupervised methods, like Latent Semantic Indexing (LSI) [1], Probabilistic Latent Semantic Indexing (PLSI) [2] and Latent Dirichlet Allocation (LDA) [3], can be used for finding correlations between latent topics and words.

1.2 Beyond Topic Modeling

Although topic modeling has helped people to understand what is talking about in a corpus, they cannot deal with other unstructured or structured information such as author information, ratings, networks, opinions and so on. How ever, real applications often have needs to explain topics using both structured and unstructured text information together.

Some work has been done on this direction. Author topic model [5] is proposed to model the relations between authors and their research interest topics. Supervised topic model [6] is proposed to model the relations between topics and movie ratings. Several sentiment topic models [7, 8, 9] are proposed to model the relations between topics and human's sentiments or opinions.

However, structured and unstructured text information is so diversity and only above models are not enough to explain them. In this thesis, more powerful models are developed to deal with when the following different information is involved with texts:

1. When timestamps and social networks are involved with texts, how do we

identify topic evolution and the influence of social networks on topic evolution?

2. When timestamps and citation networks are involved with texts, how do we identify evolution of research topics and the influence of citation networks on research topics?

3. The precision of aspect-level [1] sentiment classification is not high, how can we incorporate domain specific lexicons to improve the precision of sentiment classification?

4. Aspects included in online reviews are not limited to predefined aspects. When new aspects appear in online reviews, how do we automatically identify the new aspects for sentiment analysis?

5. When opinions cannot be simply classified as positive or negative, how do we extract those representative sentences to represent opinionists' standpoints?

1.3 Introduction to Topic Evolution

Topic evolution is to solve the problem: what topics will be taking about in the future? Topics are expressed through human writings. Since humans are not independent, their writings are influenced by other humans though relationships, like social network, citation network. In this thesis, how topics evolves will be investigated under the context of social network and citation network.

[1] In the context of sentiment analysis, aspect has the same meaning with topic.

1.3.1 Topic Evolution with Social Network

Since the typical carrier of social network is blogsphere, topic evolution with social network will be investigated on the blog data. Research of topic evolution in the blogsphere is to predict what topics to talk about in the future for the whole blogsphere and for each bloggers. Contents conveyed by blogs provide online advertisers a vessel for effective *targeted advertising* for a new product or service. A model to predict future topics can be used to create a *recommender system* that can help people find potential academic collaborators, business partners, etc. Another potential application is *event detection*. Automated event detection has important uses. For example, a terrorism analyst may not have the time to read the millions of blogs around the world, but automatic event detectors can alert her about an external event. In this thesis, we do not build the end applications for targeted advertising, recommender systems or event detectors, but construct topic predicting [2] models that, we believe, can form the basis of such applications.

Blog data, a collection of formal or informal text communication data that arrive over time, contain more information than just texts. Compared with general web pages, blog data have the following dimensions: *Content Dimension:* topics of the blog posts; *Temporal Dimension:* blog posts are often tagged with timestamps; and *Social Dimension:* blog posts and comments are connected by quotation and by interactions between bloggers and other users via comments. There exists research in: burst detection [10], and trend detection [11], which focus on content dimension; structural and topic evolution/flow pattern extraction [12, 13, 14], which focus on the content and temporal dimensions; social network analysis [15, 16],

[2]Since the target of research on topic prediction and topic evolution is to find what topics will be taking about in the future, they express the same meaning.

which focuses on the social dimension; and the diffusion of information in the blogspace [17], which focus on content and temporal dimensions. Contents are expressed in text, temporal information is expressed through time stamps and social networks are represented by directed graph. Hence, it is a challenge for us how to combine content, temporal, and social dimensions to predict future topics of bloggers. In this thesis, a series of topic prediction models will be developed to predict what topics bloggers will talk about in the future.

1.3.2 Topic Evolution with Citation Network

Since the typical carrier of citation network is scientific literature, the evolution of topics will also be investigated on the scientific papers that are linked each other through citations. Topic evolution in scientific literature shows how research on one topic influenced research on another and helps us understand the lineage of topics. Understanding such topic evolution is an important problem with a few interesting applications. For example, in sociology of science, topic evolution analysis can help us understand and objectively evaluate the contribution of a scientist or an article. Moreover, topic evolution analysis may lead to information retrieval tools that can recommend citations for scientific researchers. Due to its importance and great application potential, topic evolution has recently attracted fast growing interest in the information retrieval community [18, 19, 20, 21, 22, 23, 24]. Existing approaches [24, 25, 26, 27] for topic evolution in scientific literature model a paper as a bag of words, and detect topics on documents in different time periods. Then, topic evolution is analyzed by comparing the changes of topics over time as well as the number of documents of different topics.

A research paper contains more information than just a bag of words. Partic-

ularly, for topic evolution, citations, the important inherent elements in scientific literature, naturally indicate linkages between topics. Surprisingly, citations have not been considered by most of the existing methods for topic evolution. Bolelli et al. [25, 26] propose a segmented author-topic model to identify topic evolution by simply using citations to identify and boost the weight for the top "topic bearing" words in documents. To our best knowledge, no existing work directly infers citations in the Bayesian framework and learns topic evolution over time. Challenges come from two aspects: 1) One challenge is that the impact of citations cannot be captured by casting them in a straightforward manner into a bag of words, and 2) another challenge is the scale problem since there exists a huge amount of literature. In this thesis, a Bayesian citation inheritance topic influence model (c-ITM) will be developed to tackle the problem of topic evolution analysis on scientific literature by leveraging citations.

1.4 Introduction to Aspect-level Sentiment Analysis

Aspect-level sentiment analysis is to find people's opinions or attitudes on certain topics. With the rapid growth of user-generated content on the internet, aspect-level sentiment analysis is becoming more and more important for effective decision making. In this thesis, research of aspect-level sentiment analysis focuses on: 1) how to incorporate lexicons to improve the precision of sentiment classification, 2) how to automatically identify new aspects for sentiment analysis, and 3) how to identify people's standpoints when their opinions cannot be simply judged as positive or negative.

1.4.1 Incorporating Lexicons for Sentiment Analysis

Two typical approaches to sentiment analysis are lexicon look up and machine learning. A lexicon look up approach normally starts with a lexicon of positive and negative words. For instance, *beautiful* is a positive word and *ugly* is a negative word. The overall sentiment of a text is determined by the sentiments of a group of words and expressions appearing in the text [28, 29]. A comprehensive sentiment lexicon can provide a simple yet effective solution to sentiment analysis, because it is general and does not require prior training. Therefore, a lot of attention and effort have been paid to the construction of such lexicons [30, 31]. However, a significant challenge to this approach is that the polarity of many words is domain and context dependent. For example, *big* is positive in *a big victory* and negative in *a big disaster*. Nevertheless, current sentiment lexicons do not capture such domain and context sensitivities of sentiment expressions. They either exclude such domain and context dependent sentiment expressions or tag them with an overall polarity tendency based on statistics gathered from certain corpus such as the web. While excluding such expressions leads to poor coverage, simply tagging them with a polarity tendency leads to poor precision.

Because of these limitations, machine learning approaches have been gaining more and more popularity in the area of sentiment analysis [32, 33]. A machine learning approach such as Support Vector Machine (SVM) does not rely on a sentiment lexicon to determine the polarity of words and expressions, and it can automatically learn some of the context dependencies illustrated in the training data. For example, if *a big victory* and *a big disaster* are labelled as positive and negative respectively in the training data, a learning algorithm can learn that *big* is positive when it is associated with the word *victory* whereas it is negative when

associated with the word *disaster*. Even though recent studies have shown that machine learning approaches in general outperform the lexicon look up approaches for the task of sentiment analysis [32], completely ignoring the advantages and knowledge provided by sentiment lexicons may not be optimal. Alternatively, in this thesis, a method is presented to incorporate sentiment lexicons as prior knowledge with machine learning approaches such as SVM to improve the accuracy of sentiment analysis. This thesis also describes a method to automatically generate domain specific sentiment lexicons for this learning purpose.

1.4.2 Identifying New Aspects for Sentiment Analysis

Extensive research has been done on sentiment analysis. Earlier research in this field focuses on determining whether the overall opinion of a text is positive or negative. However, such general information is insufficient in some practical scenarios. For example, a customer, Peter, wants to buy a camera , and cares more about specific features such as the zoom capability. The overall sentiment about a camera (positive or negative) cannot help Peter to do decision. Hence, it is much more useful to extract both the camera aspects discussed in the reviews and their associated sentiments. The sentiment analysis task described in this thesis focuses on aspect-level sentiment analysis.

The state-of-the-art methods for sentiment classification reported in the literature are supervised machine learning approaches such as Support Vector Machines (SVM) [4]. To satisfy Peter's requirement, we first construct a training data which includes the aspect of the zoom capability, and then train a SVM classifier using this training data. Applied on camera reviews, the trained SVM classifier can tell Peter whether people's opinion on the zoom capability of cameras is positive

and negative. Now, Peter cares more about a specific feature such as water proof, which is not included in the training data. Clearly, the trained SVM classifier cannot satisfy Peter's requirement in this time. This is because the model learned by a supervised approach is completely constrained by the training data. In reality, it is very difficult to construct a training corpus that covers all aspects about a product. Furthermore, new products with new features are continually emerging, which means that applying a sentiment classifier trained on older product reviews cannot correctly identify the new product aspects discussed in the reviews. On the other hand, topic models, such as the Hierarchical Topic Model (HTM) [34], can automatically identify aspects without relying on training dataset. However, as illustrated in our experiments, aspects identified by HTM have not an agreement with human labeled aspects. Furthermore, HTM cannot discriminate which aspects have already been labeled by human, and which aspects are not.

This thesis aims to address above issues by offering a method to automatically identify new product aspects that are not covered in the training data. More specifically, we propose a novel Semi-Supervised Hierarchical Topic Model (SHTM) to identify such new product aspects. To build SHTM, we further propose a Semi-Supervised Nested Chinese Restaurant Process (SNCRP) and use it as the prior for SHTM. To the best of our knowledge, no similar model has been proposed. In addition, we show that SHTM can be used to identify both product aspects and their associated sentiments.

1.4.3 Extracting Representative Sentences for Sentiment Analysis

Current opinion mining work mostly focuses on mining review data for the following reasons: 1) review data widely exists and are easy to obtain; 2) mining review data has their obvious business applications; 3) opinion words used in review normally have obvious sentiment orientations, such as good, bad and so on. However, if we extend opinion mining from the review domain to other domains, the situation becomes more complicated. For example, when a person talks about *iraq war*, someone might say "By removing Saddam Hussein, the world of the future is safer from terrorist attacks.", and others might say "The war will make people live in impoverished circumstances, and create civilian casualties." With regard to these statements, we cannot simply judge them to be either positive or negative.

When an opinionist express her opinion related to a certain topic, she will use some words more frequently than others. Continuing the above example, she will use words like *Saddam* and *war*, which tell people what topics she talks about. But these words are objective and cannot express her personal opinion. An opinionist will choose different words to express her opinion related to *iraq war* based on her stands. If one opinionist cares more about the safety situation, she will frequently use opinion words, like *safe*,*dangerous* and *attack*. If one opinionist cares more about the civilian situation, she will frequently use words, like *civilian*, *impoverished* and *injured*. From the example, we can see that although we cannot judge her opinion to be either positive or negative, we still can find associations between topic words and opinion words with regard to a certain opinion and topic. Such associations will help us to identify different stances among opinionists. In this the-

sis, a generative model will be proposed to find associations between topic words and opinion words with regard to a certain opinionist and topic, and construct a new opinion scoring model based on those found associations. The proposed model will be applied to the political domain.

1.5 Contributions of This Thesis

Specifically, this thesis makes the following contributions:

- To predict future topics in the blogsphere, we proposed the social network and profile-based topic predicting model by integrating content, social and temporal information together. Our proposed models can predict future topics in blogsphere for the next 4 weeks with high precision (0.94).

- To identify topic evolution in scientific literature, we propose a novel inheritance topic model that conceptually captures how citations can be used to analyze topic evolution. We conduct an extensive empirical study using a real dataset of more than 650,000 research papers in the last 16 years and the citation network enabled by CiteSeerX [3]. The proposed model can illustrate the topic evolution path of research papers over the last 16 years, and measure topic inuence with accuracy (0.65) comparable to human ratings (0.76).

- To improve the precision of sentiment classification, we propose a new method to incorporate sentiment lexicons as prior knowledge with machine learning approaches such as SVM. This thesis also describes a method to automatically generate domain specific sentiment lexicons for this learning purpose.

[3] http://citeseerx.ist.psu.edu/

We conduct an extensive empirical study using a real dataset of camera reviews from Amazon. The experiment results show that the accuracy of the classifier can be significantly improved with 5% by incorporating domain specific sentiment lexicons generated by our described approach.

- To identify new aspects for sentiment analysis, we propose a novel Semi-Supervised Nested Chinese Restaurant Process (SNCRP) and Semi-Supervised Hierarchical Topic Model (SHTM). SHTM can automatically decide the number of new aspects and identify new aspects different from pre-defined ones with high precision (0.82) and recall (0.78). In addition, a new combination model is proposed which performs new aspects identification and aspect-level sentiment analysis simultaneously.

- To find opinionist's representative sentences, we propose a generative model to find hidden associations between topic and opinion words in an unsupervised way. The proposed model can find and visualize most controversial topics and extract opinion sentences to represent opinionists standpoints with high accuracy (0.97).

1.6 Outlines of This Thesis

The structure of this thesis as followings. Chapter 2 will discuss related work. Chapter 3 will discuss how to predict future topics in blogspace. Chapter 4 will discuss how to identify topic evolution in scientific papers. Chapter 5 will discuss how to generated domain specific lexicons to improve the precision of sentiment classification. Chapter 6 will discuss how to use Semi-Supervised Chinese Restaurant Process to identify new aspects. Chapter 7 will discuss how to visualize most

controversial topics and extract opinion sentences to represent opinionists standpoints. Conclusion and future work will be discussed in Chapter 8.

Chapter 2

Related Work

2.1 Overview

This chapter discusses the related work of topic evolution and aspect-level sentiment analysis. The emphasis of this chapter focuses on the differences between the models proposed in this thesis with the models proposed by other researchers.

2.2 Related Work: Topic Evolution

Topic evolution has been extensive studied in recent years. In this thesis, new models are proposed to investigate how topics evolve with social network and citation network. In the followings, related work for each model will be discussed, and the difference of our models from ones proposed by other researchers will be highlighted.

2.2.1 Topic Evolution with Social Network

To investigate topic evolution in blogsphere, social network and prole-based topic prediction model is proposed. Blog data have three dimensions: temporal, content, and social dimensions [1]. Different from our model, no previous work has studied all of the temporal, content, and social dimensions and their correlations for topic predicting.

Earlier research work on blogsphere does not consider contents, and only investigates the graph structure of blogsphere. Kumar et al. [10] modeled the blogsphere as a graph of bloggers connected by hyperlinks and studied the evolution of the graph in terms of graph properties such as in-degree, out-degree, strongly connected components, and communities. Gruhl et al. [17] studied the dynamics of information propagation in two levels: a macroscopic characterization of topic propagation and a microscopic characterization of propagation from individual to individual, using the theory of infectious diseases to model the flow. Adamic and Glance [35] studied the linking patterns of political bloggers to uncover any differences in the structure of the two communities. Licamele and Getoor [36] presented a definition of social capital, and investigate the friendship relations as well as the organizer and participation relations from the social network. They show that social capital is a better publication predictor than publication history in real academic collaboration networks. However, the above social network based blog analysis approaches ignored the fact that the content, social, and temporal dimensions of blogs are interrelated and they assumed that these dimensions are independent.

There are works using content analysis as well. Traditionally, these approaches

[1] For detail, please refer to chapter 1.

are based on simple counts of entries, links, keywords, and phrases [37, 10, 17]. More recently, Chi et al. [11] introduced the *eigen-trend* concept to represent the temporal trend in a group of blogs with common interests using the singular value decomposition and higher-order singular value decomposition. Qamra et al. [16] proposed a Content-Community-Time model, which clusters the posts according to their contents, timestamps and the community structures, to automatically discover stories. In their approaches, only links between posts are taken into consideration. Shen et al. [38] proposed three novel approaches to find latent friends, which share the similar topic distribution in their blogs, by analyzing the contents of their blog entries. However, the above approaches mainly focus on either the content of blogs or combining social or temporal information to improve content analysis.

2.2.2 Topic Evolution with Citation Network

More research work on topic evolution has been done over scientific literature. The uniqueness of scientific literature is citation network. To investigate topic evolution with citation network, citation inheritance topic model (c-ITM) is proposed. c-ITM is distinguished from the previous work in three ways: 1) c-ITM considers both content and citations in a full-generative inheritance topic model, 2) c-ITM is inferred directly in the Bayesian framework, and 3) c-ITM can explicitly extract the relationship between topics.

One branch to study topic evolution uses discriminative approaches and treats each topic as a distribution over words or a mixture over documents. Morinaga and Yamanishi [21] used a finite mixture model to represent documents at each discrete time. Their algorithm detects topic changes on certain documents if the

topic mixtures drift significantly from the previous ones. Mei and Zhai [18] conducted sequential clustering and then correlated clusters via a temporal graph model, which was in turn used to represent the topic evolutions in a document stream. Mei et al. [19] used a probabilistic approach to detect spatiotemporal theme patterns and then observed the evolution of theme patterns by comparing the theme life cycles and theme snapshots. Spiliopoulou et al. [23] detected and tracked changes in clusters based on the content of the underlying data stream. Schult and Spiliopoulou [22] used a clustering approach to find out the ontology/ taxonomy evolution for documents.

Recently, more studies used generative topic models to observe topic evolution on document streams. Zhou et al. [24] used the LDA model to observe temporal topic evolution over scientific literature. Specifically, a k-component LDA model is constructed over the whole dataset to generate k global topics. For each topic, the trend is obtained by simply counting the number of papers belonging to the topic year by year. The author information is also used to explain why some topics tend to decline yet some others expand. Blei and Lafferty [39] developed a dynamic topic model (DTM) by assuming that topic models evolve gradually in time and are distributed normally. Specifically, a k-component LDA analysis is conducted at each time slice t. Each topic is modeled as a Gaussian process centered upon the previous value. Similar to [39], the topic is global and the topic trend is obtained by counting the number of papers. The dynamic topic model assumes that all papers at time t are correlated to all papers at time $t-1$ In our work, only cited papers at time $t-1$ are related to their citing papers at time t. Wang et al. [40] further extended this discrete DTM to a continuous version. Morchen et al. [20] used probabilistic topic models to annotate articles with the most likely

ontology terms. They also proposed a solution for automatically determining how new ontology terms can evolve from old terms. AlSumait et al. [41] extended the LDA model to an online version by incrementally updating the current model for new data and claimed that this model has certain ability of capturing the dynamic changes of topics. Gohr and Hinneburg [42] used latent variables to index new words while deleted those outdated words within a sliding window for a stream of documents. Those indexed new words were used to portray the topic changes for the information retrieval domain. Bolelli et al. [26, 25] proposed a generative author topic model that integrated the temporal ordering of the documents to model topic trends sequentially, where the discovered topics at an early time were propagated to influence the topics generated later. They use citations to identify *topic-bearing* words whose weights should be doubled. Mann et al. [43] used an n-gram topic model to identify the influence of one topic on another. However, these approaches modeled citations indirectly in their topic models, and the resulting topic influence is also time irrelevant.

2.3 Related Work: Aspect-level Sentiment Analysis

Sentiment analysis has attracted more and more attention of researchers in recent years. Regarding with the complexity of human natural language, the problem of sentiment analysis is far from solved. In this thesis, three different models/approaches are proposed to deal with three problems of sentiment analysis: 1) how to incorporating lexicons, 2) how to identify new aspects, and 3) how to extract representative sentences. In the followings, related work for each model/approach

will be discussed, and the difference of our work from previous studies will be highlighted. For a long and comprehensive survey, please refer to [44].

2.3.1 Incorporating Lexicons for Sentiment Analysis

As discussed in chapter 1, two typical approaches to sentiment analysis are lexicon look up and machine learning, and each of these two approaches has its own advantages and drawbacks. However, few studies have devoted to combining these two approaches to improve sentiment classification. [45] explores using a general purpose sentiment dictionary to improve the identification of the contextual polarity of phrases. A few recent studies [46, 47, 48] have shown that incorporating a general purpose sentiment lexicon into machine learning algorithms can improve the accuracy of sentiment classification in the document level. In all of these works, a general purpose sentiment lexicon contains words with context/domain independent polarities. Our work differ from these previous studies in the following ways.

First, unlike the previous works in which only a general purpose sentiment lexicon is used, we incorporate not only a general purpose sentiment lexicon but also *Domain Specific Sentiment Lexicons* into SVM learning to improve the accuracy of sentiment classification. The domain specific sentiment lexicons include lexicons indicating various topics or domains as well as lexicons consisting of words or phrases with polarities associated with a particular domain or topic. For example, in our experiment, we built domain specific lexicons regarding 'Battery Life' which include a lexicon of words such as *battery* and a lexicon of words or phrases such as *quickly:negative* and *long:positive*. The first lexicon consists of words or phrases that are good indicators for the topic of 'Camera Battery Life', and the

second lexicon consists of words or phrases with polarities specific to the topic of 'Battery Life'. For instance, *quickly* and *long* may not carry negative and positive sentiments in a different domain. They can also carry opposite sentiments if the domain is different. More importantly, our experiment results show that while a general purpose sentiment lexicon provides only minor accuracy improvement, incorporating domain specific dictionaries leads to more significant improvement for our sentiment classification task.

Second, most of the previous related works explores the advantages of incorporating lexicon knowledge to improve sentiment classification at the document level, namely, to classify an entire document to be either positive or negative. Compared to these works, our sentiment classification task is more fine grained. Our sentiment classification is performed at the sentence level, and for each sentence, we not only predict whether a sentence is positive, negative or objective, we also predict the main topic associated with that sentiment. Our experiments demonstrated that the domain specific dictionaries we built lead to improvement for both of these tasks.

Regarding the construction of sentiment lexicon, previous studies have been focusing on generating general purpose dictionaries. These methods range from manual approaches [30]to semi-automated [49, 50, 51] and automated approaches [31]. In this thesis, we present a method to build domain specific sentiment lexicons using a combination of corpus filtering, web searching using linguistic patterns and dictionary expansion techniques.

2.3.2 Identifying New Aspects for Sentiment Analysis

In this thesis, a new semi-suppervised Nested Chinese Restaurant Process is proposed to identify new aspects for sentiment analysis. The differences of our work from all previous supervised and unsupervised sentiment analysis are in three aspects: 1) we focus on new aspect identification, 2) the number of new aspects are determined automatically, and 3) a new combination method is proposed, which obtains benefits of both unsupervised and supervised aspect-level sentiment analysis, and promotes the performance of the-state-of-the-art method.

One branch research on aspect identification focuses on unsupervised methods. Some research work is to Natural Language Process (NLP) techniques to identify which aspects is talking about. In [52], authors used POS tagging and word/phrase frequencies to identify product aspects. More complex, authors used phrase dependency parsing for aspect identification. A more recent work is [53], where authors used phrase dependency parsing to find candidate aspects names and then used relations on candidate aspects to filter noisy aspects names. Another kind of automatic aspect identification applies bayesian topic model. In [54], authors proposed a joint sentiment/topic model which detects sentiment and topic simultaneously from text. In [55, 9], authors proposed multi-gain topic model to extract extract ratable aspects. However, there exist three big problems for aspects identified by topic model: 1) Aspect number need to be determined manually; 2) Not all identified topics are interpretable, and noisy topics also exist; and 3) some aspects costumers may concern with may not be identified because they belongs to small topics.

Another branch focuses on supervised methods. Supervised classification algorithms, such as multinomial Naive Bayes (NB), Maximum Entropy (ME) and

Support Vector Machines (SVM), are normally applied. In the framework of supervised methods, extra knowledge is incorporated to get a higher precision. In [46], authors incorporated lexicon with a multinomial Naive Bayes classifier for aspect-leval sentiment analysis. In [56], authors combined different features, like word tokens, sentiment words and phrases, with ME model to obtain better results. Compared to NB and ME, SVM usually obtain higher precision [32]. Based on using SVM, authors used different methods to promote the precision of SVM. In [57], authors used automatically generated rationales to improve SVM on sentiment classification. In [58], authors used Amazon Mechanical Turk to remove non-information texts to obtain a higher precision on aspect-level sentiment classification. However, the main problem is that identified aspects by supervised methods are limited to pre-defined ones, and cannot identify new aspects different from pre-defined ones. Notice that some semi-sueprvised methods exist for sentiment analysis, like[59, 48]. None of these semi-supervised work is concerned with new aspect identification.

2.3.3 Extracting Representative Sentences for Sentiment Analysis

Opinion mining has been extensively studied in recent years. The most related work to ours is aspect-level opinion mining. For a general survey, please refer to [44]. Our work is different from existing work in two main aspects: 1) Our proposed model identifies topics and associations between topics words and opinion words simultaneously , and 2) our approach does not require topic, product aspect sets and opinion word sets to be manually defined in advance.

The early representative work includes [52] which uses association rule mining method, and [60], which uses template extraction method. Their methods explored associations between product aspects and opinion words based on their explicit co-occurrence. Although they did a good job in identifying pre-defined product aspects, they could not detect aspects that were not pre-defined. They identified product features by applying the synonym set in WordNet [61] and the semiautomated tagging of reviews. Our method finds topic or aspect sets automatically.

Topic-Sentiment Model [7] calculate sentiment coverage of documents by joint modeling the mixture of topics and sentiment predictions. But their model requires post-processing to calculate sentiment coverage of documents. Rather than post-processing, Joint Sentiment/Topic model [54] can directly predict the sentiment orientation in the document level. Considering the hierarchy structure between objects and their associated aspects, Titov and McDonald [55] proposed the Multi-Grain Latent Dirichlet Allocation model to find ratable aspects from global topics. Later, they proposed Multi-Aspect Sentiment model [9] which summarizes sentiment texts by aggregating on each ratable aspects. However, in above work, researchers did not identify the associations between topics and sentiments. Our work identifies those associations automatically.

Previous works do not identify hidden relations between topics/aspects and opinion words. [62] proposed a latent variable model to predict semantic orientation of phrases by finding associations between noun clusters and adjective clusters. However, their work did not cluster adjective words which lead to sparsity problem. [63] and [64] clustered opinion word into groups and then found hidden associations between topics/aspects and opinion word groups by mutual reinforcement

and information bottleneck algorithm respectively.

However, their work need to predefine sets of words specifying positive and negative. Our goal is different. We aim to extract opinions different from finding positive and negative sentences because we cannot easily use positive or negative criteria onto sentences in the field like politics.

Chapter 3

Topic Evolution with Social Network

3.1 Overview

This chapter focuses on the problem of *how topics will evolve in the future*, that is *what topics bloggers will talk about in the future*? A formal problem definition will be given at first. Then, three topic predicting models, including *general model*, *profile-based model*, and *social network and profile-based model* will be discussed in detail. Finally, models will be evaluated on a real large dataset, Dailykos[1].

3.2 Problem Definition

The topic predicting models are to predict future topics within and across different bloggers over the temporal dimension and social dimension. There exists no automatic or systematic process for constructing topic predicting models by analyzing the social, content, and temporal information embedded in a historical blog corpus together. The definition of the problem is:

[1] http://dailykos.com

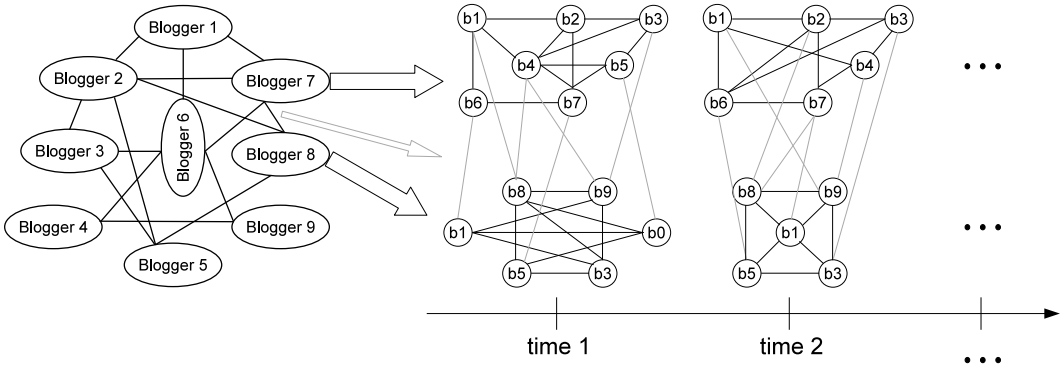

Figure 3.1. Graph Representation of Blog Data

Definition. *given the topics that were discussed in a community blog from the past time to now, how do we predict the topics that will be discussed in the future for the whole community blog, and for any given individual blogger.*

3.3 Topic Predicting Models

In this section, first we present an example collection of blog dataset and its corresponding graph representation. Then we introduce how to extract features for different models, and finally we review regression techniques which construct our models in brief.

3.3.1 Blog Data and Representation

In this chapter, we choose the political blog, DailyKos, as an example dataset. We collected *249,543* blog entries from *October 12, 2003* to *October 28, 2006*. Since some authors blog infrequently, in our experiments, authors with less than *45* blog entries are deleted. As a result, there are *131,869* blog entries left with *1,287* authors and *1,008,467* comments.

The blog dataset can be represented as a hyper-graph, where each hyper-node

represents a blogger and each hyper-edge denotes the connections between bloggers such as comments and quotations as shown in Figure 3.1. Specifically, each hyper-node is a sequence of graphs of his/her own blog entries over time. The hyper-edge consists of a set of edges that connects nodes in these graphs. For example, in Figure 3.1, the hyper-node $Blogger_7$ and $Blogger_8$ are represented as two sequences of blog entry graphs in the right hand side. At each time point or within in a time window(a given length of time), there will be a set of edges that links blog entries from one blogger to another blogger, which is represented as the gray lines in right side of this figure. In this work, we propose to represent individual blog entries in a higher level: topic, which indicates the subject of blog entries instead of concrete contents. The tools for data clustering, CLUTO [2], is used to partition blog entries into topics. Now each node is a topic and each edge represents the quotation between topics, and the hyper-edge now represents a sequence of edges, which denote the links between different bloggers at the topic level at different time points.

3.3.2 General Topic Predicting Model

The general topic predicting model is proposed to capture what topics to talk about in the entire blogspace. That is, given the list of topics that were discussed in the previous time windows, we want to predict what kinds of topics will be more likely to be discussed in the next time window. The general topic predicting model is used to monitor and predict the general trend and transition in the entire blogspere instead of that of any individual blogger.

All blog entries are first clustered into a set of topics, and then each blog entry

[2]http://glaros.dtc.umn.edu/gkhome/views/cluto

is represented by a topic. To identify the general topic features, the historical data is first partitioned into a sequence of time windows on a daily, weekly, or monthly basis. For each time window z, the content of the blog entries is represented as a topic distribution vector $\vec{T_z} = < t_1, t_2, t_3, \cdots, t_n >_z$ that represents the distributions of blog entries with respect to the list of topics, where n is the number of topics, t_i represents the weight of the ith topic within time window z. The ith component of a topic distribution vector can be calculated as the total number of blog entries belonging to ith topic divided by the total number of blog entries in time window z. Hence, the weight of each topic is the normalized value of the number of blog entries in that topic and the sum of the weights is *1*. Since a topic distribution vector can be build for each time window, general topic features will be achieved in terms of a time series of topic distribution vectors.

Based on the general topic features $\vec{T_z}$, we can train the general topic predicting model and predict future topics of the entire blogspere by using regression techniques. We take the previous k topic distribution vectors $\vec{T_z}$, from *z-k+1*th time window to the zth time window, as the input vectors, and take the topic distribution vector $\vec{T}_{z+1}$ in the *z+1*th time window as the target vector to train the model. Then, using trained regression model, the hidden transitions relations between topics can be estimated and used to predict topic distribution at the next time window.

3.3.3 Profile-Based Topic Predicting Model

Different bloggers have different background and interests, hence they have different topic patterns where we can not simply use the general topic predicting model to predict future topics of individual bloggers'. The intuition is that what a blog-

ger post in his blog entries depends on not only the overall trend of topics in the whole blogspace, but also his/her own interests.

As a result, not only the general topic distribution vector but also the profile of the corresponding user are used as the input to the regression model. For the general topic distribution, we can use the topic distribution vector in the previous section. For the profile-based topic distribution, we propose to add personal topic distribution vector $T_p\vec{(}j)_z$ to general topic features, $T_p\vec{(}j)_z = < t_{1j}, t_{2j}, t_{3j}, \cdots, t_{nj} >_z$, where t_{ij} represents the distribution of topic i for blogger j within time window z. Here the weight of t_{ij} is calculated as the percentage of blog entries posted by blogger j and belong to topic i (denoted as $|t_{ij}|$) against the total number of blog entries posted by blogger j (denoted as $|t_j|$) in the time window z.

However, from the dataset we observed that sometimes, within a time window, a blogger has no blog entries at all. Then, we propose to approximate the topic distribution vector for bloggers that have no blog entries with respect to his previous topic distribution vector and a decay factor. The intuition is that the topic distribution vector will decay to the vector $< \frac{1}{|T|}, \frac{1}{|T|}, \cdots, \frac{1}{|T|} >$, which means the blogger does not prefer any topics.

Formally:

$$t_{ij} = \begin{cases} \frac{|t_{ij}|}{|t_j|},\ if\ t_j \neq 0 \\ t'_{ij} \cdot e^{-\lambda} + \frac{1}{|T|} \cdot (1 - e^{-\lambda}),\ if\ t_j = 0 \end{cases} \tag{3.1}$$

where λ is the decay factor, t'_{ij} is the weight of topic i for blogger j in the previous time window, and $|T|$ is the total number of topics.Note that $\vec{T_p}(j)_z$ is normalized such that the sum of the weights is *1* for the second case.

Based on the profile-based topic features $< \vec{T}_z, \vec{T}_p(j)_z >$ for blogger j, we can train the profile-based topic predicting model, and predict future topics of blogger j by using regression techniques. We take the previous k combined vectors $< \vec{T}_z, \vec{T}_p(j)_z >$, from *(z-k+1)*th time window to the zth time window, as the input vectors, and take the combined vector $< \vec{T}_{z+1}, \vec{T}_p(j)_{z+1} >$ in the *(z+1)*th time window as the target vector to train the model. Then, using trained regression model, the future topics of blogger j can be predicted based on historical general topic features and his/her own historical topic features.

Besides posting blog entries, a blogger also posts comments to blog entries written by other bloggers. We improve the profile-based topic predicting model by adding another comment distribution vector. We simply treat a comment having the same topic as the corresponding blog entry. That is, if a comment written to a blog entry which is on topic i, this comment is considered on topic i too. Comment distribution vector can be represented as $\vec{C}_p(j)_z = < c_{1j}, c_{2j}, c_{3j}, \cdots, c_{nj} >_z$, where c_{ij} represents the distribution of comment on topic i for blogger j within time window z. Here the weight of c_{ij} is calculated as the percentage of comments, belonging to topic i (denoted as $|c_{ij}|$), posted by blogger j, against the total number of comments posted by blogger j (denoted as $|c_j|$) in the time window z.

By adding the comment distribution vector to the profile-based topic features, we get the improved profile-based topic predicting model. We treat the improved profile-based topic features $< \vec{T}_z, \vec{T}_p(j)_z, \vec{C}_p(j)_z >$ as the same way to train the regression model.

3.3.4 Social Network and Profile-Based Topic Predicting Model

In the profile-based topic predicting model, the assumption is that each individual blogger is independent or each blogger contributes equally to the general topic transition. However, in reality, this is not always true. Usually, not only the overall topic transition and the profile of the bloggers, but also the social neighbors and their blog entries affect the topics, of which a blogger's blog entries will talk about. The reason is that bloggers that are socially connected share similar interests and profiles. As a result, we propose the social network and profile-based topic predicting model, by adding social network features of a blogger to the improved profile-based topic predicting model.

Here, social network refers to the relations between bloggers created by comments and quotations in blog entries. Besides the general topic distribution, topic distribution and comment distribution of individual bloggers, a list of social neighbors with the weighted relations and their topic distributions are added as the input to the regression model as well. Specifically, the social network features of a blogger j in time window z are represented as a vector $\vec{S}(j)_z = < s_{1j}, s_{2j}, s_{3j}, \cdots, s_{nj} >_z$, where

$$\vec{S}(j)_z = \sum_{x=1}^{m} \frac{C_{j \to x}}{TC_j} \cdot \vec{T}_p(x)_z, TC_j = \sum_{x=1}^{m} C_{j \to x} \tag{3.2}$$

m is the total number of social neighbors of blogger j in the network, $C_{j \to x}$ represents the number of comments written by blogger j to blog entries posted by blogger x in a certain time window, and TC_j represents the total number of comments written by blogger j in the same time window.

Based on the social network and profile-based topic features $< \vec{T}_z, \vec{T}_p(j)_z, \vec{C}_p(j)_z, \vec{S}(j)_z >$ for blogger j, we can train the social network and profile-based topic predicting model, and predict future topics of blogger j by using regression techniques. We take the previous k combined vectors $< \vec{T}_z, \vec{T}_p(j)_z, \vec{C}_p(j)_z, \vec{S}(j)_z >$, from *(z-k+1)*th time window to the ith time window, as the input vectors, and take the combined vector $< \vec{T}_z, \vec{T}_p(j)_z, \vec{C}_p(j)_z, \vec{S}(j)_z >$ in the *(z+1)*th time window as the target vector to train the model. Then, by using trained regression model, the future topics of blogger j can be predicted based on historical general topics, his/her own historical topics, and his/her neighbors' historical topics.

3.3.5 Regression Techniques Used In Topic Predicting Models

For time series regression, traditional feed-forward network learning algorithms, like back-propagation algorithm, are normally used for prediction. However, considering the speed and adaptation problems of traditional feed-forward network learning algorithms, we will choose two different regression techniques in our topic predicting models: Extreme Learning Machine (ELM) [65], and Modified General Regression Neural Network (MGRNN) [66]. ELM has extremely fast learning speed which is thousands of times faster than traditional feed-forward network learning algorithm, as well as reasonable precision. MGRNN is presented as an easy-to-use 'black box' robust tool which can compete with optimized feed-forward networks, as well as reasonable speed and no adaptation required by the users.Because of the limit space available, we review ELM and MGNN techniques in brief. For more information, please refer to [65] and [66].

3.4 Performance Evaluation

3.4.1 Evaluation Standards

In this section, we evaluate the proposed topic predicting models on the Dailykos dataset. As the largest political blog web site, Dailykos can be a representative to investigate topic evolutions in blogspace. To evaluate the quality of the predicted future topics, we define *precision* as the similarity between the predicted vector and the ground truth is calculated as the metric.

$$Precision = Sim(\vec{T}', \vec{T}) = \frac{\vec{T}' \cdot \vec{T}}{|\vec{T}'||\vec{T}|} \tag{3.3}$$

The content of the Dailykos blog dataset focuses on political issues. It is reasonable to cluster the total blog entries into a small number of topics. Because the results we found from the experiments are not influenced by the number of topics, in the following experiments, we clustered the total blog entries into *30* topics and achieved well results. On the time dimension, we partitioned into *159* weeks, where blog entries within the same week are taken as equal in the temporal dimension. The first *139* weeks are taken as training data and the last *20* weeks are taken as testing data. In the following experiments, λ and η are set to *0.2* and *0.8*, respectively. *1 week* refers to the approach that uses only data in the previous week to predict topic pattern in the next week, *3 weeks* refers to the approach that uses the data in the previous 3 weeks to predict the topic pattern in the next week, similarly *5 weeks* and *10 weeks* are defined.

Further more, the selected *1287* bloggers are ranked according to the number of blog entries they have posted during past 159 weeks. In our evaluation phrase, top

50 bloggers who post blog entries larger than 325 are defined as the most active bloggers; bloggers ranked between 51 to 150 are defined as active bloggers who post blog entries less than 325 but larger than 146; bloggers ranked between 151 to 300 are defined as less active bloggers who post blog entries less than 146 but larger than 80; the rest of 787 bloggers are defined as the least active bloggers who post blog entries less than 80.

3.4.2 Evaluating and Comparing Models

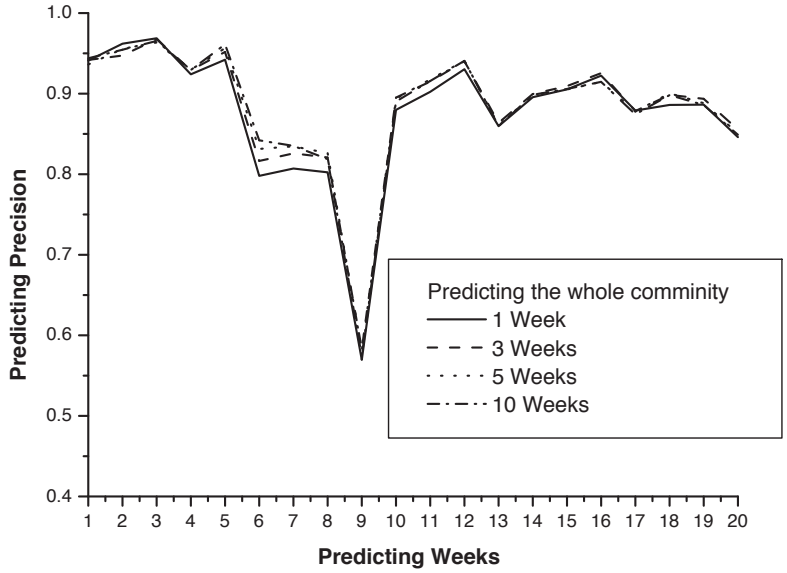

Figure 3.2. General Topic Predicting Model (Predicting community)

Figure 3.2 shows the general topic predicting model with MGRNN regression in the overall blogspere to predict the whole community. X axis refers to the distance between the week being predicted and the last week in the training data. It can be observed that the prediction based on *10* weeks is the best and all the

four approaches produce very accurate (> 0.9) prediction for the subsequent *4* weeks. That is, the more historical information being used, the more accurate is the prediction of future topics. However, the precision promoted by using more historical information is not evident. It is interesting to notice that, the precision for predicting the *9*th week (from Aug 08, 2006 to Aug 15, 2006) drops dramatically. In reality, a political event happened on Aug 10, 2006, when three-time Senator, Joseph Lieberman, lost his re-election campaign to political newcomer Ned Lamont. [3] A great number of blog entries began to talk about this unpredictable event, which causes the precision for prediction drops down. When the effects of this event subside, the precision for prediction goes up again.

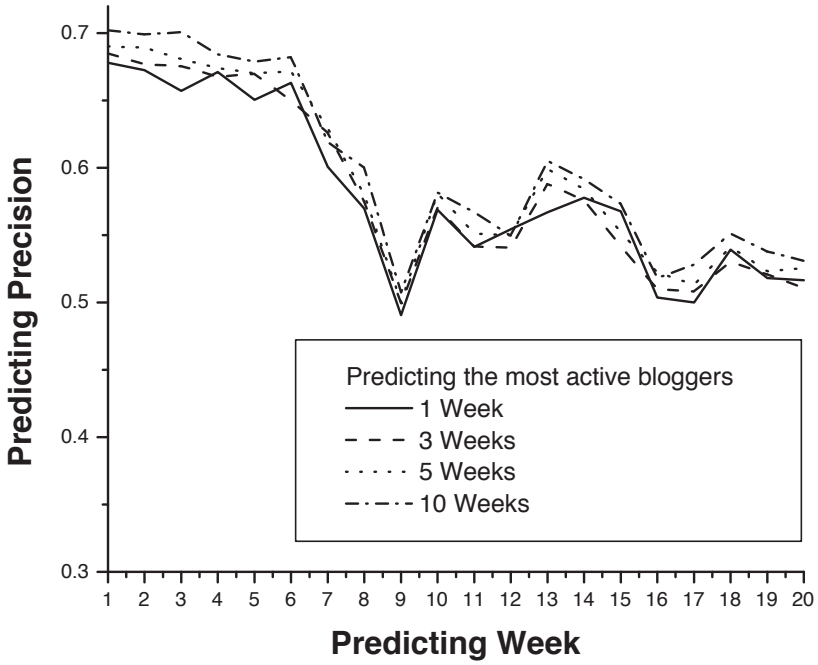

Figure 3.3. Profile-based Topic Predicting Model (Predicting individuals)

[3]http://transcripts.cnn.com/TRANSCRIPTS/0608/09/ltm.08.html

Figure 3.3 shows the average precision of the profile-based topic predicting model for the most active bloggers. It can be observed that the model can accurately predict *6* subsequent weeks ($precision > 0.7$) using *10* weeks of historical data. However, the precision promoted by using more historical information is not evident as shown in figure 3.2. In the following figures, all experiments are using *10* weeks of historical data.

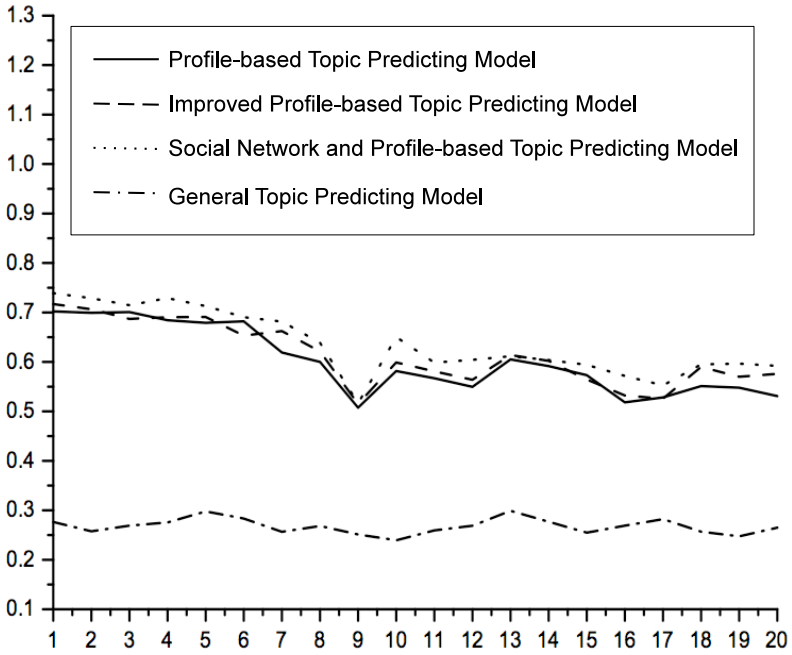

Figure 3.4. Comparison of the models (Predicting individuals)

The general topic predicting model performances well for the whole community. However, considering the diversity of individual bloggers, we can not only use one general model to predict future topics of any individual bloggers. In experiments, we choose only *50* bloggers as the most active bloggers, the blog entries posted by these bloggers almost consist of *30%* of total blog entries. Figure 3.4 (the last line) shows the average precision to predict future topics of the most active

bloggers by using the general topic predicting model. Obviously, general model does not perform well on individual level. Hence, we use the profile-based topic predicting model, and the social network and profile-based topic predicting model for predicting future topics of individual bloggers. Because of limit space, we show the compared results of these models in the same graph.

Figure 3.4 shows the average of precision of the proposed topic predicting models for the most active *50* bloggers. It can be observed that the general topic predicting model is more accurate and stable than other models. The reason is that the general trend of topics of the entire blogsphere is more robust to noises; whereas for the group of the most active bloggers, their predicted future topics are more sensitive to noise and subjective. Generally, using social network topic features improves the quality of prediction as shown in Figure 3.4, while comment distribution features used in the improved profile-based topic predicting model do not promote the precision of prediction evidently. It is interesting to notice that precision for prediction in the *10*th week goes up again. The reason is that the (improved) profile-based topic predicting model, and the social network and profile-based topic predicting model have incorporated the general topic features as their background information. Since the general topics features on the *10*th week imply the unpredictable election campaign event, the precision for prediction goes up again. However, with the subsidence of this event, the general topic features do not imply any particular event, the precision for prediction goes up again.

Although using MGRNN regression techniques we can achieve good results, training and testing these topic predicting models for the most active bloggers spends too much time. In practice, it is not efficient to train and testing models for each blogger. Therefore, we choose ELM regression techniques to train and

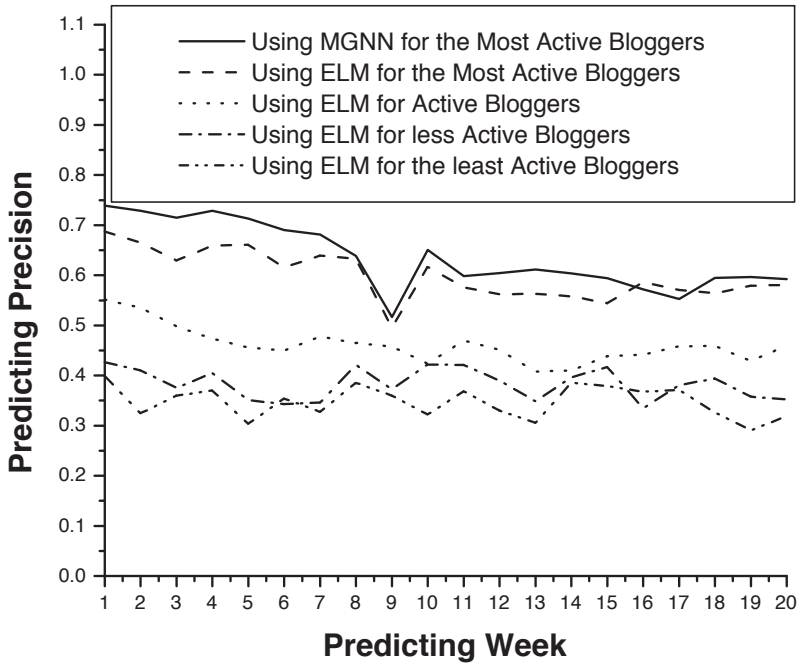

Figure 3.5. Predicting Future Topics of Bloggers At Different Active Levels (Predicting individuals)

test all bloggers. From Figure 3.5 and Figure 3.6, we see that MGRNN regression achieves a little bit better quality than ELM regression when they are used on the most active bloggers and active bloggers. And for less active bloggers and the least active bloggers, ELM and MGRNN regression achieve similar quality. However, from table 3.1, we can see the ELM regression is almost 500 hundred of times faster than MGRNN regression. Combining the precision and efficiency into consideration, we think the social network and profile-based topic predicting model with ELM regression is the best model of all our proposed models.

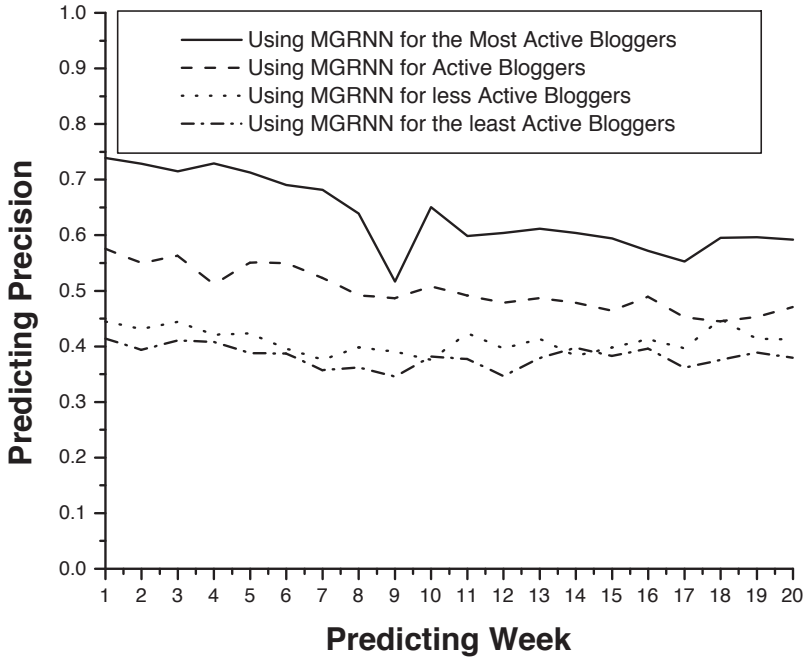

Figure 3.6. Predicting Future Topics of Bloggers At Different Active Levels (Predicting individuals)

3.5 Conclusion

In this chapter, we propose to predict future topics over blogspace from multiple dimensions: temporal, content, and social dimensions. Experiments with real blog dataset show that our topic predicting models produce promising results of predicting future topics. In the future, we will do more experiments of our topic predicting models on other kinds of blogsphere.

Time (Minutes)		Social network & profile-based	
		Train Time	Test Time
The Most Active Bloggers	MGRNN	131.42	0.03
	ELM	0.25	0.03
Active Bloggers	MGRNN	389.25	0.09
	ELM	0.74	0.09
Less Active Bloggers	MGRNN	781.82	0.17
	ELM	1.52	0.17
The Least Active Bloggers	MGRNN	1991.27	0.43
	ELM	3.67	0.43

Table 3.1. Time Comparison between MGRNN regression and ELM regression

Chapter 4

Topic Evolution with Citation Network

4.1 Overview

This chapter focuses on the problem of *how topics evolve in citation networks*? A formal problem denition will be given at rst. Then, a series of citation-unware and citation-aware models are developed to model how topics evolve with citation network. Finally, models will be evaluated on a real large dataset, CiteSeerX[1].

4.2 Problem Definition

Let $\mathbb{W} = w_1, ..., w_v$ be a vocabulary set. A (probabilistic) vocabulary distribution on $\mathbb{W}$ is a point in the $V-1$ dimensional simplex, functioned as $\mathbf{f} : W \rightarrow [0; 1]$ such that $\sum_{w \in \mathbb{W}} (f)(w)$. A vocabulary distribution f can also be written as a vector $\mathbf{f} =< w_1 : \mathbf{w_1}, ..., w_V : \mathbf{w_V} >$. For two vocabulary distributions f and g, the

[1] http://citeseerx.ist.psu.edu/

similarity between them is modeled as the cosine similarity: $sim(\mathbf{f}, \mathbf{g}) = \frac{\mathbf{f} \cdot \mathbf{g}}{\mathbf{f} \cdot \mathbf{g}}$.

Let $D = d_1, ..., d_m$ be a set of scientific publication corpus in question. A document d consists of a vocabulary distribution, a citation set L_d, and a timestamp. A topic z is a vocabulary distribution. Intuitively, a topic is popular if it is similar to many documents in D. Imagine that we virtually combine all documents in D into a single long document d'. We can get a word vector $\mathbf{w}$ for d'. Each element of $\mathbf{w}$ is a word from d'. If a word w appears n times in d', then there are n duplicates of w in $\mathbf{w}$. We call $\mathbf{w}$ the *word sampling space*. By conducting a Bernoulli trial (appear or not appear) for each element in the word sampling space, we can generate a vocabulary distribution, which is a candidate topic. Fixing the number of topics (e.g., k), the task of a *topic detection method* T is to generate k topics maximizing the likelihood of the observed data.

To conduct topic evolution analysis, we divide the document corpus D into exclusive temporal subsets $D(1), ..., D(n)$ according to the timestamps of the documents such that $D = \cup_{t=1}^{n} D(t)$. Let $Z(t)$ be the k topics generated by T from $D(t)$. The problem of topic evolution analysis at time t is to analyze the relationship between the topics in $Z(t)$ and those in $Z(t-1)$.

Concretely, we need to specify the pairwise relationship between topics in $Z(t-1)$ and $Z(t)$. For two topics $z_i(t-1) \in Z(t-1)$ and $z_j(t) \in Z(t)$, we have,

$$p(\mathbf{z}_j(t)|\mathbf{z}(t-1)) \propto sim(\mathbf{z}_i(t-1), \mathbf{z}_j(t)).$$

We simply use the raw similarity rather than computing the true conditional probability. This is a design decision because given an existing topic $z_i(t-1)$, we never know the whole topic space which could evolve from it. If we simply

assume that k topics in $Z(t)$ consist of the candidate set (each has a uniform prior $1/k$), then probabilities conditioned on different previous topics are incomparable to each other. Fortunately, the raw similarity does not take any topic as the reference object and thus affords a fair measure for all pairs of topics in comparison. The raw similarity is also constrained within the unit range $[0, 1]$, making the fair comparison practical by setting some global parameters.

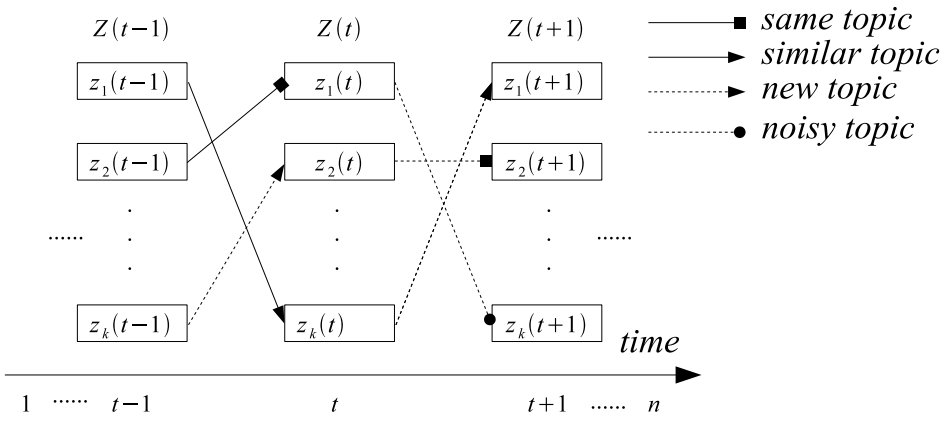

Figure 4.1. General Blogging Behavior Model (Predicting community)

Using two user-specified parameters ϵ_1 and ϵ_2 such that $1 \geq \epsilon_1 > \epsilon_2 > 1/k$, we define three types of relationships between $\mathbf{z}_i(t-1) \in Z(t-1)$ and $\mathbf{z}_j(t) \in Z(t)$:

- **Same Topic:** $z_j(t)$ and $\mathbf{z}_i(t-1)$ are very similar. Specifically, $p(\mathbf{z}_j(t)|\mathbf{z}_i(t-1)) \geq \epsilon_1$;
- **Similar Topic:** $z_j(t)$ are similar to $\mathbf{z}_i(t-1)$, that is, $\epsilon_1 > p(\mathbf{z}_j(t)|\mathbf{z}_i(t-1)) \geq \epsilon_2$;
- **New Topic:** $z_j(t)$ looks new compared to $\mathbf{z}_i(t-1)$, that is, $p(\mathbf{z}_j(t)|\mathbf{z}_i(t-1)) < \epsilon_2$.

The two threshold parameters ϵ_1 and ϵ_2 may be determined experimentally. A user may also judge whether a topic is meaningful. We thus set up the fourth type, noisy topic, which means such a topic does not correspond to any meaningful topic, i.e., it contains mainly stop words that are always present. For simplicity, in this chapter, the number of topics for each discrete time is fixed as k. It can be easily extended to any dynamic number of topics using algorithms such as Hierarchical Dirichlet Process [67].

Based on the above four types of topic evolution, we can generate a *topic evolution bipartite* over time for the whole document corpus D, as elaborated in Figure 4.1. An arc from one topic $z_i(t-1)$ to another one $\mathbf{z}_j(t)$ indicates that within $Z(t-1)$, $\mathbf{z}_i(t-1)$ has the maximum conditional probability to $\mathbf{z}_j(t)$.

4.3 Citation-Unaware/Aware Approaches

In this section, we will propose three different approaches for topic evolution from simple to complex, two citation-unaware approaches *independent topic evolution model* and *accumulative topic evolution model*, and two citation-aware approach *citation topic evolution model and inheritance topic evolution model*.

4.3.1 Citation-unaware Approaches

A simple way to discovery topic evolution is to learn topics for each independent time slot. That is to say, given the current time t, the independent topic evolution learning method detects topics only from $D(t)$. In other words, $Z(t)$ is independent from $Z(t-1)$. The learning process is defined as follows.

$$Z(t) = \arg\max_{Z(t)} \prod_{d \in D(t)} p(d|Z(t)) \tag{4.1}$$

where $p(d|Z(t))$ is the likelihood of document d given $Z(t)$ by assuming all documents in $D(t)$ are equally important for $Z(t)$.

Can we consider the dependence of the topics in $Z(t)$ on the documents at time instant t and before? The accumulative topic evolution learning method, learns the current topic space $Z(t)$ from all papers published at time t and before, i.e., from document set $\cup_{i=1}^{t} D(i)$. The learning process is

$$Z(t) = \arg\max_{Z(t)} \prod_{d \in \cup_{i=1}^{t} D(i)} p(d|Z(t)) \tag{4.2}$$

assuming all documents in $\cup_{i=1}^{t} D(i)$ are equally important for $Z(t)$.

Both methods are citation-unaware since they do not consider the citations. The independent topic evolution learning method tends to generate a large number of isolated new topics irrelevant to existing topics. In the accumulative topic evolution learning method, the existing topics tend to dominate the topic space as time goes by.

To learn topic spaces in the two citation-unaware methods, i.e., maximizing the likelihood of the data, any traditional topic models can be applied. Here, we use one of the most popular models in machine learning and information retrieval, the Latent Dirichlet Allocation (LDA) [3] framework, to generate topics. Collapsed Gibbs sampler can be used to infer the LDA posterior probabilities [68]. We denote by *i-LDA* the Gibbs sampling algorithm of independent topic evolution learning, and by *a-LDA* the Gibbs sampling algorithm of accumulative topic evolution learning.

4.3.2 Citation-aware Approach

Above two citation-unaware models have their limitations. Since topic spaces learned from *i-LDA* are independent among different time slots, those spaces tends to have larger average number of new topics and noisy topics. On the other hand, topic spaces learned from *c-LDA* are heavily dependent on historical data, those spaces tends to have larger average number of same topics and smaller average number of new topics. Is their an approach to balance same topics and new topics?

One improvement is that topic spaces $Z(t)$ are learned from both $D(t)$, and $L_{D(t)}$ the set of papers cited by papers in $D(t)$, instead of all papers $\cup_t^{i=1} D(i)$. A time t, a simple citation aware method to compute $Z(t)$ is,

$$Z(t) = \arg\max_{Z(t)} \prod_{d \in D(t) \cup L_{D(t)}} p(d|Z(t)), \tag{4.3}$$

Again, we use LDA for topic generation. We denote it as *c-LDA*.

Is *c-LDA* a good solution to identify new topics and existing topics through considering all cited papers? There are two problems. First, in *c-LDA*, all documents in $D(t) \cup L_{D(t)}$ are equally important for $Z(t)$. In the real situation, not all citations are equally important. Among all papers cited by a document d, typically only a small subset is topic-related to d. Therefore, treating all citations equally may dilute the truly important topics. Second, due to the sheer number of historical papers, some out-of-date topics may be resurrected by citations solely if the citations are not properly associated with the current topics. We call such topics *ghost topics*.

To address above two problems, we need a new topic evolution model which can balance the cited and citing papers as well as considering different weights of

cited papers on citing papers. In this purpose, we propose *Citation Inheritance Topic Model c-ITM* which use parameter λ to control the balance between cited papers and citing papers, and γ to control the weights of cited papers on citing papers. The learning process is,

$$Z(t) = \arg\max_{Z(t)} \prod_{d \in D(t) \cup L_{D(t)}} p'(d|Z(t)), \tag{4.4}$$

where

$$p'(d|Z(t)) = \lambda \cdot p(d|Z(t)) + (1-\lambda) \cdot \sum_{d_j \in L(d)} \gamma_{d_j} \cdot p'(d_j|Z(t)) \tag{4.5}$$

is the likelihood of paper d given $Z(t)$ which considers two factors: 1) topic spaces $Z(t)$ balance citing and cited papers through λ, which controls the balance between new topics and old topics, and 2) cited papers $d_j \in L(d)$ are weighted by γ_{d_j}, which controls the influence of cited papers on citing papers. Since $p'(d_j|Z(t))$ can also be learned using the same approach, Eq. 4.5 defines an interactive approach to learn topic spaces from both citing and cited papers. To the best of our knowledge, no existing topic model is able to support the iterative learning process to learn topic spaces.

How do we understand Eq. 4.5? In reality, we can think each paper d represents a distribution $\mathbf{z}_d$ on words. Thus, the distribution of $\mathbf{z}'_d$ defined by Eq. 4.5 is a linear combination of the distribution of $\mathbf{z}_d$ defined by Eq. 4.3 and the distributions of $\{\mathbf{z}'_{d_j}|d_j \in L(d)\}$ defined by Eq. 4.5.

From the prospective of probability, to generate a word w from the distribution $\mathbf{z}'_d$ first needs randomly choose the distribution $\mathbf{z}_d$ or the set of distributions $\{\mathbf{z}'_{d_j}|d_j \in L(d)\}$ according to a Bernoulli distribution $(\lambda, 1-\lambda)$. If $\mathbf{z}_d$ is selected,

we generate w from $\mathbf{z}_d$. else, if the distribution set $\mathbf{z}'_{d_j}|d_j \in L(d)$ are selected, we then randomly choose a distribution $\mathbf{z}'_{d_j}$ according a multinomial distribution $(\gamma_{d_1}, ..., \gamma_{d_j}, ..., \gamma_{d_{|L_{D(t)}|}})$. At last, we generate w from $\mathbf{z}'_{d_j}$. Hence, Eq. 4.5 can be written as,

$$
\begin{aligned}
p'(d|Z(t)) &= \prod_{w \in d} p'(w|Z(t)) \\
&= \prod_{w \in d} \int p(w|\mathbf{z}'_d) p(\mathbf{z}'_d|Z(t)) d\mathbf{z}'_d \\
&= \prod_{w \in d} (\lambda \cdot \int p(w|\mathbf{z}_d) \cdot p(\mathbf{z}_d|Z(t)) d\mathbf{z}_d \\
&\quad + (1-\lambda) \sum_{d_j \in L(d)} \int p(w|\mathbf{z}'_{d_j}) \cdot p(\mathbf{z}'_{d_j}|Z(t)) d\mathbf{z}'_{d_j})
\end{aligned}
\tag{4.6}
$$

$\int p(w|\mathbf{z}_d) \cdot p(\mathbf{z}_d|Z(t)) d\mathbf{z}_d$ can be learned as LDA, where $\mathbf{z}_d$ is a combination of a set of topics $Z(t) = \{\mathbf{z}_1, ..., \mathbf{z}_i, ..., \mathbf{z}_k\}$ through a multinomial distribution θ drawn from a Dirichlet Distribution $Dirichlet(\alpha_\theta)$. We use a random variable z to indicate which topic is chosen. We also assume $Z(t) = \{\mathbf{z}_1, ..., \mathbf{z}_i, ..., \mathbf{z}_k\}$ is drawn from a Dirichlet Distribution $Dirichlet(\alpha_\phi)$ by k times.

Furthermore, to make the process of learning parameters tractable, we assume a Bernoulli distribution $(\lambda, 1-\lambda)$ is drawn from a Beta distribution $Beta(\alpha_\lambda)$, and use a random variable s to indicate which component is chosen. We also assume a multinomial distribution $(\gamma_{d_1}, ..., \gamma_{d_j}, ..., \gamma_{d_{|L_{D(t)}|}})$ is drawn from a Dirichlet Distribution $Dirichlet(\alpha_\gamma)$, and use a random variable s to indicate which cited paper d_j is chosen.

By adding those assumptions, Eq. 4.5 can be represented as Fig 4.2.

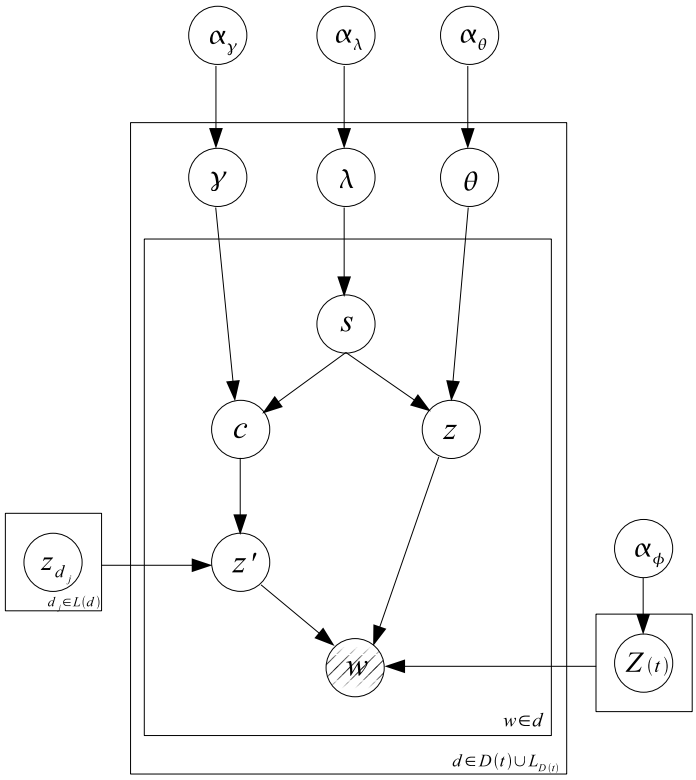

Figure 4.2. Citation Inheritance Topic Model (*c-ITM*)

4.3.3 Learning *c-ITM* using Gibbs Sampling

We use collapsed Gibbs sampler algorithm to learn parameters and variables in Fig 4.2. At each iteration of Gibbs sampling, we update the latent variables for every word position using the following processes until the latent variables converge.

$$
\begin{aligned}
p(c|d,z,s=0,data) &\propto (inhe(d,c)+\alpha_\gamma-1)\times\frac{auto(c,z)+inhe(c,z)+\alpha_\theta-1}{auto(c)+inhe(c)+k\cdot\alpha_\theta-1}\\
p(s=0|d,c,z,data) &\propto (inhe(d)+\alpha_{\lambda_0}-1)\times\frac{auto(c,z)+inhe(c,z)+\alpha_\theta-1}{auto(c)+inhe(c)+k\cdot\alpha_\theta-1}\\
p(s=1|d,c,z,data) &\propto (auto(d)+\alpha_{\lambda_1}-1)\times\frac{auto(d,z)+inhe(c=d,z)+\alpha_\theta-1}{auto(d)+inhe(c=d)+k\cdot\alpha_\theta-1}
\end{aligned}
$$

$$p(z'|w,d,c,s=0,data) \propto (auto(c,z') + inhe(c,z') + \alpha_\theta - 1) \times \frac{n(w,z') + \alpha_\phi - 1}{n(z') + V \cdot \alpha_\phi - 1}$$
$$p(c|w,d,s=1,data) \propto (auto(d,z') + inhe(c=d,z) + \alpha_\theta - 1) \times \frac{n(w,z) + \alpha_\phi - 1}{n(z) + V \cdot \alpha_\phi - 1}$$

where, $n(w,z)$ is the number of times that the word w was assigned to topic z, $n(z)$ is the number of total words assigned to topic z, $auto(d,z)$ is the number of words in the autonomous part of paper d that have topic z, $auto(d)$ is the number of words in the autonomous part of paper d, $auto(c,z)$ is the number of words in the autonomous part of citation c assigned to topic z, $auto(c)$ is the number of words in the autonomous part of citation c, $inhe(d)$ is the number of words in the inherited part of paper d, $inhe(d,c)$ is the number of words in paper d inherited from citation c, $inhe(c,z)$ is the number of words inherited from citation c and assigned to topic z over all papers that cited c, and $inhe(c)$ is the number of words inherited from citation c over all papers that cited c.

4.3.4 Motivation Matrix

One advantage of *c-ITM* can further refine the newly generated topic space by monitoring the inheritance relations among topics. For example, among the k topics in $D(t)$ produced by *c-ITM*, a few topics may not truly exist in $D(t)$ but are instead inherited from $D(t'), t' < t$ via citations. Since we sample words from the inherited and autonomous parts of a document separately, we can similarly separate the topic space $Z(t)$ into two parts: an inherited part and an autonomous part, to each of which a topic $z_j(t)$ has a certain probability.

One simple way is to use a $k \times k$ topic motivation (correlation) matrix Q for $D(t)$. Each cell Q_{ij} represents the motivation probability of topic z_i on z_j . Each

row sums to be 1. Given document d, a word w in its inherited part d^0 is assigned a topic $z_i(t) \sim Multi(\psi)$. We can assume that $z_i(t)$ motives another autonomous topic $z_l(t) \sim Multi(\theta)$ if $l = \arg\max_j p(\mathbf{z}_i(t) \rightarrow p(\mathbf{z}_j(t)))$.

The motivation probability relies on how frequently the words in d^0 and $z_i(t)$ co-occur with the words in d^1 and $z_l(t)$. As long as $l \neq i$ and $z_i(t)$ has a same topic in $Z(t-1)$, topic $z_i(t)$ can be regarded as an inherited topic that is no longer hot in the current topic space. This is reasonable because if $z_i(t)$ were popular at time t, there should have been many papers in topic $z_i(t)$ that cite papers from the same topic, so that the motivation probability to itself at time t is still significant. Ideally, diagonal probabilities should dominate the motivation matrix for the topic evolution category of "same topic".

If we consider $z_j(t)$ as a word instead of an indicator of topics, we can learn the motivation matrix as same as the learning process in LDA.

4.3.5 Complexity

The time complexity of the four algorithms fully depends on the efficiency of the k-topics. Let N be the dimensionality of the word sampling space w. There are a total of kN parameters to infer during each iteration for the LDA model under Gibbs sampling. Let n be the number of iterations, the time complexity for *i-LDA*, *a-LDA* and *c-LDA* is $O(nkN)$. For *c-ITM*, there are $kN + 2N + \overline{|L_d|} \times N$ parameters to be inferred in each duration, where the dimensionality of the indicator vector s is 2 and $\overline{|L_d|}$ is the average number of citations of each paper in the dataset (i.e., the average dimensionality of the dummy index vector $\mathbf{c}$). Since $\overline{|L_d|}$ is a constant given a document, the k-topic *c-ITM* model does not grow with the size of the data. The time complexity of *c-ITM* is $O(n(k + 2 + \overline{|L_d|} \times N)$.

Since both LDA and c-ITM can be convergent after a limited number of iterations, given n, all four algorithms thus have a linear scalability with respect to N, the only factor that solely relies on the size of data.

4.4 Empirical Evaluation

4.4.1 Dataset

We tested topic evolution models on the literature archived at CiteSeerX. The dataset contains research papers in computer and information science. We selected papers published in the last 16 years $(1993 - 2008)$. After removing duplicate papers, papers without explicit publication timestamps, we obtained $650,918$ unique papers dated until early 2008. For each paper, we extracted its title and abstract as content, ignoring the rest. We used a year as the time unit in our analysis. The set of papers published in year t $(1993 \leq t \geq 2008)$ is fed into i-LDA to learn the topic space of the year. a-LDA uses all papers published in or before year t to learn the topic space of the year. For both c-LDA and c-ITM, we extract all cited papers prior to each year. For simplicity, only 1-hop citations are considered. Please note that only those cited papers in the dataset are used by c-LDA and c-ITM.

For the LDA model, we used the free Mallet tool [2]. We implemented our ITM model in $C++$. For the hyper parameter settings, $\alpha_\theta = 0.1$, $\alpha_\phi = 0.01$, $\alpha_\gamma = 1.0$, $\alpha_{\lambda_0} = 0.1$, $\alpha_{\lambda_0} = 0.1$ and $\alpha_\theta = 0.1$. All these hyper parameter settings simply follow the tradition of topic modeling [3]. All experiments were conducted on a Linux server with 7 CPU processors of 2.4GHz and $16G$ memory.

[2] http://mallet.cs.umass.edu

4.4.2 Evaluation on Topic Evolution Categorization

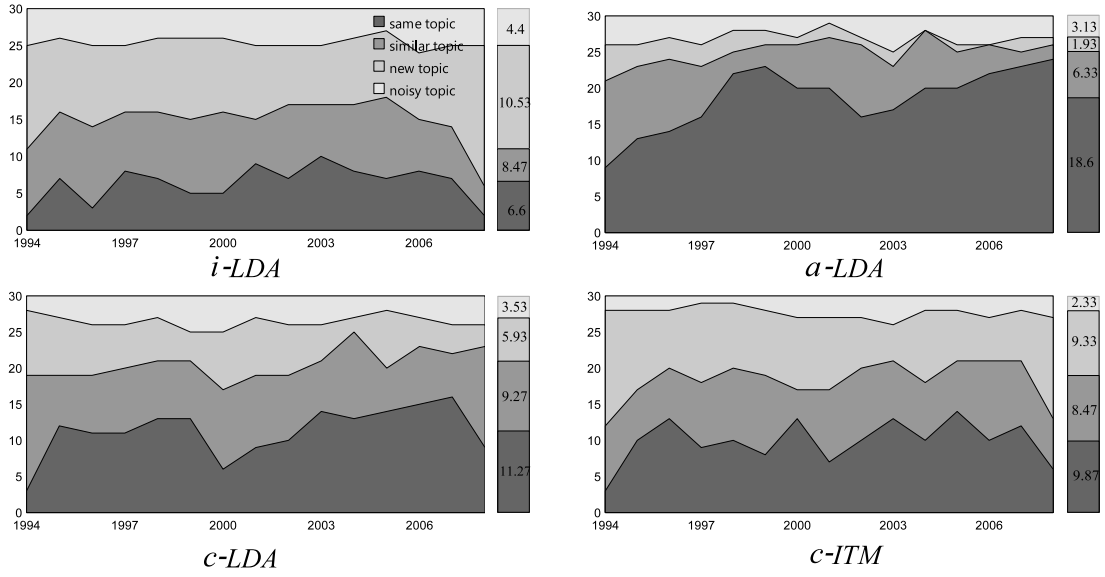

Figure 4.3. Categorization of 15-year topic evolution. The right bar of each sub-figure shows the average number of topics fallen into the according category.

We extracted the top 30 topics, and set the parameters $\epsilon_1 = 0.5$ and $\epsilon_2 = 0.2$. Figures 4.3 show the distribution of the different types of topic evolution found by *i-LDA*, *a-LDA*, *c-LDA*, and *c-ITM*, respectively. We can obtain some interesting observations. *i-LDA* tends to produce the largest average number of new topics (10.53) and noisy topics (4.4). On average, almost half of the topics (14.93) generated by i-LDA are either new or noisy. As the data volume in years $1998 - 2005$ increases, *i-LDA* shares more topics from the topic spaces in the previous years. However, as the data in year 2008 is incomplete (papers crawled after early 2008 are not included) and thus much smaller (less than 1/10) compared to the other years, almost all generated topics are either new or noisy.

a-LDA is on the other end of the extreme: historical topics tend to dominate the topic space every year. For example, after year 1999, as the accumulation of

historical data, the topic space of the current year is almost completely dominated by the previous topic space (on average, 2/3 generated topics are same topics). In contrast to *i-LDA*, a smaller data volume (of the current year) results in fewer new topics in *a-LDA*. *a-LDA* generates noisy topics without a clear trend: on one hand, the dominance of historical topics can eliminate noisy topics; on another hand, the noisy words contributed to noisy topics are also accumulated along the time. The two citation-unaware methods suffer from either heavy topic drifting or heavy topic inheritance, both are undesirable for topic evolution. Moreover, both methods are very sensitive to changes in data size.

Compared to *i-LDA*, *a-LDA* and *c-LDA*, *c-ITM* produces the fewest noisy topics (2.33 on average) among all models. Not all topics generated by *LDA* are interpretable, and a few noisy topics exist. *i-LDA* is just as the standard *LDA*. *a-LDA* and *c-LDA* incorporate all topics including noisy topics to generated topic spaces. So that they still generate a larger number of noisy topics (3.13 and 3.53 on average). On the other hand, *c-ITM* assigns higher wights to topics in cited papers which have influence on topics in citing papers. Those influenced topics are not noisy topics, and then non-noisy topics are boosted in *c-ITM*. Hence, *c-ITM* generates the fewest noisy topics among all models.

The topic similarity trends tell the differences among our four topic evolution methods: *i-LDA* always has the smallest topic similarities so that topics oscillate the most. *a-LDA* always has the highest topic similarities so that topics tend to retain. *c-LDA* and *c-ITM* stay in the middle yet sometimes *c-ITM* has a bit smaller topic similarities, so that *c-ITM* can generate a bit more new topics. Last, when the data volume increases in some year, the differences among the four methods become smaller.

4.4.3 Filtering Ghost Topics

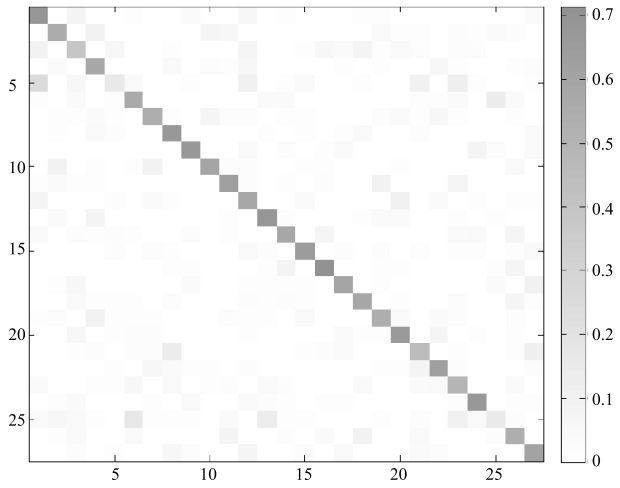

Figure 4.4. Topic×Topic motivation matrix in 2006.

Table 4.1. Top topic motivation probabilities for ghost topics in 2006

cited topics	citing topics	probability
topic5: mining patterns	topic1: clustering similarity	0.2455
	self	0.1657
	topic23: graph algorithms	0.1173
	topic21: logic fuzzy	0.1115
	topic12: streams information	0.1026
topic25: coding compression	topic6: quantum complexity	0.1685
	self	0.1520
	topic13: memory cache	0.1319
	topic23: graph algorithms	0.109

Although *c-ITM* strikes a good balance between new topics and same topics, some old topics that have been declining in the current year still may be inherited along the citations. We can optionally build the topic motivation matrix to filter the topic space produced by *c-ITM*. We used year 2006 as an example to generate the topic motivation matrix (27×27 after removing 3 noisy topics), as shown in Figure 4.4.

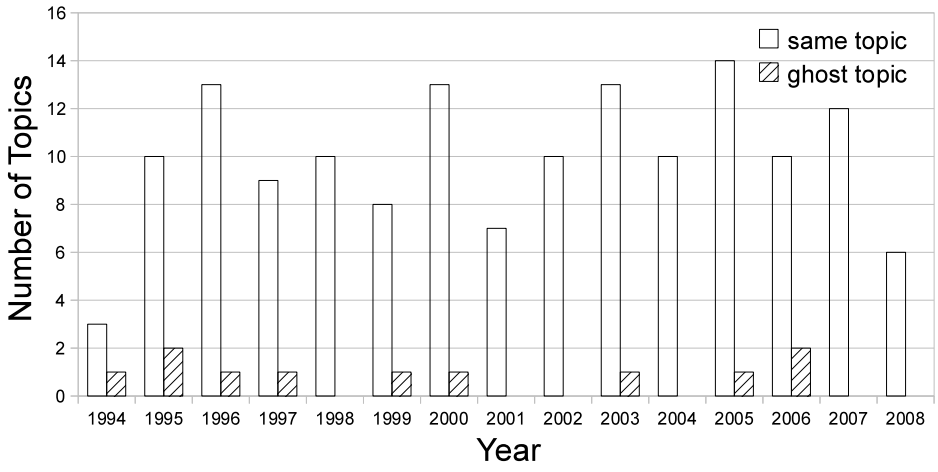

Figure 4.5. Distribution of detected ghost topics.

Among the 10 same topics, we identified two ghost topics (topics 5 and 25) that exist in the previous year topic space, and have high motivation probability to the other topics, but low motivation probabilities to themselves. These ghost topics exist only in the cited papers. Table 4.1 shows in detail how the two ghost topics were cited by other valid 2006 topics. Limited by space, only the top 2 words were used to represent each topic without manual labels.

Figure 4.5 shows the number of ghost topics yearly. Compared to the number of *same topics*, only a small portion of *same topics* are ghost topics (not hot any more in the current year). Clearly, other three models cannot do that.

4.4.4 Topic Evolution Case Study

To better understand the topic evolution process, here we present some real topic evolution examples related to the category of *image processing*, as shown in Figure 4.6. Each topic is described using the top 5 words without any human labels.

There are two main topics related to image processing: *image compression*

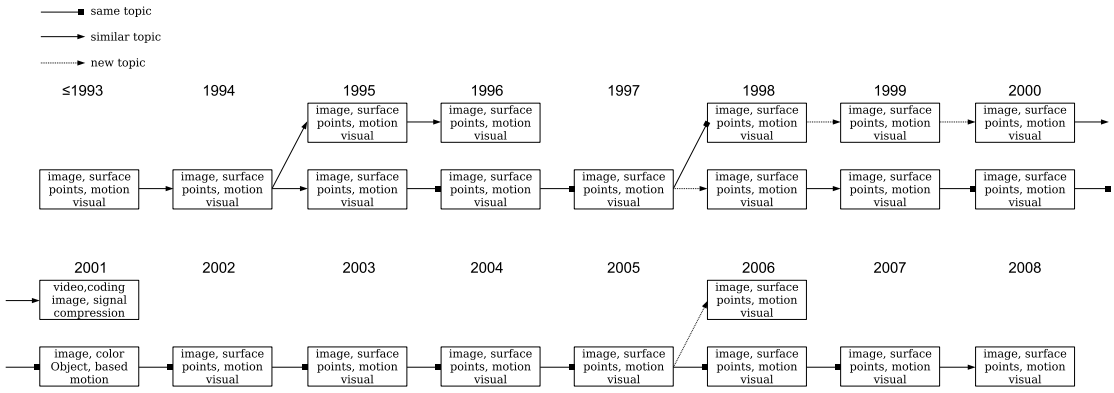

Figure 4.6. Topic×Topic motivation matrix in 2006.

which mainly evolved from 1994 to 2001, and *face recognition* which mainly evolved from 1998 to 2007. Specifically, *image compression* evolved from the topic *image surface* in 1994 and subsequently evolved into the new topic *face recognition* in 1998. In 1998, *image compression* further evolved from *static image compression* to *video compression*. Except for the year 1999 in which *video compression* was suddenly interrupted by channel coding which was still hot for *image compression*, we believe the other evolutions are consistent and reasonable. We can also conclude that wavelet coding is a very important tool for both *static image compression* and *video compression*, rather than other like *channel coding*.

4.4.5 Scalability and Time Efficiency

We analyze the scalability and time efficiency of our topic evolution methods. The scalability of LDA has been tested in many previous work. Here, we only test the *c-ITM* model. The total number of word occurrences N across 16 years in our dataset is $42,389,066$. Accordingly, we sampled 10%, 20%, . . . , 100% of the word occurrences to test the scalability. Note that except for *a-LDA*, we will never have a chance to use 100% of all word occurrences. We showed that the time

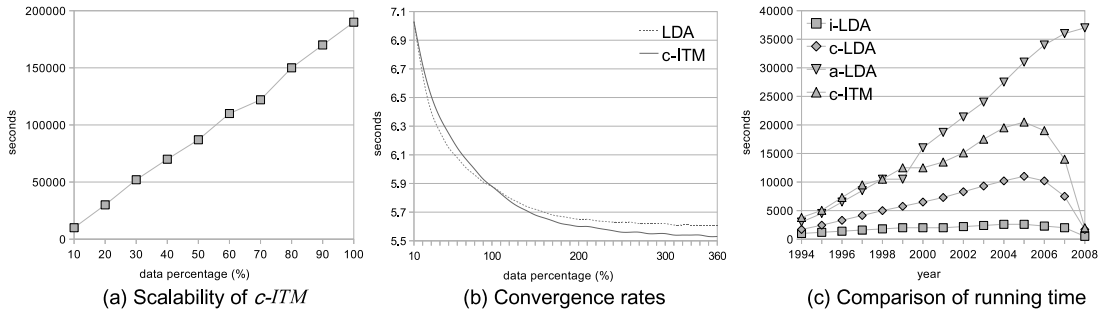

Figure 4.7. Scalability and Time Efficiency.

complexity of *c-ITM* is linear with respect to the number of word occurrences, the number of iterations, and the size of topic space. Figure 4.7(a) further verifies our claim with $k = 30$ and $1,000$ Gibbs sampling iterations.

Previous work [68] reported that LDA under Gibbs sampling normally requires around 500 to 1000 iterations to reach convergence. Here, we also compared the convergence rate for two topic models: *c-ITM* and LDA under Gibbs sampling. We used the minus likelihood of the data to measure how our model fits the data (the whole word space $\mathbf{w}$), which is defined as $p(\mathbf{w}) = -\frac{1}{N}\sum_{w_i \in \mathbf{w}} \log p(w_i)$. Suppose that d is the document from which the word w_i originates, $p(w_i) = \sum_{zinZ} \phi_z(w_i)[\theta_d(z)p(s=1) + \sum_{d' \in L(d)} \theta_{d'}(z)\gamma_{d'}p(s=0)]$.

Figure 4.7(b) shows the convergence rate of models. After about 100 iterations, the likelihood of the data stabilizes and does not change significantly for both models. Overall, LDA converges a bit faster and stabilizes after 200 iterations. Instead, *c-ITM* cannot improve the likelihood further after 300 iterations. But after convergence, *c-ITM* has a higher likelihood. The result indicates that the convergence speed of our model is comparable to LDA under the Gibbs sampling; and our model fits the citation graph data better.

Lastly, we tested the running time of all topic evolution methods with $k = 30$,

as shown in Figure 4.7(c). For a fair comparison, we ran $1,000$ iterations for each method. The running time of all methods grows/declines linearly as the data volume and the word sampling space increase/decrease. Under the same data distribution, *c-ITM* is slower than *c-LDA* as the former needs to infer 2 additional latent variables.

4.5 Conclusion

In this chapter, we studied the topic evolution problem for scaled scientific literature. We first investigated the citation-unaware approaches based on the LDA model, along with their limitations on topic evolution, i.e., the correlated topics were generated independently. We then proposed the citation-aware approaches for topic evolution. Moreover, an iterative topic learning framework based on citation network was presented to fully utilize the impact of citations. A novel Inheritance Topic Model was then naturally proposed for this learning process. Our algorithm can be quickly convergent under the Gibbs sampling and has a linear scalability with respect to the size of dataset. The experimental results show that our approach can track the topic evolution in a large dataset containing more than 650,000 papers over 16 years. The experimental results clearly indicate that citations are able to portray the inherent dependence among correlated topics, and citation-aware approaches are thus good choices for tackling the sequential topic evolution problem.

Chapter 5

Aspect-level Sentiment Analysis: Incorporate Lexicons

5.1 Overview

This chapter focuses on incorporating lexicons to improve the precision of sentiment classification. We give the problem definition at first. Then we present a new method to generate domain specific lexicons, and describes a new approach to incorporating generated lexicons with the SVM algorithm. Finally, our approach is evaluated on real camera reviews.

5.2 Problem Definition

Suppose we have a sentence set $\mathbb{D} = \{D_{train}, D_{test}\}$ and an sentiment set $\mathbb{O} = \{positive, negative, none\}$ (*none* means no sentiment included in a sentence.). $D_{train} = \{s_1^l, \cdots, s_i^l, \cdots\}$ is the sentence set for training, where each sentence s_i^l is labeled with one of sentiment (*positive*, *negative* or *none*) from $\mathbb{O}$. $D_{test} =$

$\{s_1^u, \cdots, s_i^u, \cdots\}$ is the unlabeled sentence set. The goal is to classify each unlabeled sentence s_i^l as one of sentiment (*positive*, *negative* or *none*). This chapter focuses on how to find useful features from sentences for supervised learning to achieve a higher precision on sentiment prediction. The problems come from two aspects:

1. Which words are useful indicators for supervised learning?

2. How do we incorporate those useful words with supervised learning process?

Followings, we will present how to solve above two problems step by step.

5.3 Generating Domain Specific Lexicons

This section describes our approach to generating domain specific lexicons. We use the area of digital cameras as an example to illustrate our approach. However, our method is applicable to other areas as well.

As discussed above, the sentiments of many words or phrases are context or domain dependent. For example, *long* is positive if it is associated with the camera aspect of 'Battery Life'. However, the same word carries negative sentiment when it is associated with the camera aspect of 'Shutter Lag'. Therefore, it is critical to know the topic/domain being discussed when we try to determine the associated sentiment.

Based on this observation, we aim to build domain/topic specific lexicons covering both expressions indicating a specific domain and expressions indicating different sentiments associated with that particular domain. For example, our lexicon

regarding 'Camera Picture Quality' would consist of two sub-lexicons. One includes words and phrases such as *picture, image, photo, close up* etc, which are good indicators for the topic of 'Picture Quality' in the area of digital cameras. The other one includes words and expressions that carry positive or negative sentiments if the associated topic is camera picture quality. For example, this second sub-lexicon would indicate that while *sharp* and *clear* are positive, *blurry* is negative when they are associated with camera picture quality. We achieved our goal by using a combination of corpus filtering, web search with linguistic patterns and dictionary expansion. Each of these techniques are described in detail in the following subsections.

5.3.1 Corpus Filtering

Since we have a training corpus, in which each camera review sentence is annotated with a camera aspect as well as the associated sentiment being expressed in that sentence, it is straightforward to use this resource to build a foundation for our domain specific lexicons. Our approach is as follows.

First, for each camera aspects such as *Durability*, we extract all of the content words and phrases occurred in the training sentences labelled as expressing that aspect. The content words and phrases we extracted include nouns, verbs, adjectives, adverbs as well as their negated forms. This step produces an initial list of lexicon for each camera aspect.

Second, for each word and phrase in the list for each of the camera aspects, we check to see if that word or phrase also occurs in any other camera aspect lexicon. If yes, we remove it from the lexicon. After this step of filtering, we obtained a list of lexicon for each camera aspect, which contains only words and phrases unique

to that camera aspect in our training corpus.

The quality of the lexicons produced using this approach is in general very high. For example, the following lexicon regarding the camera *Durability* was generated based on our relatively small training corpus with 2131 sentences covering 22 camera aspects and a category of none of the 22 camera aspects was discussed.

Durability Lexicon: *[scratch, construct, build, rock, repair, damage, flimsy, not flimsy, junk, sturdy, sturdier, solid, durable, tough, bent, hard, not worth, firm, rug, broke, bulletproof]*

However, the drawback of this approach is that the coverage of the lexicons would completely rely on the coverage of the corpus, and annotating broad coverage training corpus is time consuming, expensive and sometimes very difficult for a task such as sentiment analysis because of the richness of natural language.

We overcome this drawback by augmenting the initial domain specific lexicons we obtained from the training corpus through web search and filtering using linguistic patterns as well as dictionary expansion. These two approaches are illustrated in the next two subsections.

5.3.2 Web Search and Filtering Using Linguistic Patterns

To improve the coverage of the domain specific lexicons we obtained from our training corpus, we designed two linguistic patterns and use them as searching queries to find more words and phrases conceptually associated with the camera aspects. The two linguistic patterns we used are as follows.

Pattern 1: "Camera Aspect include(s) *" Pattern 2: Camera Aspect + "Seed Word and *"

In these two patterns, 'Camera Aspect' refers to expressions such as *camera*

accessories and *camera price*. 'Seed Word' refers to seed words for a particular camera aspect. For example, *cheap* and *expensive* can serve as seed words for camera aspect *price*. Note that in Pattern 1, the camera aspect name is included as part of an exact search query, whereas in Pattern 2, the camera aspect name serves as the context for the search query.

Depends on the semantic nature of a camera aspect, we choose one of these two patterns to find expressions conceptually related to that aspect. For example, while "camera accessories include *" is very effective for finding accessory expressions, 'camera picture + "clear and *"' is better for finding expressions related to camera pictures.

When we use Pattern 1, we send it as a query to a search engine such as Bing[1]. We then extract words following 'include' or 'includes' in the top 50 results returned by the search engine. In each returned result, we extract words following 'include' or 'includes' until we hit the sentence boundary. The final step is to remove common stop words such as *the* and function words such as *with* and *of* from the extracted words. As an example, the following lexicon for camera accessory is generated using this method.

Accessory Lexicon: *[chip, chips, case, bag, card, software, tripod, strap, cable, adapt, charger, port, storage, hood, connector, kit, accessory, glove, belt, usb, mic, beltloop, flash, program, leather, pack, connect, not belt, not strap, zipper]*

When we use Pattern 2, we also extract words in the top 50 returned results. However, we adopt a different algorithm for filtering out noises in the returned results. For example, for finding expressions conceptually related to camera's picture quality, we use 'camera picture' as context words and 'clear' as a seed word.

[1]In our experiments, we used Bing for convenience. However, our approach is applicable using other search engines such as Google as well.

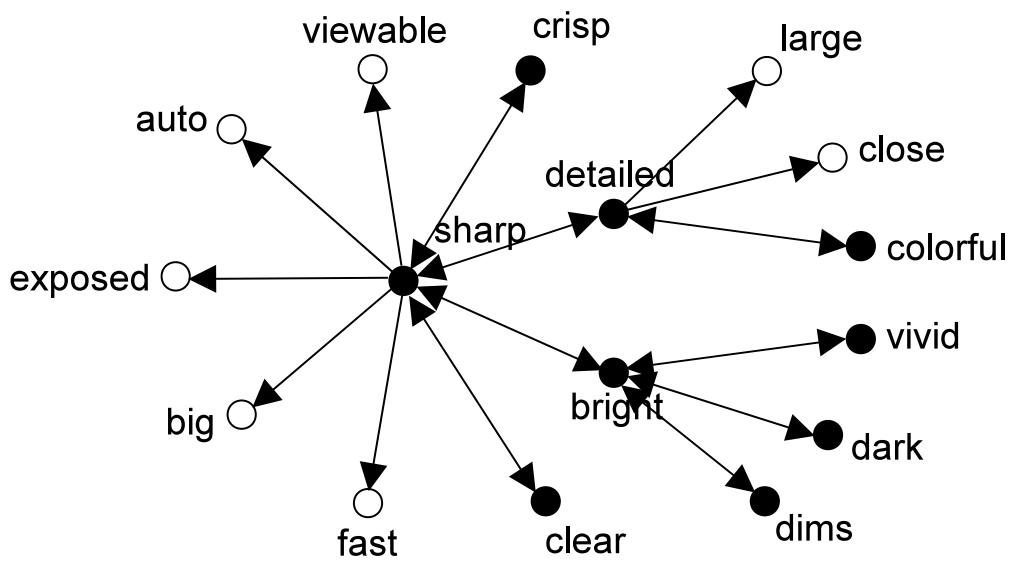

Figure 5.1. Noisy Words v.s. Non-noisy Words for Camera Picture Quality

This pattern would match both 'clear and sharp' and 'clear and normal'. However, while 'sharp' is commonly used to describe picture quality, 'normal' is not. To filter noisy words such as 'normal', we use each of the candidate word as a new seed word in Pattern 2, and if the top 50 results returned by the new query include the original seed word 'clear', the candidate word is retained. Otherwise, it is discarded. For example, in our experiments, while 'camera picture + "sharp and *"' would return results matching 'sharp and clear', 'camera picture + "normal and *"' would not return results matching 'normal and clear'. Through this way, we can distinguish 'sharp' from 'normal', and identify 'normal' as a noisy word. Figure 1 shows some of the noisy words identified by this approach when we extract expressions conceptually related to camera pictures. In this figure, words represented by hollow circles are identified as noises and removed from the camera picture quality lexicon. By contrast, words represented by solid circles are retained in our lexicon.

The algorithms we adopted for constructing domain specific lexicons using Pattern 2 are summarized below as 'FindingRelatedWords', which in turn uses algorithms 'HarvestByBing' and 'isReversable'.

Algorithm: FindingRelatedWords
Input: seedword, contextword, depth
Output: relatedwordset
unprocessed = [seedword] ;
relatedwords = [seedword] ;
foreach *Depth* **in** [1...*N*] **do**
  tempset = [ ] ;
  foreach *word* **in** *unprocessed* **do**
    newwords = HarvestByBing(word, contextword) ;
    foreach *newword* **in** *newwords* **do**
      if *isReversable(word, newword, contextword)* **then**
        Add newword to tempset ;
  foreach *newword* **in** *tempset* **do**
    Add newword to relatedwords
  unprocessed = tempset ;
return relatedwords

Algorithm: HarvestByBing
Input: word, contextword
Output: newwords
LPattern = contextword + "word and *" ;
newwords = *words matchig * in texts of top 50 results returned from Bing using LPattern as a query* ;
return newwords

Using this method, we build the following lexicon for camera picture quality by using Pattern 2 as search queries with two seed words 'clear' and 'blurred'.

PictureQuality Lexicon: *[clear, sharp, color, bright, kyocera, response, sober, stable, tidy, vivid, disassemble, detail, texture, safe, fluid, dark, sunny, dim, crisp, focus, pattern, curve, blue, humid, fuzzy, orange, yellow, gray, blurry, blur, cyan,*

Algorithm:isReversable
Input: word, newword, contextword
Output: True or False
newwords = HarvestThroughBing(newword, contextword) ;
if *word* **in** *newwords* **then**
| **return** True
else
| **return** False

indistinct, grainy, hazy, blurred]

5.3.3 Dictionary Expansion

Although expansion through looking up synonyms and antonyms recorded in dictionaries is a commonly used approach when a general purpose sentiment lexicon is built [49], we found this approach not always suitable for building domain specific lexicons. The reason is that building domain specific lexicons requires finding expressions that are conceptually related, but expressions that are conceptually related are not necessarily synonyms or antonyms. For example, 'sharp' and 'clear' are conceptually related to camera picture qualities, but they are not really synonyms from a linguistic perspective.

However, in some cases, using dictionaries can still be very effective. For example, we built the following lexicon for camera price through web searching and filtering using Pattern 2.

Price Lexicon: *[cheap, lowest, discount, promo, coupon, promote, expensive, worthy, value]*

By including the synonyms of 'cheap' and 'expensive' in WordNet [61] as shown below, we are able to further expand the Price Lexicon.

Synonyms of 'expensive' in WordNet: *[expensive, big-ticket, high-ticket, dear, high-priced, pricey, pricy, dearly-won, costly, overpriced]*

Synonyms of 'cheap' in WordNet: *[cheap, inexpensive, bargain-priced, cut-rate, cut-price, catchpenny, dirt cheap, low-budget, low-cost, low-priced, affordable, dime, penny, halfpenny]*

5.3.4 Domain Specific Polarity Lexicon

So far we have described how we build domain specific lexicons for different camera aspects, and the next step is to separate expressions that carry positive sentiment from those that carry negative sentiment in each domain lexicon.

For example, we want to be able to build the following sub-lexicons for 'Picture Quality'.

PictureQuality Positive Lexicon: *[clear, sharp, bright, sober, stable, tidy, vivid, sunny, crisp]*

PictureQuality Negative Lexicon: *[dark, dim, humid, fuzzy, gray, blurry, blur, indistinct, grainy, hazy, blurred]*

Our approach is as follows. for each expression in the Picture Quality Lexicon we constructed through the combination of corpus filtering, web search and dictionary expansion, we check to see if it only appears in the training data labelled as expressing a positive opinion or a negative opinion about the camera's picture quality. If it is the former case, we include that expression into the PictureQuality Positive Lexicon, and if it is the latter case, we include that expression into the PictureQuality Negative Lexicon.

Having illustrated our approach for constructing domain specific sentiment lexicons, we describe how we incorporate lexicon knowledge into SVM learning to

improve sentiment classification in the next section.

5.4 Incorporating Lexicon Knowledge into SVM Learning to Improve Sentiment Classification

Our sentiment classification task is as follows. For each review sentence about cameras, we need to predict both the camera aspect discussed in that sentence as well as the associated sentiment regarding that camera aspect. For example, for the following review sentence,

(1) *It uses 2 batteries and the batteries last longer than my last camera lasted using 4.*

We want to be able to identify that this sentence expresses a positive opinion about the battery life of the camera.

We achieve this goal by performing a two step classification. In step 1, we train a classifier to predict the camera aspect being discussed. In step 2, we train a classifier to predict the sentiment associated with that camera aspect. Finally we aggregate the two step prediction results together to produce the final prediction.

In both steps, we incorporate the lexicon knowledge into conventional SVM learning. To illustrate our approach, we use sentence (2) as an example.

(2) *The case is rigid so it gives the camera extra nice protection.*

Using nouns, verbs, adjectives and adverbs as feature words in a conventional SVM learning, this sentence can be represented as the following vector of words.

[case, rigid, give, camera, extra, nice, protection]

By incorporating the knowledge encoded in the lexicons, we automatically generate and insert additional features into the above representation.

For example, when we perform the step 1 aspect classification, because the feature word 'case' in the above representation is listed in our domain specific lexicon about camera accessories, we would insert an additional feature word 'accessory', and produce the following new representation.

[case, rigid, give, camera, extra, nice, protection, accessory]

By doing this, we promote the possibility of the camera aspect being 'accessory' if expressions of camera aspects occur in the sentence.

In the next step of polarity prediction, we incorporate both of our domain specific sentiment lexicon and a general purpose domain independent sentiment lexicon extracted from the MPQA opinion corpus [30] [2].

For example, because 'nice' is indicated as a positive word in the MPQA lexicon, we would insert a feature word 'positive'. In addition, if the first step prediction result for sentence (2) is 'accessory', and 'rigid' is also a positive word in our domain specific lexicon regarding camera accessories, we would generate an extra feature word 'positive' in our final representation for sentence (2) for the second step polarity prediction as shown below.

[case, rigid, give, camera, extra, nice, protection, positive, positive]

By doing this, we promote a 'positive' prediction regarding the aspect of 'accessory'.

Our experiments show that incorporating lexicon knowledge into SVM learning significantly improves the accuracy for our classification task, and compared to the general purpose MPQA sentiment lexicon, the domain specific lexicon we

[2]We only extracted the words that are indicated as strongly subjective out of context from the MPQA opinion corpus

constructed is more effective. Our experiment setting and results are reported in the next section.

5.5 Experiment Setting and Results

For the experiment, we randomly selected 6100 sentences in total from multi-domain sentiment dataset created by Blitzer at al. [69]. We use crowd-sourcing techniques to obtain training dataset from untrained non-expert workers such as the ones on the Amazon Mechanical Turk (ATM) platform [3]. For each sentence, we ask the workers to judge whether a sentence indicates an opinion towards a certain aspect of the camera, and if so, whether the opinion is positive, negative or neutral. We design the following interface for ATM workers to annotate each sentence. For example, Fig 5.2 shows the interface for the sentence

(3) *"On my trip to California, the camera fell and broke into two pieces."*.

Feature Name	Not Invoked	Positive	Negative	Neutral
Construction Quality	○	○	◉	○
Picture Quality	◉	○	○	○
Battery Life	◉	○	○	○
...				

Figure 5.2. Interface for annotation by ATM workers.

Concerning the quality of annotations by workers, we ask two ATM workers to annotate each sentence independently. We believe that it is very unlikely for two reliable ATM workers to annotate any given sentence exactly the same way merely by chance. Therefore, we consider an annotation to be gold when both

[3]This is an online market place that offers a small amount of money to people who perform some "Human Intelligence Tasks". http://www.mturk.com/mturk

annotators marked the same sentiment toward the same aspect. We obtained 2718 gold-standard annotations from the ATM workers. We use 2131 sentences in total for training and 587 sentences for hold-out testing. There are total 23 classes, consisted of 22 aspects [4] as well as a class of *no opinion* about any of the 22 aspects. Fig 5.3 shows the histogram of our dataset.

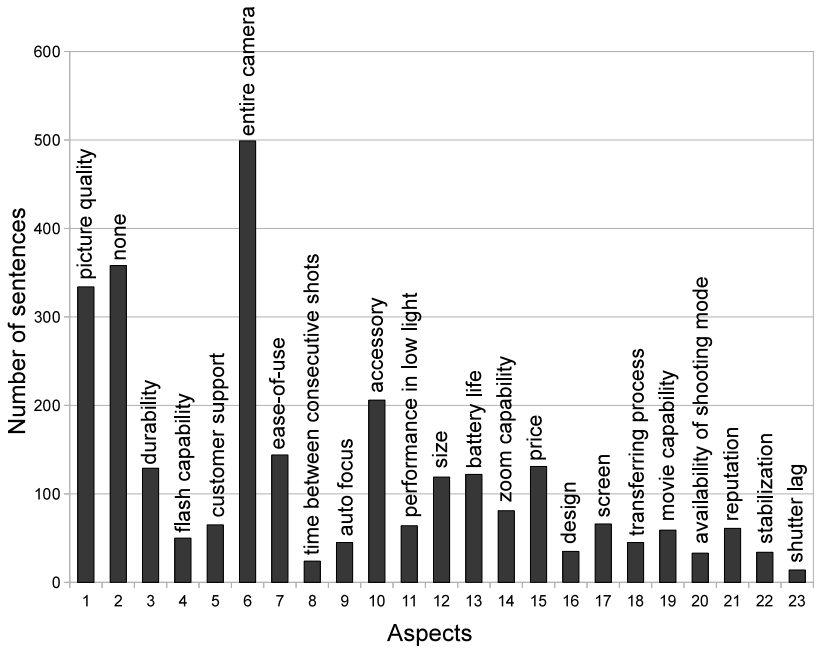

Figure 5.3. Histogram of dataset.

As mentioned in the previous section, we performed a two step classification for our task. Namely, our final combined classifier consists of two classifier. The first is an 'Aspect Classifier', which performs a 23 way camera aspect classification. The second is a 'Polarity Classifier', which performs a 3 way (*positive*, *negative* and *none*) classification. The final predictions are aggregated from the predictions produced by these two classifiers.

[4]

The classification accuracy is defined as follows.

$$Accuracy = \frac{NumberofSentencesCorrectlyClassified}{TotalNumberofSentences}. \quad (5.1)$$

We compared our approach to incorporating lexicon knowledge with SVM learning with a conventional SVM learning, because the latter is the state-of-the-art algorithm reported in the literature for sentiment analysis.

We selected the Nouns, Verbs, Adjectives and Adverbs as our unigram word features from the sentences for training and testing. All of them are stemmed using the Porter Stemmer [70]. Negators are attached to the next selected feature word. We also use a small set of stop words[5] to exclude copulas and words such as *take*. The reason that we choose these words as stop words is because they are both frequent and ambiguous and thus tend to have a negative impact on the classifier. The SVM algorithm we adopted is implemented by Chang et al. [71]. We use linear kernel type and use the default setting for all other parameters.

We conducted 4 experiments. In experiment 1, we used the conventional SVM algorithm, in which no lexicon knowledge was incorporated, and we refer to this experiment as SVM. In experiment 2, we incorporated only the knowledge encoded in the domain independent MPQA opinion dictionary into SVM learning, and we refer to this experiment as''MPQA + SVM'. In experiment 3, we incorporated only the knowledge encoded in the domain specific lexicons we constructed into SVM learning, and we refer to this experiment as 'DomainLexicons + SVM'. In experiment 4, we incorporated both the knowledge encoded in the MPQA and the domain specific lexicons we constructed into SVM learning, and we refer to this

[5]The stop words we use include copulas and the following words: *take, takes, make, makes, just, still, even, too, much, enough, back, again, far, same*

experiment as 'DomainLexicons + MPQA + SVM'.

Our experiment results show that incorporating both the domain independent MPQA lexicon and the domain specific lexicons we built achieves the best overall performance. However, compared to the domain independent general purpose MPQA lexicon, the domain specific lexicons are more effective, and they contributed the most to the improvement of the classification accuracy. Our experiment results are summarized in Table 1.

Learning Method	Accuracy
SVM	41.7%
MPQA + SVM	44.3%
DomainLexicons + SVM	46.2%
DomainLexicons + MPQA + SVM	**47.4%**

Table 5.1. Overall Performance Comparison

Our results reported in Table 2 further illustrate that incorporating lexicon knowledge with SVM learning significantly improves both the accuracy for camera aspect classification and the accuracy for polarity classification.

Learning Method	Aspect Accuracy	Polarity Accuracy
SVM	47.1%	65.6%
DomainLexicons + MPQA + SVM	**56.2%**	**66.8%**

Table 5.2. Breakdown Performance Comparison

Since there are 23 classes of aspects, we use confusion matrix to explain the improvement of our algorithm in more detail. See Fig 5.4, when domain specific lexicons are used, classifications on aspects of *accessory*, *battery life*, *durability*, *picture quality* and *prices* have been improved significantly. Human can easily identify what is the aspect of a sentence through one or two indicator words. Taking the sentence (1) in section 5.4 as an example, human can easily classify (1) as an aspect of *bettery life* just based on words "*battery*" and "*life*". However,

SVM learning methods cannot acquire such indicator words based on training data. Hence, we use domain specific lexicons to tell SVM which words are more important for aspect classification, and achieve significant improvement.

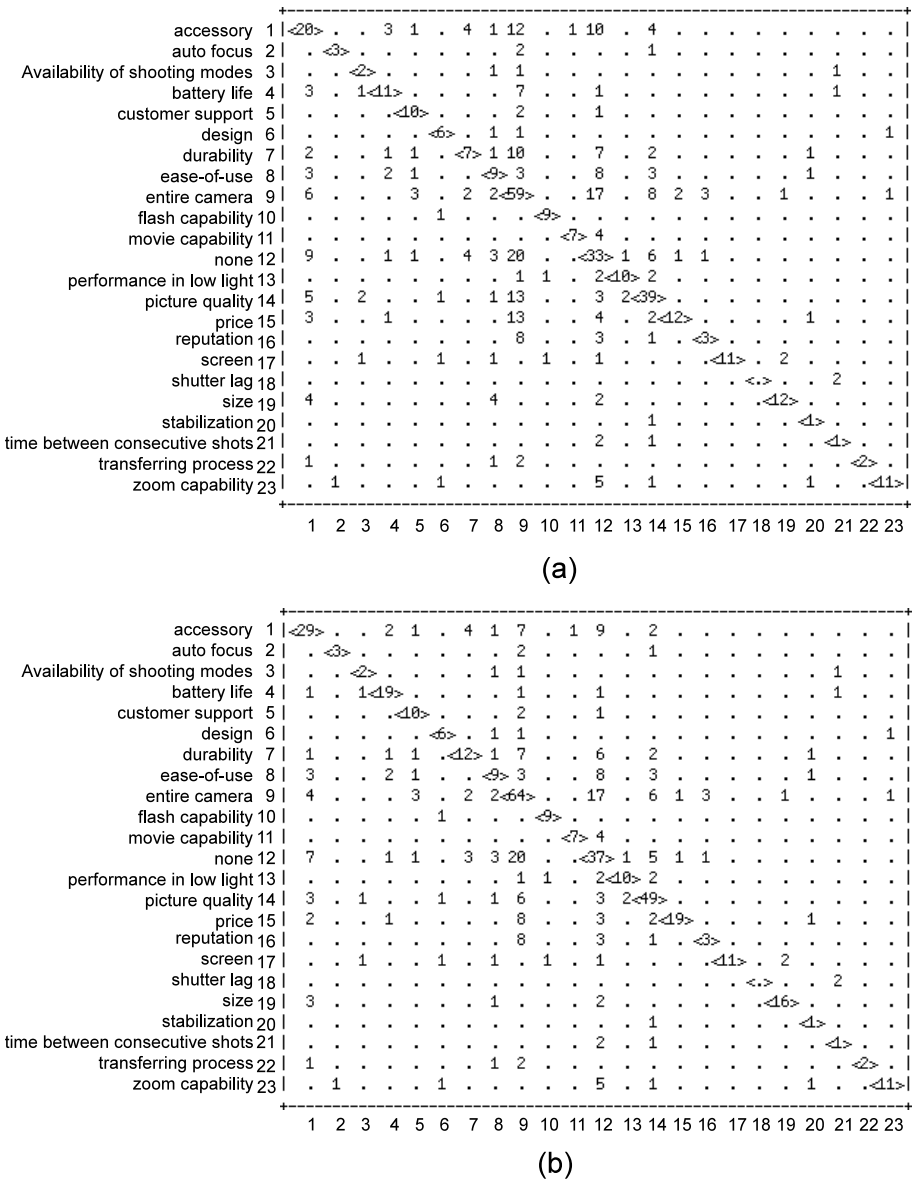

Figure 5.4. Confusion matrix (a) using SVM as learning method. (b) using Domain-Lexicons+MPQA+SVM as learning method.

On the other hand, we still can see that accuracy on some aspects is not im-

proved after using domain specific lexicons. There are two reasons: 1) for some aspects, it is hard for us to build lexicons such as aspects of *none* and *reputation*; 2) for some aspects, indicator words are so obvious that have been learned by SVM algorithm such as aspect of *screen* and *shutter lag*. Overall, building domain specific lexicons is an effective way to improve the accuracy of classification using supervised learning methods such as SVM.

5.6 Why Our Approach Works

Section 5 provides empirical evidence that incorporating lexicon knowledge into SVM learning improves the accuracy of sentiment classification. This section offers a theoretical proof on why this is true.

In the case of support vector machines, a data point is viewed as a p-dimension vector, and the strategy of SVM is to find a $(p-1)$-dimension hyperplane, which yields the largest separation, or margin between any two classes. The larger the margin between two classes is, the more separable these two classes are. The reason why our method can improve the accuracy of classification is because the extra features we inserted based on our sentiment lexicons enlarge the distances among points belonging to different classes, while keep the distances of points belonging to the same class unchanged. Our proof is illustrated below.

Suppose point $\vec{a} = (x_1^a, x_2^a, \cdots, x_p^a)$ and point $\vec{b} = (x_1^b, x_2^b, \cdots, x_p^b)$ belonging to class **A**, and point $\vec{c} = (x_1^c, x_2^c, \cdots, x_p^c)$ and point $\vec{d} = (x_1^d, x_2^d, \cdots, x_p^d)$ belonging to class **A** and class **B** respectively.

In our experiments, we used SVM with linear kernel in which the distance among data points is measured by Euclidean distance among those points. For

example, the distance between $\vec{a}$ and $\vec{c}$ and the distance between $\vec{c}$ and $\vec{d}$ equal to $Distance_{old}(\vec{a}, \vec{c})$ and $Distance_{old}(\vec{c}, \vec{d})$ below.

$$\begin{aligned} & Distance_{old}(\vec{a}, \vec{c}) \\ =& \sqrt{(x_1^a - x_1^c)^2 + \cdots + (x_p^a - x_p^c)^2} \\ & Distance_{old}(\vec{c}, \vec{d}) \\ =& \sqrt{(x_1^c - x_1^d)^2 + \cdots + (x_p^c - x_p^d)^2} \end{aligned} \tag{5.2}$$

When we add an extra feature μ to all points belonging to class **A** and a different extra feature ν to all points belonging to class **B** according to our Class/Domain specific lexicons, we are adding an extra dimension to each of the data point. Then the new distance between $\vec{a}$ and $\vec{c}$ and the new distance between $\vec{c}$ and $\vec{d}$ can be calculated as follows.

$$\begin{aligned} & Distance_{new}(\vec{a}, \vec{c}) \\ =& \sqrt{(x_1^a - x_1^c)^2 + \cdots + (x_p^a - x_p^c)^2 + (\mu - \mu)^2} \\ =& Distance_{old}(\vec{a}, \vec{c}) \\ & Distance_{new}(\vec{c}, \vec{d}) \\ =& \sqrt{(x_1^c - x_1^d)^2 + \cdots + (x_p^c - x_p^d)^2 + (\mu - \nu)^2} \\ >& Distance_{old}(\vec{c}, \vec{d}) \end{aligned}$$

It is clear from the above calculations that the distance between $\vec{a}$ and $\vec{c}$ remains the same whereas the distance between $\vec{c}$ and $\vec{d}$ will be enlarged after the extra

feature word μ is added to all points in class **A** and ν is added to all points in class **B**.

To summarize, as illustrated in Figure 2, while the distance between points belonging to the same class remains unchanged, the distance between points belonging to different classes will be enlarged after the extra feature words are inserted according to our Class/Domain specific lexicons.

See Figure 5.5.

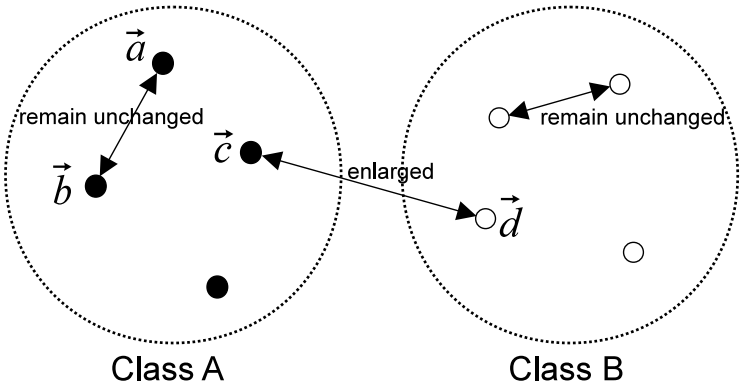

Figure 5.5. Distances between points belonging to different classes are enlarged

This also means that after adding the extra features, SVM can find a hyperplace with larger margin, or with same length of margin but less support vectors, that can separate classes more effectively. This in turn leads to higher accuracy for classification.

5.7 Conclusion

To summarize, we have shown that incorporating the knowledge encoded in sentiment lexicons, especially domain specific lexicons, can significantly improve the accuracy for fine-grained sentiment analysis tasks. We have also described how

we constructed our domain specific sentiment lexicons for the domain of camera reviews through a combination of corpus filtering, web searching and filtering and dictionary expansion. In addition, we have developed a method to incorporate the lexicon knowledge into machine learning algorithms such as SVM to improve sentiment learning. Our conclusions are supported both by our empirical studies and by our theoretical proof.

Chapter 6

Aspect-level Sentiment Analysis: Identify New Aspects

6.1 Overview

In this chapter, we focus on automatically identifying new aspects that are not covered in the training data. A formal problem denition will be given at first. And then a novel Semi-Supervised Hierarchical Topic Model (SHTM) is proposed to identify new aspects. To build SHTM, we further propose a Semi-Supervised Nested Chinese Restaurant Process (SNCRP) and use it as the prior for SHTM. Finally, SHTM is evaluated on real camera reviews.

Before presenting the problem definition, we would like to give more details on two things:

1. Differences between the terms, topic and aspect. The definitions of topic and aspect in this chapter are consistent with the ones in previous chapters. Topic is defined as multinomial distributions on words. Aspect is human la-

beled information used to describe the product features to talk about. Each sentence is assigned with an aspect. In this chapter, each node in a Hierarchical Topic Model [34] and a Semi-Supervised Hierarchical Topic Model (SHTM) represents a topic, and each sentence is assigned with a topic path that represents an aspect. Since the target of this chapter is to find new aspects that have no human labeled information, we use the topic words of the leaf node of a topic path to represent new aspects.

2. Assumptions that each sentence has one aspect and one sentiment. Some complex sentences may contain more than one aspects and sentiments. For example, the sentence "The picture quality of this camera is good, but the price is expensive." talks about the aspects of *picture quality* and *price*, and *positive* sentiment on *picture quality* and *negative* sentiment on *price*. In order to make our problem simpler, we break such complex sentences into two parts in our dataset at the positions of conjunction words like *and* and *but*. After such a preprocessing, it is reasonable to treat each sentence has one aspect and one sentiment.

6.2 Problem Definition

In this section, we formally present our problem. Suppose we have a sentence set $\mathbb{D} = \{D_1, D_2\}$, which consists of a labeled data set $D_1 = \{S_1^l, S_2^l, ..., S_i^l, ...\}$ and an unlabeled data set $D_2 = \{S_1^u, S_2^u, ..., S_i^u, ...\}$. Each sentence $S_i^l \in D_1$ has been labeled with an aspect A_i belonging to a predefined aspect set $\mathbb{A} = \{A_1, ..., A_i, ..., A_m\}$ as well as a sentiment O_i belonging to a sentiment set $\mathbb{O} = \{O_1, ..., O_i, ..., O_m\}$. Aspects can be any product features and sentiments can be

any feelings such as positive or negative opinions. The goal of this chapter is to:

- find new aspects $\mathbb{A}^{new} = \{A_1^{new}, ..., A_n^{new}\}$ from sentence subset D_2, s.t. $(\mathbb{A}^{new} \cap \mathbb{A} = \emptyset)$[1].
- assign an aspect label $A_j \in \mathbb{A}^{new} \cup \mathbb{A}$ as well as a sentiment label $O_j \in \mathbb{O}$ for each sentence $S_j^u \in D_2$.

We propose to solve this problem by using SNCRP and SHTM. Our method is described in detail in the following sections.

6.3 Proposed Method

6.3.1 Semi-Suppervised Nested Chinese Restaurant Process

Before we describe our approach in detail, we first introduce the Chinese Restaurant Process (CRP) [72] and the Nested Chinese Restaurant Process (NCRP) [34]. Since our task is aspect-level sentiment analysis on sentences. We introduce CRP and NCRP in terms of sentences, topics, aspects and sentiments.

In probability theory, CRP is a discrete-time stochastic process that models a distribution over partitions of integers. To illustrate the basic idea of CRP in the context of our task, we can imagine the following scenario as shown in Figure 6.1. There are an infinite number of topics in a topic room, each of which can be assigned to infinite number of sentences. A sequence of n sentences arrive, labeled with integers $\{1, 2, 3, ..., n\}$. K topics have been assigned to previous $n-1$

[1]Any new identified aspect $A_i^{new} \in \mathbb{A}^{new}$ is different from any $A_i \in \mathbb{A}$.

sentences, and X_k denotes the number of sentences currently choosing the topic k. For the nth sentence C_n, it will choose an topic $k \in [1, K]$ that has been assigned to some sentences or choose a new topic $K+1$ according to Eq 7.1. The parameter $\gamma \in [0, \infty)$ determines whether a sentence chooses a new topic or not. The probability of choosing a new topic is higher if a larger value of γ is chosen. No new topic will be chosen if γ is set to zero.

$$
\begin{aligned}
& p(C_n = k | X_1, \cdots, X_k, \cdots X_K, \gamma) \\
= & \begin{cases} \frac{X_k}{\gamma+n-1}, & \text{if topic } k \in [1, K] \text{ is chosen;} \\ \frac{\gamma}{\gamma+n-1}, & \text{if topic } K+1 \text{ is chosen.} \end{cases}
\end{aligned} \tag{6.1}
$$

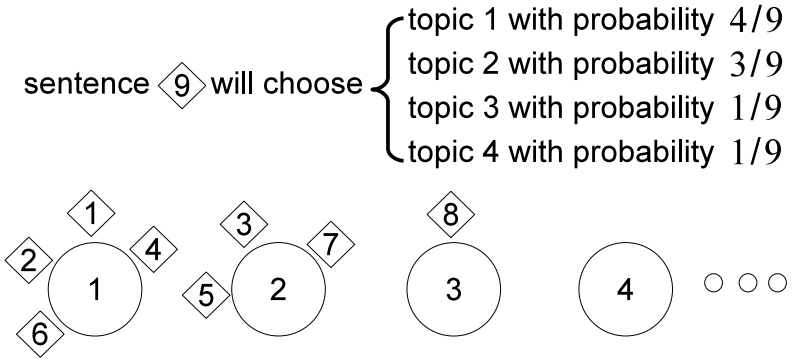

Figure 6.1. The Chinese Restaurant Process. Each circle represents a topics, and each diamonds around a topic is a sentence choosing that topic. Probabilities are calculated by using Eq 7.1 with $\gamma = 1$.

NCRP is derived from CRP by imaging that there are an infinite number of topic rooms, and each of them has an infinite number of topics. All of these topic rooms are organized into a tree structure[2], and one of them is assigned as the root room. Associated with each topic in each topic room, there is a card referring to

[2]an infinitely branched and infinitely deep tree

another topic room, and each topic room is referred to only once. Thus, all topics are connected by an infinitely branched and infinitely deep tree.

Based on this tree structure, a sentence first chooses a topic at the root topic room according to the CRP distribution Eq. 7.1. The card associated with that chosen topic indicates which topic room it should go to next. Then, this sentence goes to the topic room identified the previous topic room and chooses a topic in the same way. This sentence can repeat this process indefinitely. Thus, each sentence will choose a topic path from the root topic room to the leaf topic room, and each topic path represents an unique aspect.

In NCRP, sentences choose topics without considering any labeled or unlabeled information. However, to solve our problem, we need assign labels to unlabeled sentences. Since NCRP does not contain any labeled information, it cannot solve our problem. We need a new process, which can incorporate both labeled and unlabeled sentences, and assign labels to unlabeled sentences through label information.

To achieve this goal, we propose a novel Semi-Supervised Nested Chinese Restaurant Process (SNCRP) built upon CRP and NCRP. SNCRP differs from CRP and NCRP in that it works on both labeled and unlabeled setences, and sentences with the same label are more likely to choose the same topic and thus choose the same topic path. By contrast, in CRP or NCRP, the only factor governing whether a sentence chooses a topic is the number of sentences already choosing that topic. Furthermore, we use two parameters μ and ν in SNCRP to control the topic selection preference, where $\mu \in [1, 0]$ is used to minimize the number of sentences with different labels choosing the same topic, and $\nu \in [1, \infty)$ is used to maximize the number of sentences with the same label choosing the same topic.

The probability for sentences with the same label to choose a same topic becomes higher if a smaller value of μ and a larger value of ν are chosen. In practice, we can set μ to zero if we want to make sure that sentences with different labels will not choose the same topic. If the label information is aspect, SNCRP can assign aspects to unlabeled sentences. If the label information is sentiment, SNCRP can assign sentiment information to unlabeled sentences.

Formally, for a new labeled sentences in SNCRP, the probability of choosing a topic is calculated by Eq 7.4:

$$
\begin{aligned}
& p(C_n = k | X_1, \cdots, X_k, \cdots X_K, \gamma) \\
= & \begin{cases} \frac{X_k^{diff} \times \mu + X_k^{same} \times \nu}{\gamma + n^{diff} \times \mu + n^{same} \times \nu - 1}, & \text{if topic } k \in [1, K] \text{ is chosen;} \\ \frac{\gamma}{\gamma + n^{diff} \times \mu + n^{same} \times \nu - 1}, & \text{if topic } K+1 \text{ is chosen.} \end{cases}
\end{aligned} \tag{6.2}
$$

where X_k^{same} is the number of sentences having the same label with the new sentences choosing the topic k, and n^{same} is the total number of sentences having the same label with the new sentences in the topic room; X_k^{diff} is the number of sentences having different labels from the new sentences choosing the topic k, and n^{diff} is the total number of sentences having different labels from the new sentences in the topic room.

For a new unlabeled sentence in SNCRP, we still use Eq. 7.1 to calculate the possibility of choosing a topic.

6.3.2 Semi-Supervised Hierarchical Topic Model

Using the SNCRP defined in the previous section, we propose a Semi-Supervised Hierarchical Topic Model (SHTM). Similar to the Hierarchical Topic Model [34] based on NCPR, SHTM places a prior distribution on each path in an infinitely branched and infinitely deep tree. Each path in this tree represents an unique aspect. The depth of the tree can be determined in advance or through a stick-breaking construction[3]. SHTM differs from HTM in that while the topic path of a document is determined by the NCPR in HTM, it is determined by SNCRP in SHTM. Therefore, documents having the same label are more likely to choose the same topic path in SHTM. We describe how to model the topic of a document in the framework of SHTM below.

In SHTM, a document d is defined as a collection of 1 or more sentences, and is generated by choosing a topic path according to Eq. 6.3.

$$p(\mathbf{c}_d|\mathbf{w}_d, \mathbf{c}_{-d}, \mathbf{z}_d, \eta, \gamma) \propto p(\mathbf{c}_d|\mathbf{c}_{-d}, \gamma)p(\mathbf{w}_d|\mathbf{c}, \mathbf{w}_{-d}, \mathbf{z}, \eta), \tag{6.3}$$

In Eq. 3, $\mathbf{c_d}$ denotes the current chosen path for d, $\mathbf{c_{-d}}$ denotes the chosen paths for all of the other documents except d and $\mathbf{c}$ represents all of the paths for all of the documents; $\mathbf{w_d}$ denotes the current word in d; $\mathbf{w_{-d}}$ denotes all words except the ones in d; $\mathbf{z}$ represents the topic assignments of all documents. $p(\mathbf{c}_d|\mathbf{c}_{-d}, \gamma)$ is the prior on $\mathbf{c_d}$ implied by SNCPR (Eq 7.4), and $p(\mathbf{w}_d|\mathbf{c}, \mathbf{w}_{-d}, \mathbf{z}, \eta)$ is the posterior probability of the current document d given a particular choice of path $\mathbf{c}_d$. γ and η control the size of the inferred tree. Large values of γ encourage documents choosing new paths, while small values of η encourage fewer words dominating a

[3]For more details about stick-breaking construction, please see [72, 34]

topic.

The assignment of a topic $z_{d,n}$ to the n_{th} word $w_{d,n}$ in document d is determined by Eq. 6.4,

$$\begin{aligned} & p(z_{d,n}|\mathbf{z}_{d,-n}, \mathbf{z}_{-d}, \mathbf{c}, \mathbf{w}, m, \pi, \eta) \\ \propto\ & p(z_{d,n}|\mathbf{z}_{d,-n}, m, \pi) p(w_{d,n}|\mathbf{z}, \mathbf{c}, \mathbf{w}_{-(d,n)}, \eta), \end{aligned} \tag{6.4}$$

where $\mathbf{z}_{d,-n}$ denotes topic assignments for words in document d except the n_{th} word $w_{d,n}$; $\mathbf{z}_{-d}$ denotes topic assignments of all documents except the document d; $\mathbf{w}$ denotes all words in all documents; and $\mathbf{w}_{-(d,n)}$ denotes all words in all documents except the n_{th} word $w_{d,n}$ in document d. $p(z_{d,n}|\mathbf{z}_{d,-n}, m, \pi)$ is the prior on topics implied by the stick-breaking construction with parameters m and π, which determine whether a new topic should be generated or not. $p(w_{d,n}|\mathbf{z}, \mathbf{c}, \mathbf{w}_{-(d,n)}, \eta)$ is the posterior probability of the n_{th} word in d ($w_{d,n}$) given the current topic choices of all words in all documents, the current path choices of all of the documents, and all of the words except the n_{th} word in d.

Following this approach, SHTM can automatically assign a topic path to each document and thus discover all of the topics discussed in all of the documents. Our experiments illustrate that the topics discovered by our SHTM are more interpretable and more accurate compared to those discovered by HTM. Our experiments also show that if the topic path of some documents is a unlabeled and new aspect, SHTM can successfully identify it as well. Compared to SHTM, a conventional supervised learning approach such as SVM cannot achieve this goal. We further demonstrate that we can also use SHTM to perform polarity classification at the sentence level and achieve comparable results with those of supervised

learning approaches. Our experiments are described below.

6.4 Experiments and Results

6.4.1 Data Set and Experiment Design

The widely used data for research on sentiment analysis are product review data downloaded from Amazon website. Current available product review data are released by Blitzer [4] and Bing Liu [5], and used in research papers [69, 49, 73, 74, 75]. However, Blitzer's dataset do not provide human labeled information, and Liu's dataset do not provide human labeled information on sentence level. Regarding with our problem, we need construct dataset by ourselves.

To train and test our SHTM model, we created two separate datasets D_1 and D_2. For D_1, we manually labeled 2553 sentences extracted from the Multi-Domain Sentiment Dataset created by Blitzer at al. [69]. Each sentence is labeled with a camera aspect such as *picture quality* or *durability* and a sentiment towards that aspect. There are total 22 [6] aspects and a special category of *none*, meaning that none of the 22 camera aspects is mentioned in the sentence. Each sentence is also labeled with three categories of sentiments: positive, negative and *none*. For D_2, we extracted sentences from two different types of camera reviews. The first type includes normal digital camera reviews. We downloaded reviews for *Canon PowerShot SD780IS, SD1200IS and SD1400IS* from Amazon, from which

[4] http://www.cs.jhu.edu/ mdredze/?datasets/sentiment/?index2.html

[5] http://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

[6] The 22 aspects are: picture quality, durability, flash capability,customer support, entire camera, ease-of-use, time between consecutive shots, auto focus, accessory, performance in low light, size, battery life, zoom capability, price, design, screen, transferring process, movie capability, availability of shooting modes, reputation, stabilization, shutter lag.

we randomly selected 974 sentences (annotated as D_2^{normal}) and labeled them with camera aspects as well as their associated sentiments. Because these reviews are similar to the Blitzer dataset, all sentences in D_2^{normal} fall into the 23 aspects defined in D_1. To test SHTM's ability to find new camera aspects, we downloaded reviews for waterproof cameras including *Pentax W90, Fujifilm XP10, Fujifilm Z33WP and Olympus Stylus Tough 6000* from Amazon, from which we randomly selected 970 sentences (annotated as $D_2^{waterproof}$) and labeled them with aspects as well as their associated sentiments. In addition to the predefined 23 aspects, we identified a new aspect called "waterproof performance" for $D_2^{waterproof}$.

Our tasks are as follows: for each camera review sentence, we need to classify it in terms of the dominant camera aspect discussed in that sentence as well as the polarity towards that aspect. In our approach, we treat each sentence as a document. In terms of aspect classification, we would like to compare the performance of our approach with that of HTM and SVM. In particular, we would like to test our approach's ability to identify new camera aspects. In terms of polarity classification, we would like to compare our approach with SVM, which is considered as the state-of-the-art approach. In keeping with these objectives, we designed the following experiments, which are described in detail in the following subsections.

1. Aspect Identification Using SHTM and HTM.

2. New Aspect Identification Using SHTM.

3. Aspect and Polarity Classification Using SHTM and SVM.

Our implementation of SHTM is based on Blei's code for HTM [7]. In our experiments, SHTM runs at the sentence level, and we set the depth of the hierarchical

[7]http://www.cs.princeton.edu/ blei/downloads/hlda-c.tgz

tree as 2, the number of iterations for running SHTM as 1000, and other parameters as follows: $\eta = \{0.25, 0.125\}$, $\gamma = 1$, $m = 0.3$, $\pi = 100$, $\mu = 0.0$ and $\nu = 10.0$.

Small values of η encourage fewer words dominating a topic, while larger values of η make more words in a topic get relative higher probabilities. Most of words in the root topic are stop words or background words that have similar probabilities. Hence, we set a higher value of η for the root topic. On the other hand, leaf topics are very specific and small number of specific words dominate those topics. Hence, we set a small value of η for leaf topics. γ controls the number of topic paths. Large values of γ prefer to generate larger number of topic paths. Otherwise, smaller number of topic paths are generated. m and π control whether a new topic generates or not when a topic path is chosen. Small values of m encourage to generate new topics. π determines the variance of the distribution on topics when a topic path is chosen, and small values of π means the distributions of topics on a given topic path are close to each other. Small values of μ decrease the effect of choosing a topic by the number of sentences already choosing that topic. Larger values of ν increase the effect of choosing a topic by the number of sentences already choosing that topic and having the same label. Setting a large value of ν and a small value of μ keeps sentences of the same label choosing the same topic path.

6.4.2 Aspect Identification Using SHTM and HTM

In this experiment, we run HTM and SHTM on dataset D_1. The goal of the experiment is to compare these two models in terms of accuracy of identifying camera aspects as well as interpretability of the identified aspects. To compare the accuracy, we compare the agreement between sentences clustered into an aspect by

these two models with those manually labeled by people as belonging to the same aspect. To compare the interpretability of the identified aspects, we compare the top key words of three camera aspects extracted by SHTM and HTM.

Figure 6.2 illustrates the agreement level of the aspects identified by these two models with those that are manually labeled. In Figure 6.2, each row represents an aspect generated by HTM or SHTM. There are 21 and 25 aspects generated by HTM and SHTM respectively, which are comparable to the 23 manually-labeled aspects. Each cell in each column represents the HTM or SHTM's labeling of sentences that are manually grouped into a particular aspect. Figure 6.2(a) shows that the aspects identified by HTM are noisy, because sentences that are manually clustered into one aspect are often classified into many different aspects by HTM. For example, sentences manually labeled as "entire camera" (aspect No. 5) in Figure 6.2(a) are classified into almost all of the 23 different aspects by HTM with a high probability (shown by the light color). This means that topics identified by HTM are very different from those identified by human judgement. By contrast, Figure 6.2(b) illustrates that the topics/aspects identified by SHTM are not noisy at all, and that each aspect identified by SHTM only includes sentences belonging to exactly one manually-labeled aspect.

Next we compare the topic words generated by SHTM and HTM. Since the root topic is shared by all paths, we use the topic words of the leaf nodes to represent each aspect. We compare the topic words for the same three camera aspects generated by SHTM and HTM. For each aspect, we present the top 8 words with the highest probability. Table 6.1 shows that aspects identified by SHTM are more interpretable than those identified by HTM.

To summarize, SHTM generates aspects that are both more accurate and more

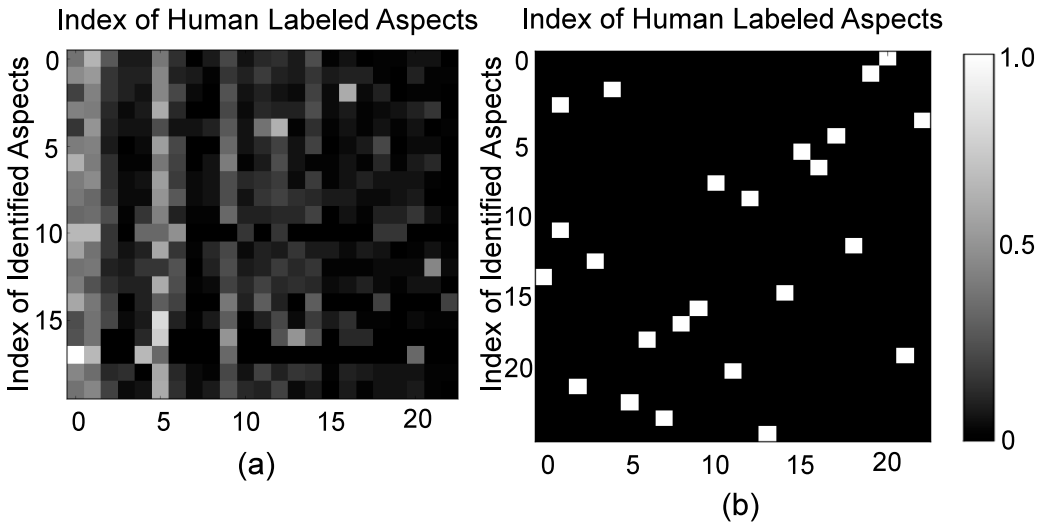

Figure 6.2. Aspect Identification Agreement. (a) Agreement Between HTM and Manual Labeling. (b) Agreement between SHTM and Manual Labeling.

Topic Words (generated by HTM)			Topics words (generated by SHTM)		
quality	battery	video	quality	battery	computer
pictures	had	them	picture	batteries	transfer
zoom	batteries	computer	sharp	life	download
takes	money	card	colors	charge	downloading
take	have	software	photos	long	card
great	been	transfer	pictures	dead	files
out	cameras	usb	image	last	minutes
light	junk	download	clear	hours	usb
picture quality	battery life	transferring process	picture quality	battery life	transferring process

Table 6.1. Topic Words Generated by HTM and SHTM

interpretable than those of HTM. We believe this is because SHTM incorporates manual labeling information in the data.

6.4.3 New Aspect Identification Using SHTM

This section describes our experiments to test SHTM's ability to identify new aspects. We ran our experiments both on dataset D_1 and D_2. We define our approach for identifying a new aspect as follows.

In each of our experiments, SHTM will generate a hierarchical tree with a set

of topic paths $\{P_1, P_2, ..., P_i, ..., P_n\}$. Each topic path P_i includes a set of sentences $\{S_i^l\} \in D_1$ (or an empty set). If the number of sentences $|\{S_i^l\}|$ is greater than zero, the majority sentence in $|\{S_i^l\}|$ becomes the aspect of P_i. If the number of sentences $|\{S_i^l\}|$ is zero, A_i^{new} becomes a new aspect of P_i. For simplicity, we use the top 6 words with the highest probability in the topic of leaf node on path P_i to label the new aspect A_i^{new}. Formally,

$$Aspect(P_i) = \begin{cases} A_i = \text{Majority}(\{S_i^l\}), & \text{if } |\{S_i^l\}| > 0 \\ A_i^{new}, & \text{if } |\{S_i^l\}| = 0 \ . \end{cases} \tag{6.5}$$

In our first experiment, we manually removed the sentences in dataset D_1 belonging to aspect *screen*, *stabilization* and *battery life* and used the rest of the sentences to construct $D_{1.1}^{test}$. We then constructed $D_{1.2}^{test}$ which consists of 100 sentences belonging to the aspect *screen*, 100 sentences belonging to *stabilization* and 100 sentences belonging to *battery life*. The sentences in $D_{1.1}^{test}$ are labeled and the sentences in $D_{1.2}^{test}$ are not labeled. The goal of this experiment is to test whether SHTM can correctly group sentences in $D_{1.2}^{test}$ into three new aspects. As shown in Figure 6.3(a), SHTM successfully identified these three new aspects (No.12, No.16 and No.21). What is more important is that none of the sentences in $D_{1.1}^{test}$ was incorrectly classified into these three new aspects; this again attests to the high accuracy of SHTM for identifying aspects, especially new topics.

In our second experiment, we ran SHTM on datasets D_1, D_2^{normal} and $D_2^{waterproof}$. The sentences in D_1 are labeled and the sentences in D_2^{normal} and $D_2^{waterproof}$ are not labeled. The goal is to test whether a new aspect about "waterproof performance" can be found from $D_2^{waterproof}$, and whether no new aspect is found from D_1 and D_2^{normal}. This is exactly the case as shown in Figure 6.3(b) and (c). Fur-

thermore, the top six words with the highest probability for the identified new aspect (No. 23 in Figure 6.3(c)) are *water, pool, beach, took, fun, under*, which suggests that the new aspect is very interpretable as well. Note that only one new aspect identified in this experiment does not mean SHTM can only handle one new aspect. As in our first experiment, more new aspects can be identified if they exist in the dataset.

We quantify the performance of SHTM on new aspect identification in two ways:

1) Precision and Recall: There are 96 sentences manually labeled as discussing "waterproof performance" in $D_2^{waterproof}$. SHTM finds 92 sentences belonging to this new aspect. The precision is 53.3% and the recall is 51.0%. An initial error analysis shows that a large portion of the sentences discussing "waterproof performance" but not clustered into this new aspect were incorrectly classified into the category of "entire camera" by SHTM. This is not surprising given that the category of "entire camera" collects all of sentences that do not fit into any of the other aspects, and thus is very noisy. Once we removed those sentences from our datasets, the precision improved to 81.5% and recall improved to 78.1%.

2) Sensitivity of SHTM: Our experiments show that SHTM can successfully identify new aspects. Further analysis illustrates that SHTM does not rely on a large number of new aspect sentences to identify the new topic. For example, in our second experiment, the percentage of sentences belonging to the new aspect of "waterproof performance" in $D_2^{waterproof}$ is $\frac{96}{970} = 9.9\%$. We have shown that SHTM can successfully identify the new aspect in this case. In addition, when we manually decreased the percentage of "waterproof performance" sentences to 5.0%, SHTM was still able to find this new aspect. This implies that SHTM is

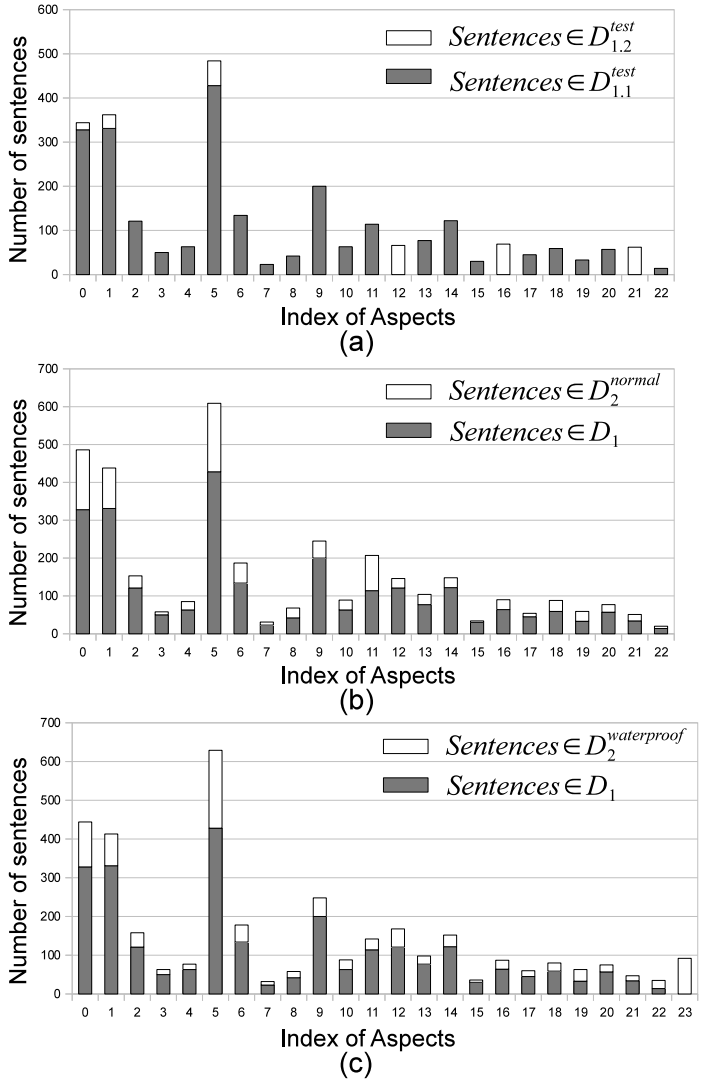

Figure 6.3. New Aspect Identification.

very sensitive to the emergence of new aspects, and can detect a new aspect even if it is only sparsely represented in the data.

In real world applications, when a new product aspect just emerges, it typically will not appear in a large number of review sentences. The high sensitivity of our SHTM, which ensures that we can identify a new aspect when it just emerges, is

therefore an important advantage in practice.

6.4.4 Aspect and Polarity classification Using SHTM and SVM

In this section, we examine the performance of SHTM on product aspect and polarity classification, and compare it with the performance of SVM, which is the state-of -the-art approach for these tasks.

For aspect classification, we compared the following three methods:

- AC1: For each unlabeled sentence $S_j^u \in D_2$ on a topic path P_i generated by SHTM, we assign the aspect name of P_i to S_j^u according to Eq 6.5.
- AC2: We first train an SVM [8] classifier on labeled sentences in D_1. We then use the trained classifier to identify aspects for sentences $S_j^u \in D_2$. Note that no new aspects can be identified using AC2.
- AC3: For sentences $S_j^u \in D_2$ identified as discussing new aspects by AC1, we keep the new aspect labels. For the rest of the sentences, we use the classifier trained in AC2 to identify the aspects.

Table 6.2 shows that when no new aspects are involved, SVM outperforms SHTM. This is not surprising given that the SHTM is a topic model, which is better at capturing word frequency information than word presence information. However, word presence does play an important role in determining the camera aspects and SVM can capture such information. For example, if a sentence men-

[8] We use libsvm http://www.csie.ntu.edu.tw/∼cjlin/libsvm/. We set the parameters $t = 0$ (linear kernel) and $c = 1.0$ and used words as features for training

tions "picture(s)", it is very likely that it is discussing the "picture quality" of the camera regardless of the frequency of word "picture(s)".

However, when new aspects are involved, AC3 outperforms both AC1 and AC2. Because it is almost impossible to construct a training corpus that can cover all aspects in real world applications, we believe that using SHTM to identify new aspects can significantly improve the accuracy of aspect classification in general.

Test Dataset	AC1	AC2	AC3
$D_2^{waterproof}$	41.7%	44.3%	**47.5%**
D_2^{normal}	43.8%	51.7%	-

Table 6.2. Accuracy of Aspect Classification by SHTM and SVM

We also compared SHTM and SVM's performance in identifying polarity O_j for sentence $S_j^u \in D_2$ that expresses either a positive or a negative sentiment. Our approaches are similar to those of identifying aspects. The difference is that for poalrity classification we need not find new aspects, because we only classify the sentence as "positive" or "negative". In this chapter, we compared the following two methods.

- PC1: We use a minor modification to SHTM. When SHTM assigns a topic path to a sentence $S_i^u \in D_2$ [9], we set the γ to zero in Eq 7.4. This ensures that no new topic paths will be generated and all topic paths can be identified either as "positive" or "negative".

- PC2: Similar to AC2, we first train an SVM classifier using D_1, and then use the trained classifier to identify the polarity of sentences $S_j^u \in D_2$.

[9] all sentences in D_2 are unlabeled, but SHTM incorporate the labels in D_1

Test Dataset	PC1	PC2
$D_2^{waterproof}$	66.8%	72.5%
D_2^{normal}	71.8%	77.8%

Table 6.3. Accuracy of Polarity Classification by SHTM and SVM

Table 6.3 shows that SVM performs better for this task. However, the performance of SHTM is comparable. Given that SHTM can significantly improve the accuracy for aspect classification, combining SHTM with SVM can lead to significant improvement for aspect level sentiment classification.

6.5 Conclusion

In this chapter, we have proposed a novel Semi-Supervised Nested Chinese Restaurant Process (SNCRP). We further proposed a Semi-Supervised Hierarchical Topic Model (SHTM) using SNCRP as the prior. We have shown that SHTM can successfully identify new aspects, and compared to HTM, the aspects identified by SHTM are both more accurate and more interpretable. We have also demonstrated that by combining with SVM learning, SHTM can significantly improve the accuracy of fine-grained sentiment classification, which involves identifying both the aspect discussed in a sentence as well as its associated polarity.

Chapter 7

Aspect-level Sentiment Analysis: Extracting Representative Sentences

7.1 Overview

In this chapter, we focus on extracting opinionists' representative sentences, instead of classifying sentences as positive or negative. A formal problem denition will be given at first. Then, a generative model will be proposed to automatically discover the hidden associations between topics words and opinion words. By applying those discovered hidden associations, a series opinion scoring models will be built to extract statements which best express opinionists' standpoints on certain topics. Finally, we will do experiments on political standpoints visualization, and opinion sentence extraction.

7.2 Problem Definition

We start by providing a set of definitions that will be used in the remainder of this chapter. In this chapter, we will call opinion holder as an opinionist denoted by $a \in A$. Where, A is the set of all opinion holder. An opinionist can be a person, or a group who share similar opinions. A topic is a subject matter an opinionist talks about. In this chapter, we define a topic $z \in Z$ as a multinomial distribution on noun words w^{noun}. An opinionist produces a collection of documents $\{D_1, D_2, ...D_i, ...D_n\}$, each of which expresses her opinions. Each document is a collection of statements $\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_i, ...\mathbf{w}_n\}$. In this chapter, we choose each sentence is a statement. A statement $\mathbf{w}$ of an opinionist a is a set of words $\{w_1, w_2, ..., w_i, ...\}$, with i indicating the position in $\mathbf{w}$. The task of this chapter is to build an opinion scoring model $Score(\mathbf{w}; a, z) = f(\{f_1(\mathbf{w}; a, z), f_2(\mathbf{w}; a, z), ..., f_i(\mathbf{w}; a, z), ..., f_n(\mathbf{w}; a, z)\})$ which assigns a real value to an opinionist a's statement $\mathbf{w}$ on a topic z, where $f_i(\mathbf{w}; a, z)$ represents i-th feature function and f is a map from a feature vector to a real value. If a statement $\mathbf{w}$ can better express her opinion on z, the opinion scoring model will assign a higher value to $\mathbf{w}$ than statements that cannot. By applying those feature functions f_i and the scoring function f, we will visualize opinionists' political standpoints, and find sentences that are the most representative of their opinion on a topic z.

7.3 Opinion Scoring Model

7.3.1 Model Overview

A statement $\mathbf{w}$ of an opinionist a could be either objective or subjective. His/her opinion is expressed through subjective ones. Even inside a subjective statement, objective and subjective information is mixed in an integrated and complex way. In order to score a statement $\mathbf{w}$ given an opinionist a and a topic z, we need to identify if it is subjective or objective. And if it is subjective, we also need to identify what topics she talks about as well as her opinion. Hence, we consider three kinds of features to score the opinion expressed by $\mathbf{w}$: subjective features, topic features and opinion features.

1. **Subjective features**. Subjective features captures whether a statement $\mathbf{w}$ expresses an opinion or not. A feature function $f_1(\mathbf{w})$ is defined on those features. If subjective features are found in a statement $\mathbf{w}$, $f_1(\mathbf{w})$ will return higher value than those statements without subjective features.

2. **Topic features**. Topic features identify what an opinionist talks about. Topics concerned in a statement $\mathbf{w}$ are expressed through noun words. A topic $z \in Z$ is defined as a multinomial distribution on noun words w^{noun}. $f_2(\mathbf{w}; \mathbf{z})$ is defined to capture topic features in $\mathbf{w}$. It will return a higher value if $\mathbf{w}^{noun}$ is more likely to be generated from a topic z.

3. **Opinion features**. Topics an opinionist talks about are conveyed by nouns, while opinions are expressed through adjective, verb and adverb words. If two opinionist have different opinions on a same topic, she will use different adjective, verb and adverb words to express their special opinions. Therefore,

the usage patterns of adjective, verb and adverb words are effective feature to capture an opinionist a's opinions on a topic z. We use three feature functions $f_3(\mathbf{w}^{adj}; a, z)$, $f_4(\mathbf{w}^{verb}; a, z)$ and $f_5(\mathbf{w}^{adv}; a, z)$ to capture the usage patterns of adjective, verb and adverb words respectively. $f_3(\mathbf{w}^{adj}; a, z)$ will return a higher value if $\mathbf{w}^{adj}$ is more likely to represent the usage of adjective words when a express her opinions on a topic z. $f_4(\mathbf{w}^{verb}; a, z)$ and $f_5(\mathbf{w}^{adv}; a, z)$ have same properties.

By incorporating above subjective, topic and opinion features, we can define the opinion scoring function as,

$$\begin{aligned} Score(\mathbf{w}; a, z) = f(&f_1(\mathbf{w}), f_2(\mathbf{w}^{noun}; z), \\ &f_3(\mathbf{w}^{adj}; a, z), f_4(\mathbf{w}^{verb}; a, z), f_5(\mathbf{w}^{adv}; a, z)). \end{aligned} \tag{7.1}$$

Obviously, Eq.1 is quite general, more feature functions can be included if needed. For convenience, we call $f_1(\mathbf{w})$ as the subjective function, $f_2(\mathbf{w}^{noun}; z)$ as the noun function, $f_3(\mathbf{w}^{adj}; a, z)$ as the adjective function, $f_4(\mathbf{w}^{verb}; a, z)$ as the verb function, $f_5(\mathbf{w}^{adv}; a, z)$ as the adverb function and f as the combination function. In the following we will discuss how to define them in detail.

7.3.2 Defining the Subjective Function

We choose *opinion clues* as basic criteria to judge whether a statement expresses an opinion or not. Or we could use OpinionFinder [76] to label which sentences are subjective. *Opinion clues* are effective features used in [77] to extract opinion sentences from blog pages. In this chapter, we use rule-based method to define

some *opinion clues*. Experiments show that rule-based clues are good enough for our application. It is also possible to collect more *opinion clues* though learning method as applied in [78]. The following lists six clues we used. For more detail, please refer to [77]:

- Thought: think, consider, ...
- Impression: confuse, bewilder, ...
- Emotion: glad, worry, ...
- Modality about propositional attitude: should, would, ...
- Utterance-specific sentence form: however, nonetheless, ...
- Certainty/Uncertainty: wondering, questioning ...

In addition, we augment the above *opinion clues* by adding their synonyms through WordNet [61] and those opinion words included in MPQA, a corpus of opinion words [30].

The subjective feature function $f_1(\mathbf{w})$ is defined on the above *opinion clues*. If one or more *opinion clues* are found in a statement $\mathbf{w}$, the returned value is 1, otherwise 0. Notice that judging whether a statement $\mathbf{w}$ is sentiment or not is independent from a specific opinionist a or topic z. We have found that this simple subjective function works well for our purpose.

7.3.3 Noun Function

We use $p(\mathbf{w}^{noun}|z)$ to calculate $f_2(\mathbf{w}^{noun}; z)$. $p(\mathbf{w}^{noun}|z)$ is the probability of generating noun words in a statement $\mathbf{w}$ given a topic z. A widely used method is

to treat $\mathbf{w}$ as a unigram model. We choose five different methods to calculate $f_2(\mathbf{w}^{noun}; z)$ from $p(w^{noun}|z)$. We use LDA model to calculate $p(w^{noun}|z)$. The only difference is that we use noun words to train the LDA model instead of all words. We run LDA on document level instead of statement level, which is too fine for LDA model. Through experiments, we find topics learned from noun words become more clear than topics learned from all words. Because of limited space, we do not introduce LDA model here, and please to refer to [3] if interested.

1. **SumLog**. A simplest way is to choose the logarithm of the product of $p(w^{noun}|z)$. By considering the length of each statement, we divide the logarithm by the length of $\mathbf{w}^{noun}$.
   $f_2(\mathbf{w}^{noun}; z) = \sum_{w^{noun} \in \mathbf{w}^{noun}} \frac{1}{|\mathbf{w}^{noun}|} log(p(w^{noun}|z))$.

2. **SumBasic**. This algorithm is introduced from the **S**UM**B**ASIC (Nenkova and Vanderwende 2005), which is a simple effective sentence extraction algorithm for multi-document summarization.
   $f_2(\mathbf{w}^{noun}; z) = \sum_{w^{noun} \in \mathbf{w}^{noun}} \frac{1}{|\mathbf{w}^{noun}|} p(w^{noun}|z)$.

3. **Max@n(n=1,2,...)**. Instead of considering all noun words, we only consider n noun words $w^{noun} \in \mathbf{w}_n^{noun}$ which have higher values $p(w^{noun}|z)$ than the rest of noun words in a statement $\mathbf{w}$. In this chapter, we will test Max@1, Max@2 and Max@3.
   $f_2(\mathbf{w}^{noun}; z) = \sum_{w^{noun} \in \mathbf{w}_n^{noun}} \frac{1}{n} p(w^{noun}|z)$.

4. **SimCos**. This algorithm treats $\mathbf{w}^{nouns}$ having an empirical unigram distribution $P_{\mathbf{w}^{noun}}$ on noun words. We use *cosine* function to calculate the similarity between $P_{w^{noun}}$ and z.
   $f_2(\mathbf{w}^{noun}; z) = cosine(P_{w^{noun}}, z)$.

5. **SimKL**. Similar to **SimCos** , we use KL-Divergence to calculate the similarity between $P_{w^{noun}}$ and z. Considering f_2 has a higher value if $P_{w^{noun}}$ is close to z, we take the reciprocal form as,
   $f_2(\mathbf{w}^{noun}; z) = 1/KL(P_{w^{noun}}||z)$.

7.3.4 Adj/Verb/Adv Function

We still apply the same ideas used in **SumLog**, **SumBasic**, **Max@n**, **SimCos** and **SimKL** to calculate $f_3(\mathbf{w}^{adj}; a, z)$. Here, we only present how to calculate $f_3(\mathbf{w}^{adj}; a, z)$. The algorithm for calculating $f_4(\mathbf{w}^{verb}; a, z)$ and $f_5(\mathbf{w}^{adv}; a, z)$ is same. Similarly, we need to calculate $p(w^{adj}|a, z)$.

$p(w^{adj}|a, z)$ is trying to capture the usage pattern of adjective words when an opinionist a talks about topic z. For example, if an environmentalist talks on topics of energy, some adjective words, like *renewable*, *sustainable* and *clean* will be used more frequently than others. That is, $p(w^{adj}|a, z)$ is to discover relations between noun and adjective words. If we model their relations directly, we will face data sparsity problem. In order to reduce such a problem, we introduce a concept of *adjective class*, c^{adj}, to reduce the dimension of adjective words, like the concept *topic* used in LDA. Thus the question is changed to find relations between adjective classes c^{adj} and topics z.

We propose a generative model to learn the Adj function. We assume an opinionist a has a multinomial distribution ψ_t on c^{adj} classes given a topic t. Given an opinionist a's statement $\mathbf{w}$, we have obtained its topic distribution θ after running LDA. Adjective words $w^{adj} \in \mathbf{w}$ are dependent on the topic distribution θ. The process of generating a adjective word w^{adj} is: first generate a topic z from θ, then generate an adjective class c^{adj} from ψ_t, and finally generate w^{adj}

from c^{adj}. Formally, as illustrated in Fig. 7.1, the ADJ component assumes the following generative process for each adjective word w^{adj} in a statement $\mathbf{w}$ (with topic distribution θ) of an opinionist a:

1. Draw $|C^{adj}|$ multinomials $\phi_{c^{adj}}$ from a Dirichlet prior β, one for each c^{adj};
2. Draw $|A| \times |T|$ multinomials $\psi_{a,t}$ from a Dirichlet prior γ, one for each a and z;
3. For each adjective word w^{adj} in $\mathbf{w}$ of a:
   (a) Draw a topic z from a multinomial θ;
   (b) Draw a adjective class c^{adj} from a multinomial $\psi_{a,z}$;
   (c) Draw a adjective word w^{adj} from a multinomial $\phi_{c^{adj}}$.

The full joint probability is,

$$\begin{aligned} & p(\mathbf{w}^{adj}, \mathbf{c}^{adj}, \mathbf{z}|\theta, a, \beta, \gamma) \\ =& p(\mathbf{w}^{adj}|\mathbf{c}^{adj}, \beta) \cdot p(\mathbf{c}^{adj}|a, \mathbf{z}, \gamma) \cdot p(\mathbf{z}|\theta) \\ =& \int p(\mathbf{w}^{adj}|\phi, \mathbf{c}^{adj}) p(\phi|\beta) d\phi \cdot \int p(\mathbf{c}^{adj}|\psi, \mathbf{z}) p(\psi|\gamma) d\psi \cdot p(\mathbf{z}|\theta). \end{aligned} \tag{7.2}$$

Using Gibbs sampling techniques, we obtain following update equations for hidden variables c^{adj} and z on the i-th position as,

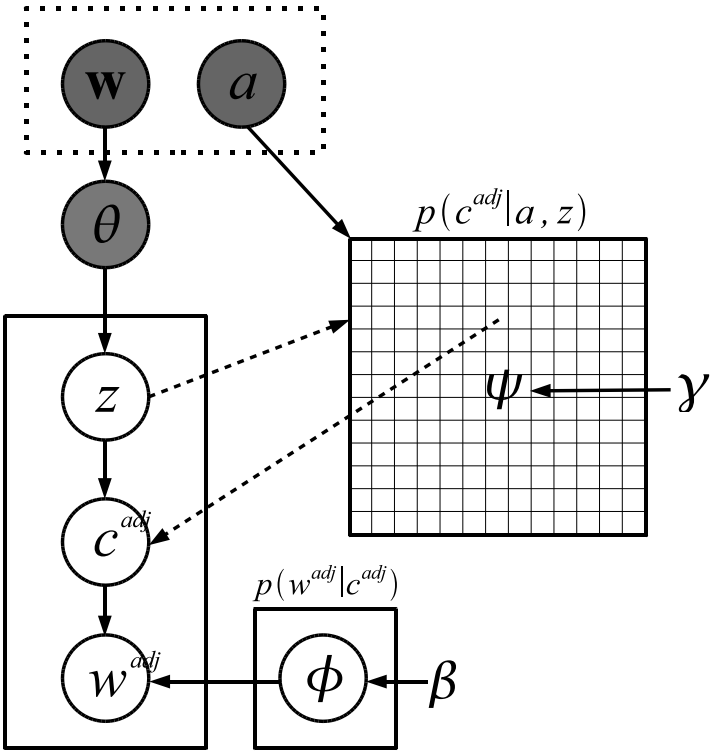

Figure 7.1. Generative Model for ADJ Component

$$
\begin{aligned}
& p(c_i^{adj}|w_i^{adj}, z_i, \mathbf{c}_{-i}^{adj}, \mathbf{z}_{-i}) \\
=& \frac{N_{a,z,c^{adj}}(a, z_i, c_i^{adj}) + \gamma}{N_{a,z}(a, z_i) + |C^{adj}| \cdot \gamma} \cdot \frac{N_{c^{adj},w}(c_i^{adj}, w_i^{adj}) + \beta}{N_{c^{adj}}(c_i^{adj}) + |V^{adj}| \cdot \beta}, and \\
& p(z_i|c_i^{adj}, \mathbf{z}_{-i}, \mathbf{c}_{-i}^{adj}, \theta) \\
=& \frac{N_{a,z,c^{adj}}(a, z_i, c_i^{adj}) + \gamma}{N_{a,z}(a, z_i) + |C^{adj}| \cdot \gamma} \cdot p(z_i|\theta),
\end{aligned}
\tag{7.3}
$$

where $N_{a,z,c^{adj}}(a, z_i, c_i^{adj})$ is the number of adjective words belonging to opinionist a simultaneously assigned with adjective class c_i^{adj} and topic z_i; $N_{a,z}(a, z_i)$ is the integration of $N_{a,z,c^{adj}}(a, z_i, c_i^{adj})$ over adjective classes; $N_{c^{adj},w}(c_i^{adj}, w_i^{adj})$ is the number of adjective words w_i^{adj} assigned with adjective class c_i^{adj}; $N_{c^{adj}}(c_i^{adj})$ is the integration of $N_{c^{adj},w}(c_i^{adj}, w_i^{adj})$ on all adjective words; $|C^{adj}|$ is the number of

adjective classes; and $|V^{adj}|$ is the size of adjective vocabulary.

From the model, we can learn $p(c^{adj}|a,z)$ and $p(w^{adj}|c^{adj})$. The Adj component $p(w^{adj}|a,z)$ can be obtained from $p(w^{adj}|a,z) = \sum_{c^{adj} \in C^{adj}} p(w^{adj}|c^{adj}) \cdot p(c^{adj}|a,z)$.

In essence, the relations between noun and adjective words we hope to discover are based on their co-occurrence. The boundary of co-occurrence in the current model is considered on statement level. If we use dependency parsing on statements in advance, we can reduce the boundary of co-occurrence, and find more accurate relations between noun and adjective words. We will leave it to future research.

7.3.5 Combination Function

We use two methods to combine above features. One is to train a linear regression model, as

$$\begin{aligned} f_{Linear} =& \alpha_0 + \alpha_1 \cdot f_1(\mathbf{w}) + \alpha_2 \cdot f_2(\mathbf{w}^{noun}; z) \\ &+ \alpha_3 \cdot f_3(\mathbf{w}^{adj}; a, z) + \alpha_4 \cdot f_4(\mathbf{w}^{verb}; a, z) \\ &+ \alpha_5 \cdot f_5(\mathbf{w}^{adv}; a, z)). \end{aligned} \tag{7.4}$$

The other is to train a SVR model, as

$$\begin{aligned} f_{\text{SVR}} =& \text{SVR}(f_1(\mathbf{w}), f_2(\mathbf{w}^{noun}; z), f_3(\mathbf{w}^{adj}; a, z) \\ & f_4(\mathbf{w}^{verb}; a, z), f_5(\mathbf{w}^{adv}; a, z)). \end{aligned} \tag{7.5}$$

We manually use some labeled data to learn a linear model and a SVR model. By incorporating **SumLog**, **SumBasic**, **Max@n**, **SimCos** and **SimKL** , we con-

struct 10 opinion scoring model, annotated as **Linear-SumLog**, **Linear-SumBasic**, **Linear-Max@n**, **Linear-SimCos**, **Linear-SimKL**, **SVR-SumLog**, **SVR-SumBasic**, **SVR-Max@n**, **SVR-SimCos** and **SVR-SimKL** .

7.4 Experiments

7.4.1 Data Collection

We downloaded the statement records of senators through the Project Vote Smart WebSite [1]. These statement records present the political stances of senators. Because some senators retired and their records are not publicly available, we got a total 15,512 statements from 88 senators. On average, each senator issued 176 statements of 214 words each. Then, we used the Part-of-Speech tagging function provided by MontyLingua Python library [2] to classify tokens into nouns, adjectives, verbs and adverbs. We total obtain 2,146,052 noun words, 695,730 adjective words, 412,468 verb words, and 56,033 adverb words. We also build a baseline where only subjectivity is considered.

7.4.2 Political Standpoints Visualization

Visualization of opinion can reduce users' cognitive efforts. Our opinion scoring model can be used for opinion visualization although it is not the main focus of our paper. In our first set of experiments, we use the associations identified by our model to visualize the similarities and dissimilarities between Republican and Democratic senators with respect to various topics.

[1] http://www.votesmart.org
[2] http://web.media.mit.edu/hugo/montylingua/index.html

We set the number of topics, Z, to be 200. We grouped adjectives, verbs, and adverbs into opinion word classes C^{opi}. Each topic was given 2 classes of opinion words (the idea is that one of the classes would be frequently associated with statements by Democrats and the other with statements by Republicans), so that the total number of opinion word classes C^{opi} is 400. Now, since some senators rarely make statements on certain issues, so for each of the discovered topics we examined the 20 senators who made the most statements about that topic. To quantify the difference between the Republican and Democratic stances on a topic z, we used the function $Diff(z)$ defined as:

$$Diff(z) = |\frac{1}{|A^1|}\sum_{a\in A^1}(x^a_{z,1} - x^a_{z,2}) - \frac{1}{|A^2|}\sum_{a\in A^2}(x^a_{z,1} - x^a_{z,2})| \tag{7.6}$$

where a represents a senator, A^1 is the set of Democratic senators, and A^2 is the set of Republican senators. For each topic z and for each senator a, the quantities $x^a_{z,1}$ and $x^a_{z,2}$ are the components of the multinomial distribution (associated with senator a) over the two opinion classes associated with topic z. Due to space constraints, we only present 8 representative topics as well as how differences exist between two parties. The results are shown in Figure 7.2 (for readability, we manually labeled these 8 topics).

From Figure 7.2, we can see that the Democratic and Republic parties have quite different stances on topics of *Iraq war*, *health insurance* and *stem cell research*. On the other hand, two parties have quite similar stances on topics like *homeland security*, *veteran service*, *market investment* and *climate research*. With respect to the topic *oil energy*, two parties have mild differences.

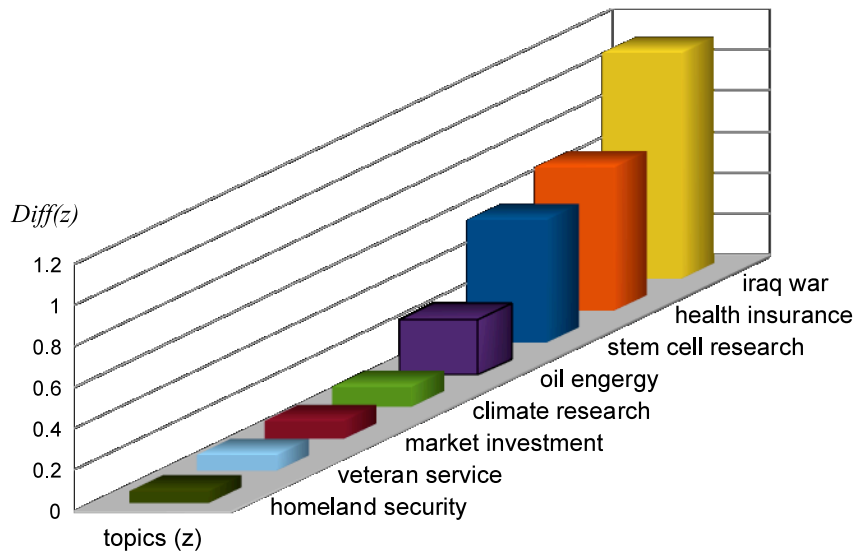

Figure 7.2. Different Stands Between Two Parties (For convenience, we only show human labeled topics instead of original distribution on noun words)

We also manually checked the corresponding statements on these topics, and obtained the same results. For the topic of *Iraq war*, senators from the two parties hold entirely different views. Democrats think "The Iraq War has made America less secure and has been a major influence on our weakening economy. We owe it to all Americans to change course in Iraq and bring a responsible end to this war. (Harry Reid)". They are criticized by the Republicans as "having given up on the idea of winning in Iraq (Lindsey Graham)". *Stem Cell* research is another controversial battlefield. While the Democrats overwhelmingly praise it as "holding promise for the treatment of a number of diseases and conditions, and giving new hope for scientific breakthroughs (Bob Casey)", the Republicans concern more on the ethicality issues. They emphasize that "Destroying viable embryos is not a choice we should or have to make.(Lindsey Graham)". *Climate Change Research* gets support from both aisles. While John Kerry claims that "it's about time we see the issue of global climate change receiving the attention it deserves.", Olympia

Snowe also states that "with science indicating a ninety-percent certainty of a direct link between human activity and climate change, Americans must take hold of this window of opportunity to reduce our current levels of carbon dioxide."

This experiment shows that our model can effectively extract hidden associations between topic and opinion words for different opinionists. Those hidden associations also effectively represent opinionists' stances on various topics. People are inclined to paying close attention to controversial topics. Our model provides a way to automatically discover those controversial public issues.

7.4.3 Opinion Sentence Extraction

Visualizing topics which are controversial or consistent between two parties is not enough. We also need to know their personal points of view. In this part, we will do a quantitative experiment to evaluate the performance of our proposed opinion scoring models. For the selected senator and topic, we will extract 5 sentences which can express their stands best using the opinion scoring model. Since our model is different from the models [63, 64] mentioned in the related section which need opinion word sets in advance. Both cannot be compared directly, and thus we only compare the models proposed in this chapter. We instantiate the method **Max@n** to **Max@1** , **Max@2** and **Max@3** . So in total, we have 14 models for comparison.

We manually labeled $1,250$ sentences for training the combination model. We randomly selected 5 topics and selected one senator for each topic. For each combination of a topic and a senator, we extracted 250 sentences. We gave a score to each sentence based on the following criteria: 1) score 5: strong opinion sentence related to the given topic, 2) score 4: weak opinion sentence related to the given

topic, 3) score 2: not an opinion sentence but related to the given topic, and 4) score 1: not an opinion sentence and not related to the given topic.

We select 15 topics, and 5 senators for each topic for testing. So we have 75 different combination of topics and senators. For each combination, we generate 5 sentences for each model. Thus we manually evaluate $5,250$ sentences. For the evaluation, we adopt three metrics, which capture the performance at different aspects:

- **Mean Reciprocal Rank (MRR)**. MRR measures the relevance of the first ranked sentence, averaged over all results. MRR provides the insight in the ability of the opinion scoring models to return a relevant sentence at the topic of the ranking.

- **Success at rank k (S@k)**. S@k defines the success at rank k, which reflects the probability of finding a relevant sentence among the top k recommended sentences. We will evaluate the results using S@1 and S@5.

- **precision at rank k (P@k)**. P@k reports the precision at rank k, which is defined as the proportion of extracted sentences that is relevant, averaged over all results. We will evaluate the results using P@5.

We have tested different settings for the number of topics, classes of adjective, verb and adverb words. When we set the topic number $Z = 200$, adjective class number $C^{adj} = 100$, verb class number $C^{verb} = 100$, and adverb class number $C^{adv} = 50$, we could obtain reasonable results for opinion sentence selection. Because of limited space, we only report results under those settings. Table 7.1 lists the results of opinion sentences extraction using 14 models.

Method	MRR	S@1	S@5	P@5
Linear-SumLog	0.69	0.51	0.61	0.39
SVR-SumLog	0.72	0.62	0.70	0.45
Linear-SumBasic	0.67	0.52	0.81	0.53
SVR-SumBasic	0.84	0.79	0.93	0.69
Linear-Max@1	0.93	**0.90**	**0.97**	0.83
SVR-Max@1	**0.95**	**0.90**	**0.97**	0.84
Linear-Max@2	0.82	0.75	**0.97**	0.69
SVR-Max@2	0.90	0.87	**0.97**	0.78
Linear-Max@3	0.79	0.65	0.90	0.61
SVR-Max@3	0.87	0.80	**0.97**	0.71
Linear-SimCos	0.91	0.85	**0.97**	0.81
SVR-SimCos	0.93	0.89	**0.97**	**0.85**
Linear-SimKL	0.79	0.72	0.82	0.73
SVR-SimKL	0.85	0.78	0.86	0.75
Baseline Model	<0.05	<0.05	<0.05	<0.05

Table 7.1. Results of Opinion Scoring Models

From the Table 7.1, we can see the quite low precision of the baseline. Among all models, SVR non-linear method is the best. That means whether or not a sentence has strong/weak/non opinion associated with a topic is decided by a complex combination of its topic, adjective, verb and adverb features. With regard to different methods, we note that SVR-Max@1, SVR-Max@2 and SVR-SimCos obtain the best performance. From this results, we can see the opinion and topic associated to a sentence is usually determined by one or two important words. Such a result is in accordance with our intuition. When we read a sentence, we can judge what it talks about and what opinion it expresses just using a few significant words, instead of the average words in that sentence. We also note that SVR-SimCos is better than SVR-SimKL. The reason is that Cosine is more prefer to high frequent components, while KL is more prefer to low frequent components.

Next, we examine how noun, adjective, verb and adverb features contribute to the opinion sentence extraction. We will quantify contributions of noun, adjective,

verb and adverb features to the opinion sentence extraction under the SVR-SimCos model. (We obtain the same results under SVR-Max@1 and SVR-Max@2, and thus omit them).

Feature Combination	MRR	S@1	S@5	P@5
Noun	0.50	0.47	0.58	0.38
Noun+Adj	0.90	0.84	0.93	0.80
Noun+Adj+Verb	**0.93**	**0.89**	**0.97**	**0.85**
Noun+Adj+Verb+Adv	**0.93**	**0.89**	**0.97**	**0.85**

Table 7.2. Contribution of Noun, Adjective, Verb and Adverb Features

The first row in the Table 7.2 is essentially a baseline where we only consider subjective and topic-related measures. The following rows show promotions after adjective, verb and adverb features applied. We can see that adjective feature are the most important feature for opinion sentence extraction. Verb features also contribute a little for opinion mining, but not as significant as adjective words. However, we do not see any contributions from adverb features. The first reason why adverb feature is not significant is that the number of adverb words are less than 1/7 number of adjective and verb words. The associations between noun and adverb words are not clear as adjective and verb words do. Here, we give a concrete example, a topic of *Climate Change* and *California Senator Feinstein*, to show how adjective and verb features contribute for opinion sentences extraction. When he talked about this topic, he used adjective words such as *significant* and *environmental* with high frequency, and use verb words such as *combat* and *make* with high frequency. Hence, the model extracts opinion sentences, like "*Climate change is the most significant environmental challenge we face, and i believe that lowering the ethanol tariff will make it less expensive for the united states to combat global warming.*", to represent his opinion.

7.5 Conclusion

In this chapter, we build a generative model to find hidden associations between topics words and opinion words, and construct the opinion scoring models to extract sentences which can best represents opinionists' stances. In this chapter, we do not use any grammar analysis among topic and opinion words. In the future work, we will apply grammar structure of sentences to help on identifying hidden associations between topics and opinion words, and promote the performance of the opinion scoring models.

Chapter 8

Conclusion and Future Work

Five different models are proposed in this thesis. The first two models exploit topic evolution with social network and citation network: 1) predict future topics in blogspere, and 2) identify topic evolution in scientific papers . The rest three models focus on three tasks around aspect-level sentiment analysis: 1) incorporate lexicons to improve the precision of sentiment classification, 2) identify new aspects which are not labeled by human, and 3) extract representative sentences for opinionists.

All Five models are rigorously validated using both real and synthetic experimental data. (1) The first model, through exploiting social networks in the blogspace, can can predict future topics in blogsphere for the next 4 weeks with high precision (0.94); (2) The second model, by applying citation networks in scientific literature, can construct the map of research topic evolution and measure topic influence with accuracy (0.65) comparable to human ratings (0.76); (3) The third model, by incorporating sentiment lexicons as prior knowledge with machine learning approaches such as Support Vector Machine, can significantly improve the accuracy of sentiment analysis with 5% compared with the state of arts methods;

(4) The fourth model, through semi-supervised Chinese Restaurant Process, can find new aspects with high precision (0.82) and recall (0.78); and (5) The fifth model, through discovering associations between topic and opinion words, can find and visualize most controversial topics and extract opinion sentences to represent opinionists standpoints with high accuracy (0.97).

Some future work has been discussed in the previous chapters. Some more future work is discussed as followings.

The first model in this thesis is to predict future topics in the blogsphere, which only uses information within the blogsphere. In reality, topic evolution in the blogsphere is not only affected by history contents and social network in the blogsphere, but also affected by contents, like other blogspheres. It is interesting to investigate how blogspheres affect each other, and how topic evolution among several blogspheres.

The second model in this thesis is to track the topic evolution in scientific papers. However, this model does not consider why topics transit in scientific paper over years. The topic transition in scientific papers may be affected by economics, politics. It is valuable to investigate how documents about economics and politics affect the topic evolution in scientific papers.

The rest of three models in this thesis all focus on aspect-level sentiment analysis. Although lots of research has been done in this area, the performance of sentiment analysis is still far from perfect. The main reason comes from the complexity of natural language. It is necessary to develop new models that can understand natural language better to improve the accuracy of sentiment analysis.

Bibliography

[1] Dumais, S. T., G. W. Furnas, and T. K. Landauer (1990) "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, **41**, pp. 391–407.

[2] Hofmann, T. (1999) "Probabilistic Latent Semantic Indexing," in *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99)*.

[3] Blei, D. M., A. Y. Ng, and M. I. Jordan (2003) "Latent Dirichlet Allocation," *Machine Learning Research*, **3**, pp. 993–1022.

[4] Burges, C. J. C. (1998) "A Tutorial on Support Vector Machines for Pattern Recognition," *Journal of Data Mining and Knowledge Discovery*, **2**.

[5] Rosen-Zvi, M., T. Griffiths, M. Steyvers, and P. Smyth (2004) "The Author-Topic Model for Authors and Documents," in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*.

[6] Blei, D. M. and J. D. Mcauliffe (2007) "Supervised Topic Models," *Advances in Neural Information Processing Systems*, **21**.

[7] Mei, Q., X. Ling, M. Wondra, H. Su, and C. Zhai (2007) "Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs," in *In the Proceedings of the 16th International Conference on World Wide Web*, pp. 171–180.

[8] Lin, C. and Y. He (2009) "Joint sentiment/topic model for sentiment analysis," in *Proceeding of the 18th International Conference on Information and Knowledge Management*.

[9] Titov, I. and R. McDonald (2008) "A Joint Model of Text and Aspect Ratings for Sentiment Summarization," in *In the Proceedings of the 46th Meeting of Association for Computational Linguistics*.

[10] Kumar, R., J. Novak, P. Raghavan, , and A. Tomkins (2003) "On the Bursty Evolution of Blogspace," in *Proceedings of the International Conference on World Wide Web*.

[11] Chi, Y., B. L. Tseng, and J. Tatemura (2006) "Eigen-trend: Trend Analysis in the Blogosphere Based on Singular Value Decompositions," in *Proceedings of the 15th International Conference on Information and Knowledge Management*.

[12] Metzler, D., Y. Bernstein, W. B. Croft, A. Moffat, and J. Zobel (2005) "The Recap System for Identifying Information Flow," in *Proceedings of the 25th SIGIR Conference on Research and Development in Information Retrieval*.

[13] Qi, Y. and K. S. Candan (2006) "Cuts: Curvature-based Development Pattern Analysis and Segmentation for Blogs and other Text Streams," in *Proceedings of the 17th Conference on Hypertext and Hypermedia*.

[14] Song, X., B. L. Tseng, C.-Y. Lin, and M.-T. Sun (2006) "Personalized Recommendation Driven by Information Flow," in *Proceedings of the 29th International SIGIR Conference on Research and Development in Information Retrieval*.

[15] Kumar, R., J. Novak, and A. Tomkins. (2006) "Structure and Evolution of Online Social Networks," in *Proceedings of the 12th SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[16] Qamra, A., B. Tseng, and E. Y. Chang (2006) "Mining Blog Stories using Community-based and Temporal Clustering," in *Proceedings of the 15th Conference on Information and Knowledge Management*.

[17] Gruhl, D., R. Guha, D. Liben-Nowell, and A. Tomkins (2004) "Information Diffusion through Blogspace," in *Proceedings of the 13th International World Wide Web Conference*.

[18] Mei, Q. and C. Zhai (2005) "Discovering Evolutionary Theme Patterns from Text: an Exploration of Temporal Text Mining," in *Proceedings of the 11th SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[19] Mei, Q., C. Liu, H. Su, and C. Zhai (2006) "A Probabilistic Approach to Spatiotemporal Theme Pattern Mining on Weblogs," in *Proceeding of the 15th International Conference on World Wide Web*.

[20] Mrchen, F., M. Dejori, D. Fradkin, J. Etienne, B. Wachmann, and M. Bundschus (2008) "Anticipating Annotations and Emerging Trends in Biomedical Literature," in *Proceedings of the 14th SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[21] Morinaga, S. and K. Yamanishi (2004) "Tracking Dynamics of Topic Trends using a Finite Mixture Model," in *Proceedings of the 10th SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[22] Schult, R. and M. Spiliopoulou (2006) "Discovering Emerging Topics in Unlabelled Text Collections," in *Proceedings of East European ADBIS Conference*.

[23] Spiliopoulou, M., I. Ntoutsi, Y. Theodoridis, and R. Schult (2006) "Monic: Modeling and Monitoring Cluster Transitions," in *Proceedings of the 12th SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[24] Zhou, D., X. Ji, H. Zha, and C. L. Giles (2006) "Topic Evolution and Social Interactions: How Authors Effect Research," in *Proceedings of the 15th International Conference on Information and Knowledge Management*.

[25] Bolelli, L., S. Ertekin, D. Zhou, and C. L. Giles (2009) "Finding Topic Trends in Digital Libraries," in *Proceedings of the Ninth ACM/IEEE Joint Conference on Digital Libraries*.

[26] Bolelli, L., S. Ertekin, and C. L. Giles (2009) "Topic and Trend Detection in Text Collections using Latent Dirichlet Allocation," in *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*.

[27] Gohr, A., A. Hinneburg, R. Schult, and M. Spiliopoulou (2009) "Topic Evolution in a Stream of Documents," in *Proceedings of the 9th SIAM International Conference on Data Mining*.

[28] Liu, B. (December, 2006) *Web Data Mining*, Springer.

[29] Zhou, L. and P. Chaovalit (2008) "Ontology-supported Polarity Mining," *Journal of the American Society for Information Science and Technology*.

[30] Wiebe, J., T. Wilson, and C. Cardie (2005) "Annotating Expressions of Opinions and Emotions in Language," *Journal of Language Resources and Evaluation*.

[31] Mohammad, S., C. Dunne, and B. Dorr (2009) "Generating High-coverage Semantic Orientation Lexicons from Overtly Marked Words and a Thesaurus," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

[32] Pang, B., L. Lee, and S. Vaithyanathan (2002) "Thumbs up?: Sentiment Classification using Machine Learning Techniques," in *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing*.

[33] Gamon, M. (2004) "Sentiment Classification on Customer Feedback Data: Noisy Data, Large Feature Vectors, and the Role of Linguistic Analysis," in *Proceedings of the 20th international conference on Computational Linguistics*.

[34] Blei, D. M., T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum (2004) "Hierarchical Topic Models and the Nested Chinese Restaurant Process," in *Proceedings of Advances in Neural Information Processing Systems*.

[35] Adamic, L. and N. Glance (2005) "The Political Blogosphere and the 2004 U.S. Election: Divided They Blog," in *Proceedings of the 3rd international workshop on Link discovery*.

[36] Licamele, L. and L. Getoor (2006) "Social Capital in Friendship-Event Networks," in *Proceedings of the 6th International Conference on Data Mining*.

[37] Glance, N. S., M. Hurst, and T. Tomokiyo (2004) "BlogPulse: Automated Trend Discovery for Weblogs," in *WWW'04 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics*.

[38] Shen, D., J.-T. Sun, Q. Yang, and Z. Chen (2006) "Latent Friend Mining from Blog Data," in *Proceedings of the 6th International Conference on Data Mining*.

[39] Blei, D. M. and J. D. Lafferty (2006) "Dynamic Topic Models," in *Proceedings of the 23rd international conference on Machine learning*.

[40] Wang, C., D. Blei, and D. Heckerman (2008) "Continuous Time Dynamic Topic Models," in *Proceedings of Uncertainty in Artificial Intelligence*.

[41] AlSumait, L., D. Barbar, and C. Domeniconi (2008) "On-line LDA: Adaptive Topic Models for Mining Text Streams with Applications to Topic Detection and Tracking," in *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*.

[42] Gohr, A., A. Hinneburg, R. Schult, and M. Spiliopoulou (2009) "Topic Evolution in a Stream of Documents," in *Proceedings of the 9th SIAM International Conference on Data Mining*.

[43] Mann, G. S., D. Mimno, and A. McCallum (2006) "Bibliometric Impact Measures Leveraging Topic Analysis," in *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital libraries*.

[44] Pang, B. and L. Lee (2008) "Opinion Mining and Sentiment Analysis," in *Foundations and Trends in Information Retrieval*.

[45] Wilson, T., J. Wiebe, and P. Hoffmann (2005) "Recognizing Contextual Polarity in Phrase-level Sentiment Analysis," in *Proceedings of the International Conference on Human Language Technology and Empirical Methods in Natural Language Processing*.

[46] melville, P., W. Gryc, and R. D. Lawrence (2009) "Sentiment Analysis of Blogs by Combining Lexical Knowledge with Text Classification," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[47] Dang, Y., Y. Zhang, and H. Chen (2010) "A Lexicon Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews," *Journal of IEEE Intelligent Systems*, **25**(4).

[48] Sindhwani, V. and P. Melville (2008) "Document-word Co-regularization for Semi-supervised Sentiment Analysis," in *Proceedings of the 8th IEEE International Conference on Data Mining*.

[49] Hu, M. and B. Liu (2004) "Mining and Summarizing Customer Reviews," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[50] Kim, S.-M. and E. Hovy (2004) "Determining the Sentiment of Opinions," in *8ernational Conference on Computational Linguistics*.

[51] Zhuang, L., F. Jing, and X. Zhu (2006) "Movie Review Mining and Summarization," in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*.

[52] Hu, M. and B. Liu (2004) "Mining Opinion Features in Customer Reviews." in *In Proceedings of the 19th National Conference on Artificial Intelligence*, pp. 755–760.

[53] Wu, Y., Q. Zhang, X. Huang, and L. Wu (2009) "Phrase Dependency Parsing for Opinion Mining," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

[54] Lin, C. and Y. He (2009) "Joint Sentiment/Topic Model for Sentiment Analysis," in *In the Proceeding of the 18th ACM Conference on Information and knowledge Management*.

[55] Titov, I. and R. McDonald (2008) "Modeling Online Reviews with Multi-Grain Topic Models," in *In the Proceedings of 17th International Conference on World Wide Web*.

[56] Li, S., H. Zhang, W. Xu, G. Chen, and J. Guo (2010) "Exploiting Combined Multi-level Model for Document Sentiment Analysis," in *Proceeding of the International Conference on Pattent Recognition*.

[57] Yessenalina, A., Y. Choi, and C. Cardie (2010) "Automatically Generating Annotator Rationales to Improve Sentiment Classification," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*.

[58] Fang, J., B. Price, and L. Price (2010) "Pruning Non-Informative Text Through Non-Expert Annotations to Improve Aspect-Level Sentiment Classification," in *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*.

[59] Lu, Y. and C. Zhai (2008) "Opinion Integration through Semi-supervised Topic Modeling," in *Proceeding of the 17th International Conference on World Wide Web*.

[60] Popescu, A. and O. Etzioni (2005) "Extracting Product features and Opinions from Reviews," in *In the proceedings of Human Language Technology and Empirical Methods in Natural Language Processing*.

[61] Fellbaum, C. (ed.) (1998) *WordNet: An Electronic Lexical Database*, Cambridge, MA: MIT Press.

[62] Takamura, H., T. Inui, and M. Okumura (2006) "Latent Variable Models for Semantic Orientations of Phrases," in *In the Proceedings of the European Chapter of the Association for Computational Linguistics*.

[63] Su, Q., X. Xu, H. Guo, X. Guo, X. Wu, B. S. Xiaoxun Zhang, and Z. Su (2008) "Hidden Sentiment Association in Chinese Web Opinion Mining," in *In the Proceedings of 17th International Conference on World Wide Web*.

[64] Du, W. and S. Tan (2009) "An Interative Reinforcement Approach for Fine-Grained Opinion Mining," in *In the Proceedings of the North American Chapter of the ACL*.

[65] Huang, G.-B., Q.-Y. Zhu, and C.-K. Siew (2004) "Extreme Learning Machine: A New Learning Scheme of Feedforward Neural Networks," in *International Joint Conference on Neural Networks*.

[66] Tomandl, D., Andreas, and Schober (2001) "A Modified General Regression Neural Network (MGRNN) with new, efficient training algorithms as a robust 'black box'-tool for data analysis," *Journal of Nerual Networks*, **14(8)**, pp. 1023–1034.

[67] Teh, Y. W., M. I. Jordan, M. J. Beal, and D. M. Blei (2006) "Hierarchical Dirichlet Processes," *Journal of the American Statistical Association*, **101(476)**, pp. 1566–1581.

[68] Griffiths, T. L. and M. Steyvers (2004) "Finding Scientific Topics," in *Proceedings of the National Academy of Science*, 101, pp. 5228–5235.

[69] Blitzer, J., M. Dredze, and F. Pereira (2007) "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*.

[70] C.J. Van Rijsbergen, S. R. and M. Porter (1980) "New models in probabilistic information retrieval," *British Library Research and Development Report*.

[71] Fan, R.-E., P.-H. Chen, and C.-J. Lin (2005) "Working Set Selection Using the Second Order Information for Training SVM," *Journal of Machine Learning Research*.

[72] Aldous, D. J. (1985) "Exchangeability and Related Topics," *Lecture Notes in Mathematics*, **1117**.

[73] Jindal, N. and B. Liu (2007) "Review Spam Detection," in *Proceedings of the 16th international conference on World Wide Web*.

[74] ——— (2008) "Opinion Spam and Analysis," in *Proceedings of First ACM International Conference on Web Search and Data Mining*.

[75] Jindal, N., B. Liu, and E.-P. Lim (2010) "Finding Unusual Review Patterns Using Unexpected Rules," in *Proceeding of the 19th ACM International Conference on Information and Knowledge Management*.

[76] Wilson, T., P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan (2005) "OpinionFinder: a System for Subjectivity Analysis," in *Proceedings of the HLT/EMNLP on Interactive Demonstrations*.

[77] Furuse, O., N. Hiroshima, S. Yamada, and R. Kataoka (2007) "Opinion Sentence Search Engine on Open-domain Blog," in *Proceedings of the 20th International Joint Conference of Artificial Intelligence*.

[78] Riloff, E. and J. Wiebe (2003) "Learning Extraction Patterns for Subjective Expressions," in *Proceedings of the Internatioanl Conference on Empirical Methods in Natural Language Processing.*