Blog Content Mining: Topic Identification and Evolution Extraction

NG, Kuan Kit

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Master of Philosophy
in
Systems Engineering and Engineering Management

The Chinese University of Hong Kong
September 2009

Thesis / Assessment Committee

Professor YAN, Houmin (Committee Chair)

Professor LAM, Wai (Thesis Supervisor)

Professor YANG, Christopher C. (Internal Examiner)

Professor WANG, Jun (External Examiner)

摘要

部落格(Blog)是由 Web2.0 所發展出最具代表性的應用功能之一，它在近年的迅速發展令其成爲我們現今吸取資訊的主要途徑。但因部落格有別於傳統文章的獨特內容風格，使我們面臨著如何才能有效地搜尋及分析部落格上文章的問題。以見及此，我們針對部落格上的內容探勘(blog content mining)提出了一套非監督式學習方法(unsupervised learning method)名爲概念分群法(concept clustering)。我們應用概念分群法把部落格上的文章分類爲不同的主題及通過追溯在主題演化圖(Topic Evolution Graph)中相似的主題在不同時期之變化，抽取出顯著的主題演化(Topic Evolution)。

有別於已有的研究，從我們的方法所得出之主題群組有著相關的關鍵字及文章作爲其依據，另外抽取出來的主題演化也能使我們更容易去了解某一主題在不同時期的發展。最後我們把概念分群法與現今一些常用的方法作比較，結果亦顯示出我們的方法有著更好的表現，而且所得出的主題演化亦更能設合我們的需求。

Abstract

Blog is one of the most representative applications in Web 2.0 and its tremendous increase in recent years makes it to be the largest ground for our information needs. However, we face the problem of effectively searching and analyzing blog posts due to the specific nature of blog entries. In the thesis, we propose an unsupervised learning model called concept clustering for blog content mining. We apply concept clustering to group blog posts into different topics and capture the topic evolutions by tracking similar topics in different time periods.

Different from existing works, a topic extracted in our model is supported by both representative keywords and blog entries. In addition, the topic evolution extracted from our model enables us to easily keep track of the development of certain top-

ics. Moreover, the performance of our blog clustering method is much better than traditional ones and the topic evolution discovered in our model is both useful and beneficial to our needs.

Acknowledgement

I would like to thank all those who gave me support and contributed to this thesis.

First of all, I would like to thank my supervisors, Prof. Chris Yang and Prof. Wai Lam for giving me guidance and valuable advice. I cannot finish my thesis without their support. In addition, I would like to thank Prof. Houmin Yan as my examination committee for giving me suggestion to improve my thesis.

Moreover, I also wish to thank my schoolmates Chau, Dickson, Sampson, Eddie, Alex, Jacky, Shamsion, Forest, Miranda, Meiting, WingWing for serving me emotional support and entertainment. I am also grateful to my entire high school friends, particularly Gordon, Ron, Carlos, Ngaio, Stanley and Jonathan for their care.

Finally, I would like to thank my family members' encouragement and consideration. Their selfless love and dedication enable me to complete my research work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The advent of Web 2.0 has created a platform which encapsulates the idea of proliferation of interconnectivity and interactivity of web content. It changed the way web users interact and a surge of web content has been created via online media. One of the prominent applications of Web 2.0 is blogging where individuals can write anything they want on their self-developed web pages. Though blogging is not a new phenomenon that appeared in the late 1990's, it has an unprecedented increase in recent years due to the rapid growth of Web 2.0. Today, blogging has been part of our life and it has direct influence to our communication and commercial development.

1.1 Blog Overview

According to the description in [6], blog can be defined as a type of website maintained by an individual that commonly displays regular entries of commentary, descriptions of events or personal diaries in reverse-chronological order. The blog entries usually contain comments which share reader's views and references to other blog posts. Each of these entries is called a blog post. The content of a typical blog post generally combines text, images and videos. It usually contains hyperlink or reference that links to related blog posts, web pages, and media. The websites that publish blog posts are termed as blog sites. The collective community of all these blog sites is known as blogosphere. The individuals who author the blog posts are referred as bloggers who freely express their opinions and emotion through blogging and interact with each other by contributing comments in response to specific blog posts.

Blogs rapidly gained in popularity and there has been a dramatic increase of content in blogosphere during the last couple of years. Based on Technorati's blogosphere report [37] in April 2007, which started tracking the development of blogosphere since 2002, it concluded that Technorati is now tracking over 70 million weblogs nowadays and there are about 120000 new

weblogs being established each day. In other words, there are around 1.4 new blogs and 17 blog posts are created in every second. Furthermore, they find that it only took around 320 days for the blogs growing from 35 to 75 million by March 2007. It is obvious that blog has become one of the most popular media of interaction among masses through the observation shown in Technorati's report. Huge amount of information is obtained in the blogosphere.

Besides the tracking report, researchers also conducted a study of the behavior of bloggers to analyze the blogosphere. In [29] [2], they pointed out that individuals tend to express their thoughts, voice their opinions, and share their experiences through blogging. The bloggers choose blogs as their communication media because blogs make it easier to author content independent of technical challenges of internet languages and scripts. By this reason, blogs successfully transform information consumers to producers and it also matches the main idea of Web 2.0. Through the blogging platform, millions of bloggers share their feelings and experience a sense of community, a feeling of belonging, and a bonding that members care with one another.

In conclusion, blogging opens up a new standard for information sharing and it is a significant component for the global

information and communication infrastructure nowadays. Blog content has become a valuable resource to data mining and the analysis of blog content is attracting much attention in research study during the recent years.

1.2 Motivation

Blogs are springing up in the recent years. Nowadays, writing and reading blogs has become part of our life. According to [23], Doc Searls indicated that blog will inform old media in the future and it will increasingly be a source of information that traditional media rely on. In this way, we anticipate that the growth of information in blogsphere will keep increasing and it will become the key source for our information needs.

Users are eager to extract useful information obtained in the blogosphere but few of them can effectively analyze it due to the free nature of blog content and the diversified blogging systems. Through our observation, there are many different types of blogs, such as personal blogs, corporate blogs, and blogs with specific interest or topic. Personal blog is the most common blog that usually reflects on individual's life or shares the view and comment with others. Corporate blog is for business purposes which is used to enhance the communication and culture in a

corporation or for marketing, branding and public relations purposes. Besides, some blogs specifically post and discuss certain news or share their view and experience on certain interest with others.

Due to the diversity of blog system, different blogs may have different format or even writing style that it is challenging to effectively analyze the blogs and gain useful information from blog posts. Thus, it forms the ground for the blog mining research and opens up new opportunities for developing blog-specific search and blog mining techniques.

1.2.1 Blog Mining

Today, there are lots of search and discovery tools for blogs which are offered by a variety of providers. Some focus on blog access, such as Blogscope[1], Blogpulse[2], and Technorati[3] while others such as Google[4], Yahoo![5] and MySpace[6] offer specialized blog services. Most of them provide the service of browsing and searching for blog posts by keyword queries. Nonetheless, their services can hardly satisfy our needs. Lot of human effort are

[1]http://www.blogscope.net/
[2]http://www.blogpulse.com/
[3]http://technorati.com/search/
[4]http://blogsearch.google.com/
[5]http://ysearchblog.com/
[6]http://blogs.myspace.com/index.cfm?fuseaction=blog.home

still needed for the blog searching purpose. According to [14], it pointed out that existing blog search engines only can compute and show the "authority" score for each blog and organize the blog posts into predefined categories. However, blog users may be interested in specific topic or the pulse of blogosphere rather than the popular blog posts with high authority. Unfortunately, current blog services cannot support this kind of searching. Furthermore, blogs are generally grouped by predefined categories instead of topics or events in the blog systems. Users cannot easily gain an overview of blog posts, nor do they receive enough support for finding potentially interesting blog entries and trend during the blog search. In order to extract useful information, such as significant topics, categories or trends in the blogosphere, people brought up the idea of blog mining [4] for this objective and lot of researches have been done on it. In [6], authors created a general framework for different tasks in blog mining as shown in Figure 1.1. The framework consists of a blog spider for monitoring and downloading content from multiple blog systems, a blog parser for extracting information from blogs, a blog content analyzer for classification and clustering the blog posts by text mining techniques, a blog network analyzer for applying network analysis to web structure mining, and a blog visualizer

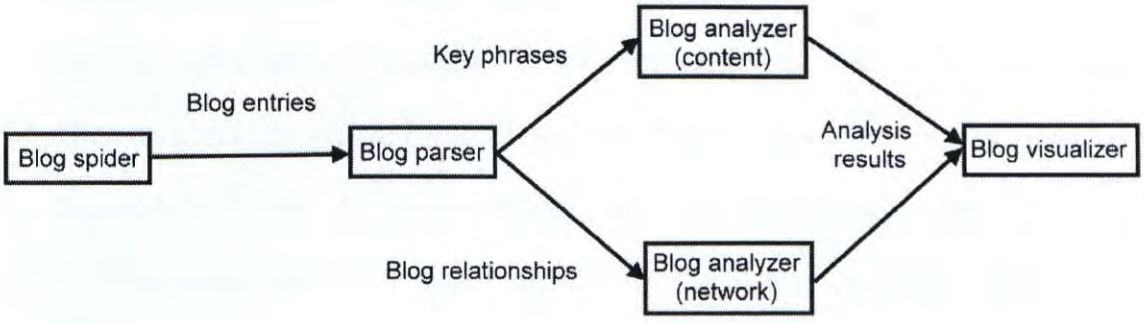

Figure 1.1: A general blog mining framework

to present content and network analysis results to users.

Since blogs have only recently become a subject of research, there are still many research issues for us to work on. This framework gives us a brief idea of blog mining. Based on our study on blog mining, most of the recent issues [2] mainly focused on the topic of blog network analyzing and blog visualization, such as blog clustering [24], community discovery and factorization [32] [8], opinion mining and social community visualization [22], etc. Little work has been done on blog content analysis due to the dynamic and chaotic blog content feature. It is not as straightforward to apply traditional web mining techniques [19] to the blog posts and it is generally believed that blog analysis is hard to be done only by the content in blog post.

Most of the researches tend to make use of the blog structure [17], such as hyperlink, comment and blog reference, to build up

a relationship network for their blog analysis. Nevertheless, the linking relation is bounded by the diversity of blog system, i.e. the users in the same blog system can only reference or comment to one another. In other words, the reference and comment is blog system oriented and this kind of linking relations cannot cross different blog systems. Though hyperlink can overcome this problem, actually there are only few hyperlink relations between blog posts that can be found because bloggers tend to use blog reference to link with each others.

In conclusion, the hyperlink and reference relation cannot effectively reflect the whole picture of global environment of blogosphere. Addressing these problems, we try to apply an unsupervised learning method for blog content analysis rather than blog structure analysis and evaluate the performance of our proposed clustering model compared with the traditional ones.

1.2.2 Topic Detection and Tracking

Besides blog content analysis, users are aware that the temporal information obtained in blog post is valuable for blog mining. It is more meaningful for us to retrieve the topic evolution rather than topic groups. That is because topic evolution can allow us to interpret the topic in different periods and find the significant

trend more easily. As discussed in [27], topic detection and tracking can monitor the news in order to spot new, previously unreported events and track the development of the previously spotted events. Nonetheless, the techniques of topic detection and tracking brought up in [27] are designed for news articles. Due to the different nature between news report and personal blog post, we need to develop another techniques specific for capturing the blog topic evolution. In this way, we extend our model from blog topic extraction to topic evolution tracking in our research.

After blog topic clustering, a Topic Evolution Graph will be built up by topic clusters and their related timestamp, then the topic evolution which contains the topic clusters in different time periods can be captured through our tracking method.

1.3 Objectives and Contributions

In our thesis, we propose an unsupervised learning model to extract blog topics in different time periods. The data set in our model is the blog entries extracted from MySpace. Our clustering model identifies the significant topic groups through concept clustering with Boundary Density-Based Clustering and document assignment. The topic clusters formed by concepts are

supported by both keywords and their related blog posts. With the topic clusters in different time periods, topic evolution can be extracted by building Topic Evolution Graph and capturing similar topic clusters in different periods which considers both content and temporal information.

Different from other existing research, our proposed model focuses on blog contents rather than the linking structures. Thus our model can overcome the problem of diversity of blog systems and applied to the blog posts extracted from different blog sites.

In addition, the topic clusters can be summarized by the related keywords and blog posts, so that we can grasp the brief outline of each topic more easily. Due to the idea of concept clustering, the clusters extracted from our model are based on semantics rather than document similarity.

Furthermore, we make use of the temporal information obtained in blog posts and capture the topic evolution through tracking the Topic Evolution Graph, which is a time-dependent and weight directed graph illustrated by related topic keywords. With the topic evolution, we can interpret the topic more easily and it is also feasible to our general needs.

1.4 Proposed Methodology

Our work aims at developing a novel model for clustering the blog entries into significant topics which can achieve high quality blog clusters. In addition, we arrange the blog topics by the temporal information, then build up the Topic Evolution Graph and extract significant topic evolution by tracking similar topics in different time periods.

The overall system design is illustrated in Figure 1.2. We first develop a blog crawler to extract blog posts from MySpace and identify their content as title, timestamp or article. Then do the feature generation and selection by tokenizing the blog content and selecting the representative keywords as shown in the first level of Figure 1.2. Afterward, we apply our proposed concept clustering model and get the concept clusters. Then we assign the blog entries to each concept cluster by measuring their similarities. Finally we get the blog clusters supported by both concepts and blog posts as shown in the second level of Figure 1.2. Furthermore, we build up a Topic Evolution Graph for organizing blog topics in different periods and design a tracking method to capture the significant topic evolutions as shown in the third level of Figure 1.2.

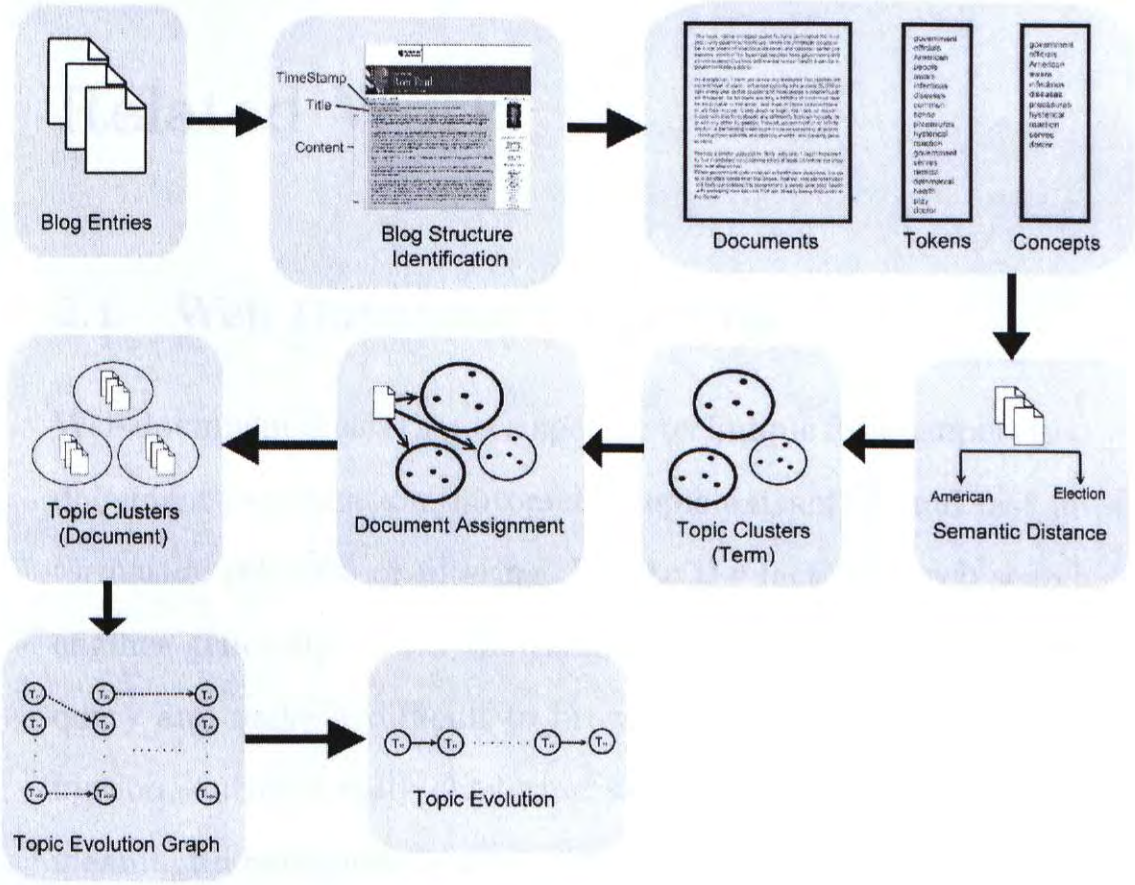

Figure 1.2: Our Proposed System Design

Chapter 2

Related Work

2.1 Web Document Clustering

Web document clustering is a specific technique for unsupervised document organization, automatic topic extraction and fast information retrieval or filtering. Due to the fact that web search engines generally return thousands of pages in response to the query and make it difficult to browse or identify relevant information, automatically clustering web document collection into meaningful categories is necessary for easy browsing and usage. In addition, document clusters can overcome the scalability issue with information retrieval because documents with similar topic can be retrieved together to reduce the query response time. The performance of searching and retrieval can also be improved.

In [13], some existing document clustering methods, such as hierarchical clustering and partitional clustering, have been introduced. Besides the traditional clustering methods discussed in [13], some researchers try to develop different techniques for the document representation in order to achieve better performance. According to [46], Zamir and Etzioni introduce Suffix Tree Clustering (STC) which does not treat a document as a set of words but rather as a string. It is feasible to web documents and has better performance than the standard clustering methods. In [34], Diego takes the ANNIE capabilities into account. He presented both WordNet lexical categories (WLC) and WordNet ontology (WO) for the creation of a low and well-structured vector space for document clustering.

Based on our study, most of the typical clustering methods apply document vector model for the presentation of documents which is feasible to the standard web documents, such as news articles or scientific articles. However, the target of our clustering is the blog entries which are different from the standard web documents. The content of blog entry is so diversified that it usually does not have significant topic and format. For example, bloggers usually post their diaries and share their own opinions on their blogs. These kinds of documents generally

have grammatical mistakes and misspellings. In addition, there may be many kinds of writing style even though they are talking about the same thing. Therefore, the document similarity measurement and even the performance of the clustering will be affected if we apply existing methods for blog clustering. In this way, specially designed techniques are needed specially for the blog clustering.

2.2 Document Clustering with Temporal Information

Lots of web documents, such as news articles and blog posts, are tagged with timestamp. This kind of temporal information can be applied in web document clustering for improving the performance. A research program called topic detection and tracking (TDT) is relevant to this approach. The program is organized by National Institute of Standards and Technology (NIST). TDT is one of the techniques that applied temporal information for organizing on-line documents like broadcast news based on the notions of events and topics. The general idea of TDT is to develop core technologies for news understanding. Specifically, TDT systems discover the topical structure in

unsegmented streams of news reporting across multiple media and keep track of topics in a constantly expanding collection of multimedia stories. TDT can be divided into topic detection for the extraction of clusters containing stories with same topic, and topic tracking to keep track of stories with similar topic or describing the same event. TDT is closely related to our work, but our approach is developed based on blog content rather than news articles.

According to [27], Juha proposes a novel topic definition which allows the topic to evolve into several directions and presents a TDT system with dynamic hierarchies that can cut down the excessive computation. Besides, some other techniques were bought up in recent years. Mei and Zhai in [28] study a particular Temporal Text Mining (TTM) task which discovers and summarizes the evolutionary patterns of themes in a text stream but their methods are just applied to the domains of news articles and literatures. In [33], authors define an Event Timeline Analysis (ETA) task to automatically organize news events by time order and present the event evolution using graph structure. Their proposed Event Evolving Graph (EEG) incrementally updates the process of events so that it can better fit the feature of news streams information. In [44], Yang, Shi and

Wei utilize the temporal relationship, event similarity, temporal proximity and document distributional proximity to identify the event evolution relationship. An event evolution graph is built to present the structure of events for efficient browsing and extracting information. Considering the problem of dynamic analysis, [41] introduces the framework for identifying topic evolution based on topic groups. So that it can analyze the topic dynamic change and impact. Nonetheless, most of the research work developed for the temporal information mainly applied to news streams rather than blog contents. The developed techniques may not be feasible to the blog contents due to the special content nature.

2.3 Blog Mining

In the view of information retrieval, blog clustering refers to a kind of web content clustering which focuses on the discovery of useful information by dividing web documents into different groups. The objective of blog clustering is to separate a given set of blog entries into different clusters such that items in the same cluster are related to a certain topic, event, or category.

Typical techniques of web content clustering [34] [46] [13] are not feasible to the blog entries because they generally applied

to the formal documents which have strong indication of topics. Typical clustering models assume that most of the words in the document are well-formatted and the style of their writing are also formalized, so that they can work well in this way. On the contrary, the format of blog entry is so free that bloggers can write anything in their own words on their blogs. The words they used and the structures of the blog content are not rigorous, so that traditional clustering methods no longer work in these documents. Thus, we need specialized mining techniques for blog content.

Regarding to these problems, some recent research has been done on blog mining. According to [40], Wang proposes a hierarchical SVM model for blog clustering and implements the model in their own blog database. In [36], authors describe a method to detect topic words from blog entries by interest and apply the topic words to support their blog clustering model. Authors in [1] propose to cluster blog sites by employing some form of collective wisdom, such as the information in blog catalog site and label information from individual bloggers. In [7], authors focus on business blogs and propose probabilistic models for blog search and mining using two machine learning techniques, latent semantic analysis (LSA) and probabilistic latent semantic

analysis (PLSA).

In analyzing the space of weblogs, some research works are done on the blog evolution besides blog clustering. In [21], authors developed a new tool to address the evolution of hyperlinked corpora. They defined time graphs to crystallize the notion of community evolution. Authors in [39] used information retrieval techniques to associate blog entries to topics and visualized the collected information in terms of a topic map. In [5], authors presented efficient algorithms to identify keyword clusters in large collections of blog posts for specific temporal intervals. Their work can be separated into two parts: generating the keyword clusters and identifying stable clusters.

Though lot of research has been done on blog mining in recent years, it is still a novel idea that we try to extract the topic clusters and topic evolution supported with related keywords and capture the blog entries as what we did in our research.

Chapter 3

Feature Extraction and Selection

Blogs are known to be semi-structured because they are a kind of web documents in HTML (Hyper-Text Markup Language), which provides a means to describe the structure of text-based information by tagging. HTML facilitates the design of document layout and provides a means of referencing from one document to another. Since HTML specifies the layout of document, it cannot structure the content of a document, i.e. document elements are not organized according to a certain schema and no semantics can be extracted from HTML designed documents. Hence blogs are regarded as "semi-structured" and the content structure cannot be specified in blog entry. Specific techniques are needed in order to extract useful information in blog entries.

Prior to the blog clustering, we have to analyze the structure of blog posts, extract useful content, and convert the content into structured form. Thus, the techniques of content extraction and feature selection are applied in the preprocessing stage of our model. In this chapter, we describe our crawler collecting the blog entries from MySpace[1], then identify and extract the content part from each blog entries, develop a tokenizer to tokenize the blog articles into words, ignores the stop words, reduces the duplicated stemming words and eliminates the noisy words by heuristic methods.

3.1 Blog Extraction and Content Cleaning

Blog content extraction is to derive information, such as blog content, from blog posts and standardize them to be our data source. Nevertheless, we face the challenge that thousands of blogs are published everyday but no standard format is defined for the blog pages. The content of blog post is unstructured and its style is so various that different blogs may have totally different settings. Thus it is hard to extract the content part from blog posts.

Neglecting the diversity of the blog format, a blog entry gen-

[1]http://blogs.myspace.com/index.cfm?fuseaction=blog.home

erally contains four parts, i.e., title, author, content, and timestamp that we can easily identify from each blog entry. In our model, we first build up a blog crawler to collect the blog entries from MySpace and extract these four kinds of content information from each blog entry.

3.1.1 Blog Parsing and Structure Identification

Our blog crawler identifies the blog structure by the specific tags in each blog entry. Since different blog systems apply their self-developed tags for their blog content tagging, we need to develop different identification rules for different blog systems. In our model, we establish a identification rule that is feasible to the blog entries from MySpace - the largest social networking website in the world. In Figure 3.1, it shows an example of the standard HTML structure of blog entry in MySpace. We find that there are some specific tags, such as , , and , for tagging the time, title, category and content respectively. By the tagging information, our blog crawler applies the given tagging rules and identifies the content information from blog entries. It collects the blog entries, analyze the HTML structure of each blog entry

```
...
<div class="blogTimeStamp">
Tuesday, May05, 2009
</div>
...
<div class="blogSubject">
<label id="pBlogSubject">
My Obama Review
</label>
<br/> Current mood: contemplative <br/>
<b>Category:</b>
News and Politics
</div>
...
<div id="pBlogBody" class="blogContent">
<p> //blog content </p>
</div>
...
```

Figure 3.1: An example of the HTML code of blog site

by tagging, parse the content, and store the content information to be our data source.

After the content is extracted, we have to break the character stream into words or tokens in order to identify the useful features from blog content. It is hard to extract higher-level information from the documents without identifying the tokens. In our model, we extract the structured information by identify-

ing the tokens from blog content. In the parsing step, we develop a tokenizer to delimit the character streams by space and punctuation marks, such as ".", "!", "?", ",", etc., and organize the tokens in structured way. Some specific characters, such as tab and newline are also assumed as delimiters. In addition, The tokenization process is language oriented due to the nature of different languages. We have to develop different rules if we want to handle the documents with different languages. Thus we only focuses on handling the documents written in English in our model.

3.1.2 Stop-word Removal

Once a character stream is segmented into a sequence of tokens, the next step is to convert each tokens to a standard form. First, stop words need to be identified and ignored from the tokens. Stop words are defined as the common words that do not provide any useful information, such as "the", "a", "and", "or", etc. It is useful to discard these words, otherwise they may mislead the clustering process by including frequent words that are not informative. Our clustering model is developed based on concepts extracted from the blog content, thus stop words will affect the performance of our clustering. Therefore, we compile

a stop word list as shown in the Appendix and eliminate the stop words by checking with the given list.

3.1.3 Word Stemming

Besides the stop words, we also have to consider the problem of stemming. Word stem is the part of word that is common to all of its inflected variants. For example, the stem of verb "waits", "waited" and "wait" is "wait". Word stemming is the process for converting the inflected words to their stem and eliminate the duplicated ones. This is essential to avoid treating different variations of a word distinctly. However, no algorithm can perfectly handle the stemming problem and completely transform all the verbs to their stem unless there is a stemming dictionary for us to check with. In our model, we apply Porter Stemmer Algorithm [30] for stemming reduction because of its high performance and efficiency. It can correctly identify a significant number of stems and it is also very efficient.

3.1.4 Heuristic Content Cleaning and Multiword Grouping

Besides the process of stop-word removal and word stemming, we also apply some heuristic methods for content cleaning. For

example, we set threshold to remove the extremely rare or common words which usually cannot represent the key feature of document.

In addition, there are cases that it is useful to consider a group of words as feature. For this reason, we extend our feature extraction model from a word to a phase with two adjacent terms, such as "president obamas", "wall street", "social security", "United States", etc. The performance of feature extraction and selection will have significant improvement because multiword features are often highly predictive to the topic. In our feature selection model, we add extra feature weight to the multiword in order to extract the meaningful phases to be our concept.

3.2 Feature Selection

The root of clustering problem lies in the extraction of the most representative features. The extracted feature have to be informative enough to represent the document content being analyzed. Otherwise, the clustering result will be misled by the non-informative features. Moreover, it is important to reduce the feature dimension due to its impact on the scalability of clustering algorithm. So Feature Extraction and Selection is

performed in our clustering model.

Feature Extraction and Selection is the process that chooses a subset from the original features according to certain criterions and transform into a reduced representation set of features. The selected features retain original meaning and provide a better understanding for the data and learning process.

Some comparative studies [25] [26] showed that quite a lot of feature selection methods have been developed in text mining, such as Document Frequency (DF), Term Contribution (TC), Term Frequency Inverse Document Frequency (TF-IDF), and Term Strength (TS). The evaluation described in [26] concludes that TC is the preferred unsupervised feature selection method for text clustering due to its low computation complexity and good performance, and TF-IDF is the most popular and common method adopted in the feature extraction. Therefore, we apply TC and TF-IDF for the feature extraction and selection in our clustering model. We will discuss them in the following section.

3.2.1 Term Frequency Inverse Document Frequency

Considering the problem of blog clustering, the characteristic features of blog contents are the tokens they contain. Without

deep analysis of the linguistic content of the blog articles, we propose to describe the documents by the features that represent the highest TF-IDF tokens.

TF-IDF (Term Frequency Inverse Document Frequency) can identify the perceived importance of a word and the formulation shown in Equation 3.1 is used to compute the term weight. Each term is weighted based on its TF-IDF corresponding to each blog article. The TF-IDF assigned to a term is its normalized term frequency modified by a scale factor for the importance of that term. The normalized term frequency is given in Equation 3.2 and the scale factor is called the inverse document frequency, which is given in Equation 3.3. It simply checks the number of documents containing that term and reverses the scaling. So that a term is considered less important and is scale is lowered if it appears in many documents. When the term is relatively unique and appears in few documents, the scale factor zooms upward because it is likely important.

$$F(t_i, d_j) = \gamma(t_i, d_j) \cdot \delta_{t_i} \tag{3.1}$$

$$\gamma(t_i, d_j) = \frac{n_{t_i,d_j}}{\sum_k n_{t_k,d_j}} \tag{3.2}$$

where

n_{t_i,d_j}: the number of occurrences of term t_i in document d_j
$\sum_k n_{t_k,d_j}$: the number of occurrences of all terms in document d_j

$$\delta_{t_i} = \log \frac{N}{|\{d_j : t_i \in d_j\}|} \tag{3.3}$$

where
N: the total number of documents in the corpus
$|\{d_j : t_i \in d_j\}|$: the number of documents where term t_i appears

Feature selection routines attempt to select a subset of words that appear in each document to have the greatest potential for topic prediction. In our model, we choose the highest TF-IDF tokens from each document to be the feature which are highly predictive to the documents and topics.

3.2.2 Term Contribution

Depending on different purposes, feature selection can be supervised or unsupervised. Our concept clustering model applies the unsupervised feature selection methods that select a subset of important feature for the concept selection.

In our feature selection model, we define the overall contribution of a specific term to the set of documents as the selection criterion of concept. We introduce a feature selection method called "Term Contribution" (TC) [25] [26] [11] that takes the term weight into account. The definition of TC is given by Equa-

tion 3.4 which defines that the contribution of a term is based on its overall contribution to the document similarities. With the measurement of Term Contribution, we select the concepts by choosing the terms with highest Term Contribution value.

$$TC(t_k) = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} F(t_k, d_i) \cdot F(t_k, d_j) \tag{3.4}$$

where

$F(t_k, d_j)$: The TF-IDF of term t_k in document d_j

N: The total number of documents

Chapter 4

Blog Topic Extraction

This chapter presents the clustering methods for blog topic extraction. We first introduce the standard clustering algorithms that are generally applied to document clustering. The properties and limitations of these clustering algorithms are also discussed.

There is a multitude of clustering techniques proposed in the literature [38] [16] [12], each of them adopts a certain strategy for detecting the grouping of data. However, there is a variation on the requirements of each algorithm, such as data representation, cluster model, and similarity measurement. These requirements more or less affect the performance of the clustering algorithms and make it infeasible to blog clustering. Addressing these problems, we develop the idea of concept clustering and Bounded Density-Based Clustering which specifically targets on

blog clustering.

4.1 Requirements of Document Clustering

Before analyzing different clustering algorithms, we first define some of the properties that is required by the clustering models.

4.1.1 Vector Space Modeling

The feature tokens form the basis for creating an array of numeric data corresponding to the document collection. Nowadays, Vector Space Model (VSM), introduced by Salton in 1975 [35], is the most popular method in representing documents in clustering systems. In vector space modeling, each document is represented by a vector d in the term space, $d = \{tf_1, tf_2, \ldots tf_n\}$, where tf_i is the term frequency in the $i - th$ document.

From the document collection, we create the Term-Document Matrix $d[i, j]$[1] with featured tokens as dimension and documents as vector. It represents natural language documents as matrix and make it possible to process them as a whole. To represent every document with the same set of terms as dimensions, we have to extract the terms found in the documents and treat them as our feature vector. Due to the problem of scalability, we sort

[1] $d[i, j]$ represents the TF-IDF value of term j occurred in document i

the tokens by their TF-IDF and extract the top 10 percent of tokens from each document to form the term space.

In our Term-Document Matrix, we regard documents as rows and terms as columns. The value of each column is the TF-IDF corresponding to each term.

4.1.2 Similarity Measurement

Given the Vector Space Model representing the documents, another key factor in document clustering is the similarity measurement between documents. In order to group similar documents together, we have to define the proximity metric to find which documents are similar. There are a large number of similarity metrics reported in the literature and we review the popular ones in the following.

The similarity between two objects is measured through distance functions. A common distance function is known as the family of Minkowski distance, i.e. given two feature vectors x and y representing two objects, the similarity between them can be measured as follows:

$$Minkowski(x, y) = \|x - y\|_p = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p} \qquad (4.1)$$

The distance function actually describes an infinite number of distances indexed by p, which assumes that its value is greater than or equal to 1. Some common values of p and their respective distance function are:

$$Hamming(x, y) = \|x - y\|_1 = \sum_{i=1}^{n} |x_i - y_i| \tag{4.2}$$

$$Euclidean(x, y) = \|x - y\|_2 = \sqrt[2]{\sum_{i=1}^{n} |x_i - y_i|^2} \tag{4.3}$$

Another common similarity measure used in document clustering is the cosine correlation which is defined as:

$$COSINE(x, y) = \frac{x \cdot y}{\|x\| \, \|y\|} \tag{4.4}$$

In our clustering algorithm, the cosine correlation and Euclidean Distance are adopted in our model. We also evaluate their performance and discuss their limitation in the clustering model.

4.2 Document Clustering

After the introduction of representation model and distance function, this section presents an overview of document clustering algorithms[42].

Document Clustering is an unsupervised process that groups documents with similar content. This kind of problem is also known as automatic document classification. It mainly concerns with text mining that helps to organize documents into similar groups without labeled training instances.

Clustering has been studied in the literature that applies the model of word co-occurrence, which is applicable to text classification problems. The clustering model using word co-occurrences results in groups of documents that contain overlapping sets of words. Existing clustering algorithms[38] can be broadly classified into partitional and hierarchical clustering that will discuss in the following sections.

One of the applications of document clustering is cluster-based retrieval, which is a method for improving the speed and effectiveness of document retrieval. When request is posed to the pre-clustered document collection, the documents that fall into the clusters related to the request are returned. In addition, if the collection is pre-clustered, the search will be faster by searching cluster prototypes instead of documents in the collection. Assuming that the contents of the documents in a cluster are related, returning the documents from clusters closest to the user's request should have the effect of improving the number of

relevant documents returned.

4.2.1 Partitional Clustering

Partitional clustering is a division of the set of data object into K non-overlapping partitions. Its goal is to optimize a certain criterion function, such as square-error criterion.

K-means clustering is one of the most frequently used clustering methods for document clustering. It is a prototype-based partitional clustering technique that aims to divide a data set into a given number of spherical shaped clusters. Each cluster is represented by its center point considering all the data point in the same cluster. Proceeding from starting solution in which all documents are distributed on a given number of clusters, it keeps trying to improve the solution by a specific change of the allocation of documents to the nearest cluster until the stop condition is met. The goal of the allocation is to minimize the objective function Z given in Equation 4.5:

$$Z = \sum_{j=1}^{K} \sum_{i=1}^{n} \left\| p_i^{(j)} - c_j \right\|^2 \tag{4.5}$$

where

$\left\| p_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between data point $p_i^{(j)}$ in cluster j and its related center point c_j.

1. Place K points into the space represented by the objects that are being clustered. These points represent the initial centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been allocated, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until all centroids no longer move. This produces a separation of objects into groups from which the metric to be minimized can be calculated.

Figure 4.1: K-means Clustering Algorithm

The cluster center c_j is an indicator of the distance of the n data points from their respective cluster centers. The algorithm is composed of the steps described in Figure 4.1:

4.2.2 Hierarchial Clustering

Hierarchical clustering produces a nested sequence of partitions, with a single all-inclusive cluster at the top and singleton clusters of individual objects at the bottom. The set of nested clusters are organized as a tree and each node is the union of its children.

Agglomerative Hierarchical Clustering (AHC) is commonly discussed in the hierarchical clustering algorithms. It starts with the disjoint set of clusters, which places each input data point in an individual clusters. At each step, pairs of items or singleton clusters are merged. This merging process is repeated until the number of clusters reduces to the user-specified number. The basic steps of the AHC are described in Figure 4.2:

1. Compute all pairwise document similarities.
2. Place each of the N documents into its own cluster.
3. Form a new cluster by combining the most similar pair of current clusters i and j.
4. Update the similarity matrix by deleting the rows and columns corresponding to i and j.
5. Calculate the entries in the row and column corresponding to the new cluster resulting from the merge of i and j
6. Repeat steps 3, 4 and 5 until one cluster remains.

Figure 4.2: Hierarchical Agglomerative Clustering Algorithm

AHC works in a greedy manner that the pair of document group chosen for agglomeration is the most similar ones under the given distance function.

For the similarity measurement, different distance functions are applied, such as Single Linkage Method[2], Complete Linkage Method[3], and Average Linkage Method[4].

4.2.3 Density-Based Clustering

Some researchers suggested that a cluster can be defined as a dense region of objects which is surrounded by a region of low density. By this idea, Density-Based Clustering is developed

[2]For two cluster S and T, the distance is $\|T - S\| = \min_{\substack{x \in T \\ y \in S}} \|x - y\|$

[3]For two cluster S and T, the distance is $\|T - S\| = \max_{\substack{x \in T \\ y \in S}} \|x - y\|$

[4]For two clusters S and T, the distance is $\|T - S\| = \frac{\sum_{\substack{x \in T \\ y \in S}} \|x-y\|}{|S| \cdot |T|}$

to locate regions of high density that are separated from one another by regions of low density. Lots of work have been done on the Density-Based Clustering and they brought up different views of the density function.

In the Density-Based Clustering algorithms, DBSCAN [10] is a simple and effective method that produces partitional clustering. It automatically determined the number of clusters by the algorithm and the points in low-density regions are classified as noise and omitted. DBSCAN assumes that the density in the area of noise is lower than that in the area of useful data. Thus, the algorithm defines the area with high density as clusters and separate them from low dense noisy area. It works well in discovering cluster with arbitrary shape and filters the noise object effectively.

DBSCAN requires two parameters - the maximum radius of neighborhood (Eps) and the minimum number of points in the Eps-neighborhood (MinPts). The algorithm starts with an arbitrary starting point that has not been visited. Then finds all the neighbor points within distance Eps of the starting point. If the number of neighbors is greater than or equal to MinPts, a cluster is formed. The starting point and its neighbors are added to this cluster and the starting point is marked as visited. The

algorithm repeats the evaluation process for all neighbors recursively until all the points are visited or cannot find any other unvisited points. If the number of neighbors is less than MinPts, the point is marked as noise.

DBSCAN can effectively detect noises and handle clusters with arbitrary shapes and sizes. Nevertheless, it suffers from the problems of varying densities and high-dimensional data. However, some researches have been done based on DBSCAN, such as FDBSCAN [20], DBCLASD [43] and OPTICS [3], describing about the improvement of DBSCAN, but most of them are focused on the efficiency issue rather than the performance of its result.

4.3 Proposed Concept Clustering

Typical document clustering methods adopting VSM relies on the notion of feature vectors mapped into n-dimensional vector space. Nonetheless, the similarity measurement between vectors are affected by the nature of blog content. For instance, though two blog posts are discussing the same topic, their similarity may be very low because bloggers tend to use different words for their description. Besides the writing style, we also need to consider the misspellings or typos in blog content according to

our observation. It is hard to define the similarity of documents by measuring the terms appeared in blog entries.

However, some sophisticated representations are being investigated which includes more structured semantic models. For example, the use of ontology was suggested in [15], where related terms such as synonyms can be aggregated resulting in a reduced document representation space. The resulting representations are much more oriented on concepts rather than term relationship.

In addition, the authors in [18] present a clustering scheme on the basis of ontology-based distance measure. The term mutual information matrix suggested in their work is calculated with the aid of Wordnet and some methods of learning ontology from textual data.

As many recent research works suggested the use of ontology information, we can see that the document clustering based on semantic information is more feasible to the web documents and blog posts than the traditional VSM.

Different from the existing works, we propose the concept clustering which is developed base on the assumption that similar keywords are used in blog content for the description of same topic. Due to the nature that bloggers tend to use sim-

ilar keywords to describe the same topic, some keywords will be appeared repeatedly in similar blog content. For instance, if a blogger is discussing about American Election in the blog post, some terms, such as "obama", "barack", "john", "maccain", "campaign", "election", "vote", will possibly be appeared in the blog content. In this way, we can classify the blog entry topic by these representative keywords instead of analyzing the whole content of the blog entries.

In the concept clustering model, we cluster concepts instead of documents. A concept is defined as a representative keyword which can contribute to some topic descriptions. After concepts are determined, we apply our proposed concept clustering to get the result cluster set. We call each result cluster as topic or topic cluster because it is formed by the concepts which are usually describing similar topic, such as "IRAQ PROBLEM", "AMERICAN ELECTION", "FINANCIAL PROBLEM", etc.

In our model, the concepts are selected by TF-IDF and TC described in Chapter 3. We propose the Boundary Density-Based Clustering with χ^2 measure for the clustering of concepts. After concept clustering, we get a set of topic clusters which is summarized with a set of concepts. With the terms in each cluster, we assign blog posts into their related cluster through

Jaccard Similarity Coefficient measurement.

Concept Clustering solves the problems of misspellings and grammatical mistakes because the concepts extracted in our model are representative for topic description. In addition, documents are grouped together by keywords describing similar topic, so that the clustering results will not be affected by the irrelevant content or words. Only significant topics will be extracted due to the feature selection and the nature of Density-Based algorithm. Furthermore, we also allow that a blog post can be assigned into more than one topic clusters because bloggers usually discuss various topics in the blog posts.

4.3.1 Semantic Distance between Concepts

For concept clustering, we have to define the semantic distance between concepts. In order to evaluate the suitable methods of ranking the dependency of term pairs, three statistical methods were implemented in our model: Associated term weight [44], Mutual Information, and χ^2 statistic [45], which are described as follows:

The χ^2 statistic measures the dependence between each pair of terms. In the measure, there are 4 types of event output about the occurrence of two terms in the same document or

paragraph which is shown in Table 4.1. Our χ^2 statistic given in Equation 4.7 is derived from the basic χ^2 test which is shown in Equation 4.6 with 4 as its degree of freedom. The χ^2 statistic has a natural value of zero if two terms are independent.

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i} \tag{4.6}$$

where

O_i = an observed frequency of an event
E_i = an expected frequency of an event
n = the number of event

	term c occur	term c not occur
term t occur	A	B
term t not occur	C	D

Table 4.1: Event Matrix

$$\chi^2(t_t, t_c) = \frac{N \cdot (A \cdot D - C \cdot B)^2}{(A + C) \cdot (B + D) \cdot (A + B) \cdot (C + D)} \tag{4.7}$$

where

A: the number of times term t_t and term t_c co-occur in the same block
B: the number of times term t_t occurs without t_c in the same block
C: the number of times term t_c occurs without t_t in the same block
D: the number of times neither term t_c nor term t_t occurs in the same block

Mutual Information measures the mutual dependence of terms. It is commonly used in statistical language modeling of word associations. The formula of the mutual information between term t and term c is given in Equation 4.8:

$$MI(t_t, t_c) = \log \frac{P(t_t \wedge t_c)}{P(t_t) \cdot P(t_c)} \tag{4.8}$$

where

$P(t_t \wedge t_c)$: the probability that both term t_t and term t_c co-occur in a document

$P(t_t)$: the probability that term t_t occurs in a document

$P(t_c)$: the probability that term t_c occurs in a document

The associated term weight measures the relevant weights with term t_t and term t_c. It divides the co-occurrence weight between term t_t and term t_c by the occurrence frequency of term t_t. The weighting factor is used to penalize some general terms. The formulation is given in Equation 4.9:

$$Weight(t_t, t_c) = \frac{\sum_{i=1}^{N} m(d_i, t_t, t_c)}{\sum_{i=1}^{N} d(d_i, t_t)} \cdot WeightingFactor(t_c) \tag{4.9}$$

and

$$m(d_i, t_t, t_c) = min(n_{t_t,d_i}, n_{t_c,d_i}) \cdot \ln \frac{N}{|\{d_j : t_t \in d_j \cap t_c \in d_j\}|} \tag{4.10}$$

$$d(d_i, t_t) = n_{t_t,d_i} \cdot \ln \frac{N}{|\{d_j : t_t \in d_j\}|} \tag{4.11}$$

$$WeightingFactor(t_c) = \frac{\ln \frac{N}{|\{d_j : t_c \in d_j\}|}}{\ln N} \tag{4.12}$$

where

n_{t_t,d_i}: the occurrence frequency of term t_t in document d_i

$min(n_{t_t,d_i}, n_{t_c,d_i})$: the minimum between term frequency of term t_t and term t_c in document d_i

$|\{d_j : t_t \in d_j\}|$: the number of documents containing term t_t

$|\{d_j : t_t \in d_j \cap t_c \in d_j\}|$: the number of documents containing both term t_t and term t_c

N: the total number of documents

After the implementation of these three statistic measurements, we find that the result of χ^2 statistic outperforms the others because its result is much more meaningful than the others as shown in the Appendix. In the following, we simply discuss why the result of χ^2 statistic is the best.

The associated term weight is asymmetric, i.e. the value of Weight(t_t, t_c) and Weight(t_c, t_t) may be different. This measurement is not feasible to our model because the distance between two terms needs to be symmetric.

The weakness of mutual information is that the score is strongly influenced by the marginal probabilities of terms, i.e. rare terms will have higher score than common terms.

Compared with mutual information, χ^2 has better performance because it is a normalized value and takes the absence of a term into account. According to the result shown in [31],

the performance of χ^2 statistic is also much better than MI. So we choose χ^2 statistic as the distance function of concept in our model.

Given a collection of feature words from blog entries, we define that terms are dependent if they tend to occur either in the same document or in the same paragraph. In our distance function, we consider the term dependency not only just in document level, but also in paragraph level. Extra weight will be added if terms are co-occur in the same document and paragraph. So our proposed χ^2_{whole} statistic measure the similarity between each pair of tokens by the weighted average of χ^2_p and χ^2_d given in Equation 4.13.

$$\chi^2_{whole}(t_t, t_c) = \beta \cdot \chi^2_p(t_t, t_c) + (1 - \beta) \cdot \chi^2_d(t_t, t_c) \qquad (4.13)$$

where

χ^2_p: the χ^2 static of term t_t and term t_c corresponding to paragraph
χ^2_d: the χ^2 static of term t_t and term t_c corresponding to document
β: the factor for tuning the weight between χ^2_p and χ^2_d

4.3.2 Bounded Density-Based Clustering

In Bounded Density-Based Clustering, we define the density by two parameter: Eps and MinPts, and form the initial cluster by the given density. Then expand the cluster by considering all

the neighbors in the initial cluster and use a boundary function to check if the expansion is within the defined boundary.

In Figure 4.3, 4.4 and 4.5, it shows the expansion of Cluster 1 after it is formed by point C with given Eps and MinPts.

When Cluster 1 is expanded by checking point B which is the neighbor point of point C, we find the unclassified point P which is density connected to point B as shown in Figure 4.3. Before we assign point P to Cluster 1, we have to check the average similarity between point P and all the medoid points in Cluster 1 as shown in Figure 4.4. If the average similarity is within the given Boundary, Point P will be assigned to Cluster 1 as shown in Figure 4.5.

The expansion is iterated by rounds and the Boundary will be released by boundary function in each round. By considering the boundary in each expansion, all entities within the cluster will be related to each other in certain degree.

For Boundary Density-Based Clustering, three types of points and the boundary function are defined:

Core point: A point is a core point if the number of points within the given neighborhood around the point determined by the distance function and distance parameter (Eps) exceeds the threshold value (MinPts) which is a user-specified parameter.

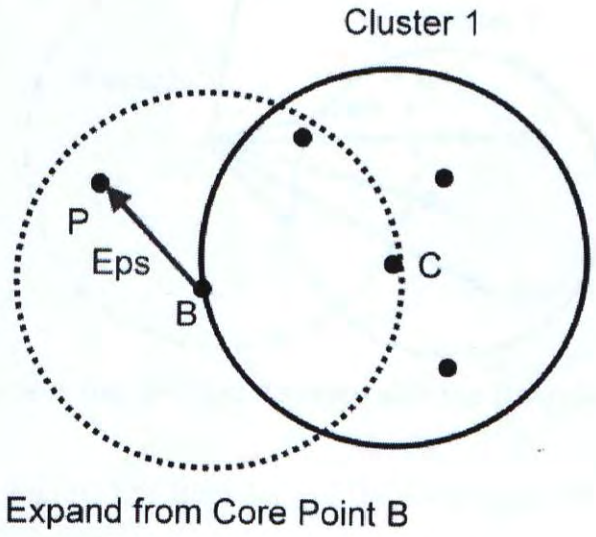

Figure 4.3: Boundary of the Expansion I

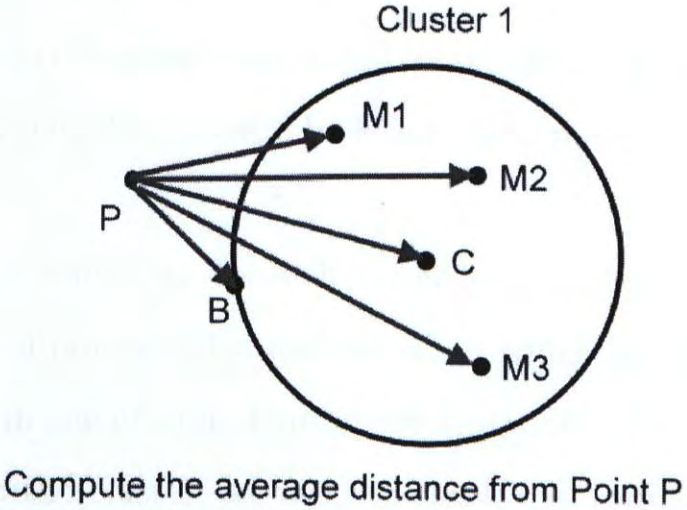

Figure 4.4: Boundary of the Expansion II

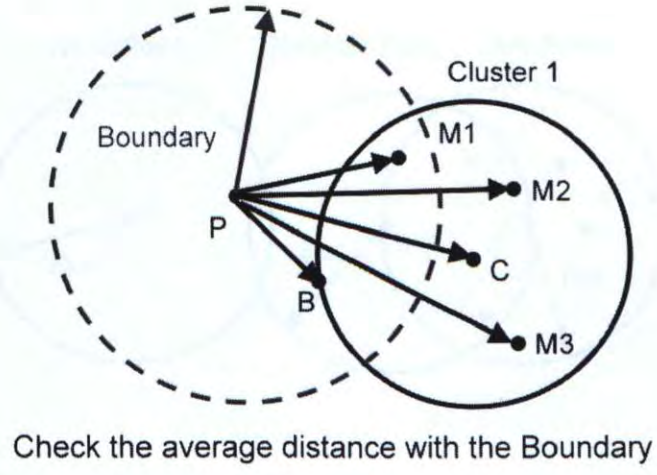

Figure 4.5: Boundary of the Expansion III

In Figure 4.6, point A is a core point.

Neighbor point: A neighbor point is the point that falls within the neighborhood of core point. In Figure 4.6, point B is the neighbor point of point A.

Noise point: A noise point is any point that is neither a core point nor a neighbor point. In Figure 4.6, point C is a noise point.

Boundary function: In each expansion of cluster, we first define a set of points called medoid set m which includes all the core points in the cluster. During the expansion, if we assigned a point p to the cluster, we need to check the average distance between point p and all the points m_i in the medoid set by the boundary function as shown in Equation 4.14. Point p will be

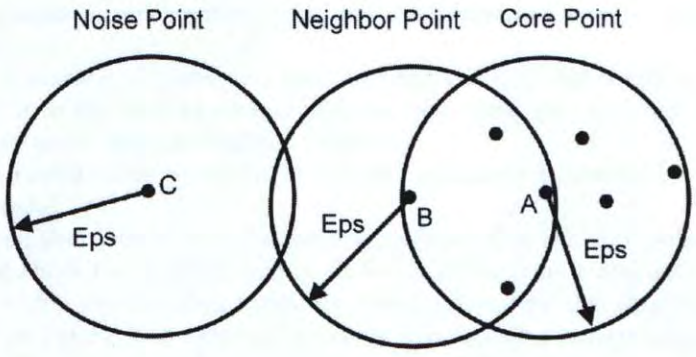

Figure 4.6: Core, Neighbor, and Noise points

assigned to the cluster only if it is within the given boundary.

$$AveDis = \frac{\sum_{i=1}^{|m|} dis(p, m_i)}{|m|} \tag{4.14}$$

$$Boundary = T \cdot \alpha^{round} \tag{4.15}$$

In Equation 4.15, the boundary function is determined by three parameters: T, α and $round$. T is a user-specified value as the input of Boundary Density-Based Clustering, α is defined to be the release factor of boundary, and $round$ is the value of expansion iteration.

With the definitions of core point, neighbor point, noise point and threshold function, Boundary Density-Based Clustering can be described by the steps in Figure 4.7:

1. Randomly choose an unclassified point and check whether it is a core point by given Eps and MinPts.
2. The point is marked as noise and back to Step 1 if it is not a core point. If it is a core point, add it to the medoid set and initialize the round with 0. Then form the initial cluster with core point and its neighbor points.
3. Update the *round* value by adding 1 and the boundary threshold by the given T, α and updated *round*.
4. Check the neighbor points with the density function. If it is a core point, add it to the medoid set and check its neighbor points. If its neighbor point's average distance to the medoid set is within the updated boundary threshold, assign this neighbor point to the formed cluster and store it as neighbor point for the checking in next round.
5. Repeat Step 3 to Step 4 until no neighbor point can be found.
6. Repeat Step 1 to Step 5 until there is no unclassified point.

Figure 4.7: The steps for Boundary Density-Based Clustering Algorithm

The basic version of Boundary Density-Based Clustering is shown in the Algorithm 1. SetOfPoints is the whole data set of our model containing the terms extracted from blog content. The terms are marked as UNCLASSIFIED at the beginning. Eps and MinPts are the global density parameters determined manually. The function SetOfPoints.get(i) returns the i-th element of SetOfPoints.

The main part of Boundary Density-Based Clustering is the expand function which is presented in Algorithm 2. SetOfPoints.regionQuery(Point, Eps) returns the Eps-Neighborhood[5] of Point in SetOfPoints. The clusterID (CID) of each term will

[5]The Eps-Neighborhood of point p is defined by $N_{Eps}(p) = \{q \in D | dis(p,q) \leq Eps\}$

Algorithm 1 Boundary Density-Based Clustering

Require: SetOfPoints, Eps, MinPts, Threshold

```
1: ClusterID := nextID(NOISE)
2: for i FROM 1 TO SetOfPoints.size do
3:    Point := SetOfPoints.get(i) {get the i-th element from the SetOf-
      Points}
4:    if Point.CID = UNCLASSIFIED AND ExpandCluster(SetOfPoints,
      Point, ClusterID, Eps, MinPts, Threshold) then
5:       ClusterID := nextID(ClusterID)
6:    end if
7: end for
```

be changed if it is density-reachable[6] from the other terms which may be the neighbor points of a cluster. Otherwise the terms are marked as NOISE if there are less than MinPts of points in an Eps-Neighborhood of that point. Our expand process is designed by rounds, it finds the points that are density-reachable from the points in cluster i and check their average distance to the medoid set which is the points of cluster i conducted by previous rounds. The checking function is shown in Algorithm 3. The point will be grouped to the cluster i if its average distance to the medoid set is less than the given boundary. In addition, the terms may belong to more than one cluster, so that we use the function addCIDs to allow that a point can belong to multi-clusters.

The main difference between Boundary Density-Based Clus-

[6]A point p is density-reachable from a point q if there is a chain of pints $p_1, \cdots, p_n$ where $p_1 = p$ and $p_n = q$

Algorithm 2 ExpandCluster Function

```
1: ExpandCluster(SetOfPoints, Point, CID, Eps, MinPts, Threshold):
   Boolean
2: seeds := SetOfPoints.regionQuery(Point, Eps)
3: if seeds.size < MinPts then {Point is not a core point}
4:   SetOfPoint.changeCID(Point, NOISE)
5:   return False
6: else
7:   round := 0
8:   SetOfPoints.addCIDs(seeds, CID)
9:   medoid.add(Point)
10:  seeds.delete(Point)
11:  while seeds ≠ Empty do
12:    round++
13:    boundary := ThresholdFunction(Threshold, 0.9, round)
14:    new_seeds.clear()
15:    for i From 1 TO seeds.size do
16:      result := SetOfPoints.regionQuery(seeds.get(i), Eps)
17:      if result.size > MinPts then
18:        medoid.append(seeds.get(i))
19:        for k From 1 To result.size do
20:          if Check(result.get(k), boundary, medoid) then
21:            new_seeds.add (result.get(k))
22:            SetOfPoints.addCID(result.get(k), CID)
23:          end if
24:        end for
25:      end if
26:    end for
27:    seeds.replaceAll(new_seeds)
28:  end while
29:  return True
30: end if
31: END {ExpandCluster}
```

Algorithm 3 Checking and Threshold Function

```
1: Check(currentP, threshold, medoid) : Boolean {Check the similarity
   between currentP and all the elements in medoid}
2: sumdis := 0
3: for i FROM 1 TO medoid.size() do
4:    sum_dis += CHI(currentP, medoid.get(i))
5: end for
6: avg_dis := sum_dis / medoid.size()
7: if avg_dis > threshold then
8:    return True
9: else
10:   return False
11: end if
12: END {Check}
13:
14:
15: ThresholdFunction(threshold, factor, round) : double {release the
   threshold by factor and round}
16: return threshold · factor^round
17: {return the value of function with round and factor as its parameters}
18: END {Threshold Function}
```

tering and DBSCAN is that Boundary Density-Based Clustering makes use of a boundary function to restrict the expansion of cluster. In addition, the clusters are expanded by rounds and the boundary will be release in each round. Besides the expansion of the cluster, we allow that a entity can be multi-assigned to the topic clusters.

Boundary Density-Based Clustering may have problem if the density of clusters varies widely. To tackle this issue, we adopt another way to measure the density which is known as Shared Nearest Neighbor (SNN) [9]. It measures the number of shared neighbors as long as the objects are on each other's nearest neighbor lists. SNN similarity can address the problem that occurs in direct similarity. Since it takes the context of an object into account by using the number of shared nearest neighbors, SNN similarity handles the situation in which an object happens to be relatively close to another object, but belongs to a different class. In such cases, the objects typically do not share many near neighbors and their SNN similarity is low.

SNN similarity also addresses the problems related to clusters with varying density. In a low-density region, the objects are farther apart than objects in denser regions. However, the SNN similarity only depends on the number of nearest neigh-

bors two objects share, not how far these neighbors are from each object. Thus, SNN similarity performs an automatic scaling with respect to the density of points. We apply this density definition to our algorithm and the definition of the core point will be changed as follows:

Core points: A point is a core point if the number of points within a given neighborhood around the point determined by SNN similarity and a supplied parameter Eps exceeds a certain threshold MinPts, which is also a supplied parameter.

4.3.3 Document Assignment with Topic Clusters

We apply the Bounded Density-Based Clustering with well defined parameters: Eps, MinPts and T. The result cluster contains a set of concept which is the summarization of certain topic. With the concepts in each cluster, we regard concept cluster as a vector in order to compute the Jaccard Similarity Coefficient between the blog article and topic clusters. Then assign the blog article to the topic cluster if its Jaccard Similarity Coefficient with the cluster is larger than certain threshold. For two term vector A and B, the Jaccard Similarity Coefficient, J, is given as:

$$J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}} \tag{4.16}$$

where

M_{11}: the number of terms that exist in both A and B
M_{01}: the number of terms that do not exist in A but exist in B
M_{10}: the number of terms that exist in A but do not exist in B
M_{00}: the number of terms that exist in neither A or B
$M_{11} + M_{01} + M_{10} + M_{00} = n$

With the process of document assignment, our topic clusters are supported by both concept terms and related blog entries.

4.4 Discussion

Our review on document clustering finds that the problem of blog content clustering has not been widely studied in the previous approaches. Most of the clustering models are not feasible to blog content. Therefore, we propose the concept based clustering method which is developed for blog content. In this section, we compare and discuss the difference between concept clustering and the traditional methods.

An important issue in document clustering is the presence of noise and outliers in blog collection. Our proposed Boundary Density-Based algorithm is robust enough to handle noise and produce high quality clusters which solve the problem of elongation. Since K-means and AHC are simple and effective, they

cannot handle noise and outlier well because they assume that every object must belong to certain cluster. Their performance may be affected by noise and outliers in this way. Thus both K-means and agglomerative hierarchical clustering are not feasible to our clustering problem. In concept clustering, the entities are the terms extracted from blog content which may belong to none or more than one topics. Our proposed clustering method also consider the problem of noise and multi clusters.

DBSCAN is designed for the clustering in spatial database and discover clusters with arbitrary shape. Though DBSCAN is efficient and has the property of noise detection, it is not suitable for concept clustering due to the effect of elongation. In concept clustering, the concepts in each topic cluster need to be correlated to each other if they are describing the same topic. According to DBSCAN algorithm, each cluster is fully expanded as long as their density meets certain threshold value. In this way, some points may not be correlated to each other even though they are formed in the same cluster. For example, point S and point R are expanded from the point O and assigned to the same cluster as shown in Figure 4.8, but in fact they are not highly related to each other. Thus DBSCAN is not feasible to our clustering requirement.

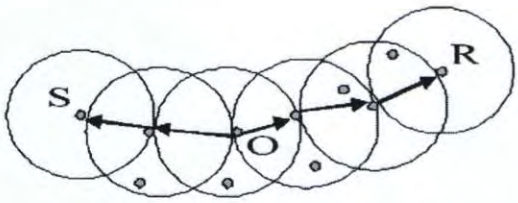

Figure 4.8: The elongation of cluster in DBSCAN

Furthermore, our blog entry collection tends to have documents covering one or more topics. In blog clustering, it is necessary to put those documents into related topic clusters, so that some blog entries may belong to more than one topic cluster. Addressing this requirement, there are few ways for generating overlapping clusters, such as fuzzy clustering [20] where objects can belong to different clusters with different degrees of membership. In our proposed concept clustering, we have overlapped clusters that documents tend to belong to more than one topic because our clustering method is based on the notion of concepts and allows the property of multi-topic document assignment, which is forbidden in the partitional document clustering.

Finally, the ability for presenting the content of cluster is an important issue in document clustering. The result of concept clustering is summarized with related terms and documents, so that users can easily grasp the outline and find out which cluster they are interested in.

Chapter 5

Blog Topic Evolution

In blogosphere, the discussion topics keep changing all the time but the existing clustering models cannot reflect the dynamic change of topic. In this chapter, we present our approach for capturing topic evolution in the stream of blog entries. Our task involves developing a time-dependent weighted graph called Topic Evolution Graph, which presents the topic development by topic clusters obtained from concept clustering discussed in Chapter 4. Then we introduce a tracking algorithm for the extraction of significant topic evolution.

5.1 Topic Evolution Graph

In order to present the temporal changes of topics, we propose a Topic Evolution Graph representing the temporal structure of

topics and the relationships between topics. Given such Topic Evolution Graph, we can easily capture the evolution of certain topics. For instance, the content of topic about "OSCAR" is different before or after the ceremony takes place. At the beginning, there are only some general discussions about the nominated films in blogosphere. Nevertheless, some discussions about the prizewinning films and actors will burst out as soon as the award list is broadcasted. In this way, the content about "OSCAR" will have significant difference after the ceremony takes place.

Our topic evolution graph is a Directed Acyclic Graph (DAG) consisting of topic clusters as nodes and topic evolution relationships as directed edges between nodes. Given a set of n distinct topic clusters $G_i = \{T_{i1}, T_{i2} \cdots, T_{in}\}$ in a given time period i, the directed edge from vertex T_{ix} to T_{jy} is created in the Topic Evolution Graph if there is topic evolution relationship between them and topic cluster T_{im} is the parent of topic cluster T_{jn} where $1 \leq i < j \leq n$. Besides, we have to define the topic evolution relationship, $R = (T_{im}, T_{jn})$ where $T_{im} \in T_i$, $T_{jn} \in T_j$.and $1 \leq i < j \leq n$. Therefore, Topic Evolution Graph is a directed graph, $TEG = \{G, R\}$ where $G = \{G_1, G_2, \cdots G_n\}$ and R is the relationship set, $R = \{R_1, R_2 \cdots R_m\}$ as shown in Figure 5.1.

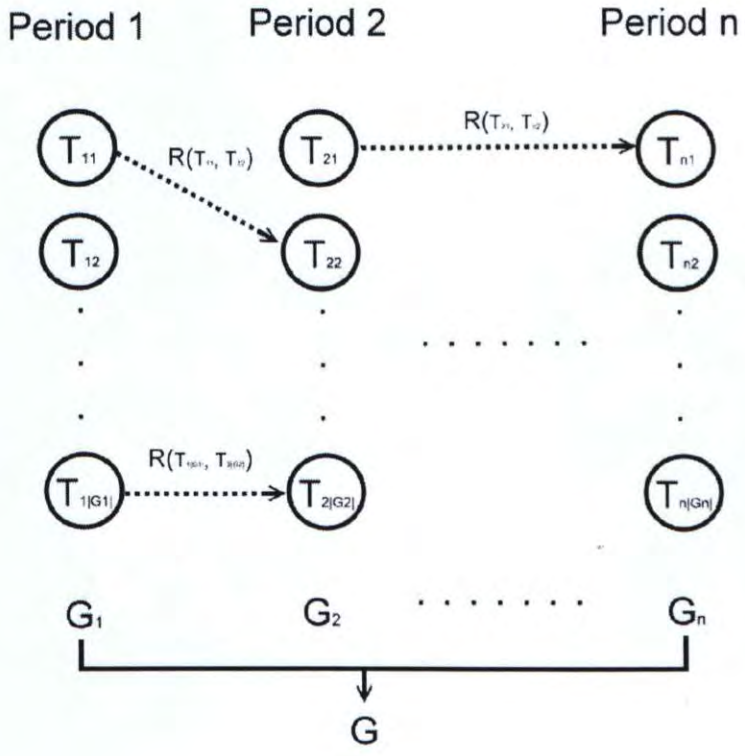

Figure 5.1: Example of Topic Evolution Graph

The task for developing Topic Evolution Graph is to capture the topic evolution.

Evaluating the evolution relationship between topic clusters such that it can be narrating the changes of topics along the timeline, we have to measure the topic similarity and temporal information of each pair of topic clusters. For the temporal information in our relationship strength measurement, we assume that if two topic clusters are distant from one another along the timeline, the topic evolution relationship will be weak and vice versa. In addition, we also consider the timing gap and relation-

ship strength in our evaluation. For topic cluster T_i and T_j, the relationship $R(T_i, T_j)$ will be ignored if it is smaller than certain threshold w or $i - j$ is larger than the given timing gap g.

Giving the topic similarity and timestamp between topic cluster T_i and T_j, the topic evolution relationship $R(T_i, T_j)$ is measured as follows:

$$R(T_i, T_j) = COSINE(T_i, T_j) \cdot TimeFactor(T_i, T_j)$$

and

$$COSINE(T_i, T_j) = \frac{\sum_{x=1}^{k} \omega_{T_i t_x} \cdot \omega_{T_j t_x}}{\sqrt{\left[\sum_{x=1}^{k} \left(\omega_{T_i t_x}\right)^2\right] \left[\sum_{x=1}^{k} \left(\omega_{T_j t_x}\right)^2\right]}}$$

$$TimeFactor(T_i, T_j) = e^{-\rho \cdot [\text{time}(T_j) - \text{time}(T_i)]}$$

where

$\omega_{T_i t_x}$: the term importance of term x in topic cluster T_i
ρ: the time decaying factor which is between 0 and 1
$time(T_i)$: the TimeStamp of topic cluster T_i

5.2 Topic Evolution

Giving a Topic Evolution Graph, we analyze the change of topic content in different periods. We define topic evolution as the set

of topic terms and blog entries describing similar topic in different time periods. Hence we can interpret the topic evolution by the terms involved in different periods.

In this section, we present the tracking algorithm for detecting and capturing significant topic evolutions. In our proposed method, we seek to find the top k topic evolutions with the highest weights as the result of our model.

Our tracking algorithm defines a heap structure to store the tracking information corresponding to the topic clusters. For instance, giving a node T_{ij} in Topic Evolution Graph, we denote a heap structure H_{ij} which represents the tracking paths ending at T_{ij} and their associated weights.

At the beginning, for all nodes T_{ij} belonging to G_i where $1 \leq i \leq n$, their associated heaps H_{ij} are initialized to $\{(T_{ij}, 0)\}$ where the first element indicates the path T_{ij} and the second element indicates its associated weight 0. The tracking algorithm traverses the nodes starting from G_1. When a node T_{ij} is traversed, the heap H_{ij} will be updated by the heaps from the nodes in previous periods. For example, in order to update the heap H_{jn} in node $T_{jn} \in G_j$, all the nodes and their associated heaps T_{im} and H_{im} where $i < j$ from previous periods are stored in memory. If there is a relationship from node T_{im} to node T_{jn},

the heap H_{jn} will be updated by storing all the heap information from H_{im}, and then update the paths by adding T_{jn} as the end of all path and adding $R(T_{im}, T_{jn})$ to the paths' associated weights.

We propose the tracking algorithms as described in Algorithm 4. Assuming that:

G: the Topic Evolution Graph $G = \{G_1, G_2 \cdots G_m\}$

G_i: the topic clusters in i-th time period $G_i = \{T_{i1}, T_{i2} \cdots T_{im}\}$

T_{ij}: the j-th topic cluster in the i-th time period

H_{ij}: the heap associated with T_{ij}

k : the number of evolution paths for the output

g : the largest time period gap for each relationship between the topic clusters in G_i

w : the threshold value for determining whether the relationship between two topic cluster is significant or not

This algorithm can capture all possible evolution paths contained in the Topic Evolution Graph by only a single pass over G and assign them with weight as shown in Figure 5.2. Then we can capture the most significant topic evolution by extracting the paths with the highest weight. In the example of Figure 5.2, the most significant topic evolution of G is $T_{11} => T_{21} => T_{31}$ and its associated wight is 0.5.

Algorithm 4 Tracking Algorithm

Require: G, g, k, w

{create the heap nodes for each topic cluster with paths and weights}
Initialize $H_{ij} = \{(T_{ij}, 0)\}$; {i.e. $H_{ij}.path = T_{ij}$ and $H_{ij}.pathweight = 0$}
for all period i, j from 1 to m where $1 < j - i < g$ and $j < m$ **do**
  {measure the relationship between each topic cluster T_{ix}, T_{jy} in the topic clusters}
  for any topic cluster T_{ix} in G_i and T_{jy} in G_j **do**
    $R(T_{ix}, T_{jy}) = \cos\_sim(T_{ix}, T_{jy}) \cdot e^{-(j-i)}$
    {determine whether the relationship is significant or not}
    if $R(T_{ix}, T_{jy}) \geq w$ **then**
      {append T_{jy} to the paths in H_{ix} and store the updated paths to H_{jy}}
      for all path $p \in H_{ix}$ **do**
        {add $[p.path => T_{jy}, p.pathweight + link(T_{ix}, T_{jy})]$ to H_{jy}}
      end for
    end if
  end for
end for

{extract the top_k_path in the Graph G}
Initialize top_k_path := null
for all time period i from 1 to m **do**
  for each topic cluster $T_{ij} \in G_i$ **do**
    {add all path $p \in H_{ij}$ to top_k_path}
    {prune all the duplicated paths from top_k_path}
    {sort the paths by their weights in top_k_path}
    {extract the top k paths from the top_k_path}
  end for
end for

return top_k_path

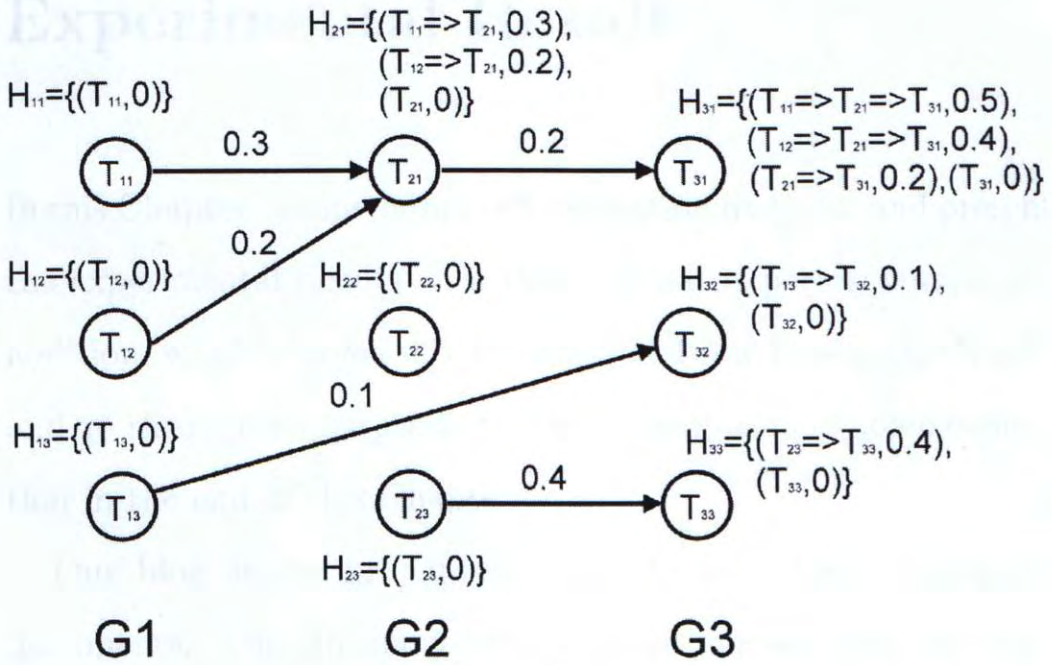

Figure 5.2: Example of tracking process

Chapter 6

Experimental Result

In this Chapter, we introduce our evaluation methods and present the experimental results with their evaluated performances. In addition, we show some selected parts of Topic Evolution Graph and go through a case study for the presentation of topic evolution in the end of this Chapter.

Our blog clustering experiments rely on a large corpus of documents. Though many large corpora are available for text mining research, our target is web documents and blog posts in particular. However, no standard blog corpora are available for our research purpose. Most experiments of the related researches were conducted on proprietary document collection. In this way, we also need to prepare our own blog corpus for the blog clustering purpose.

For the preprocessing of our experiment, we manually set the

extraction rules and built up a crawler to automatically collect blog posts from MySpace[1]. Finally, there are totally 12315 blog posts extracted and stored in our blog corpus starting from November 2008 to April 2009.

6.1 Evaluation of Topic Cluster

In our experiment, we choose 400 valid blog posts from our blog corpus and annotated them with related topics, such as "American Financial Problem", "OSCAR", "Iraq Problem", "Terrorism", etc. Then we apply different clustering methods and evaluate their performances by computing the precision and recall value based on these labels.

6.1.1 Evaluation Criteria

In order to evaluate the effectiveness of our clustering result, we apply the F-measure [16] [38] which considers both precision and recall of result set.

Precision is the ratio of number of records that actually turns out to be classified correctly to the total number of records in the cluster. On the other hand, recall is the ratio of number of positive examples correctly predicted by the classifier to the total

[1]http://blogs.myspace.com/index.cfm?fuseaction=blog.home

number of all records. Both recall and precision are widely used metrics. For the computation of precision and recall, we generally need to prepare a true cluster set in which all the documents are labeled. Thus we annotate 400 blog entries with different topics, such as "American Election", "Terrorism", "Olympics" and "Financial Problems", etc, and apply them in our clustering model.

Given the labeled cluster set, the precision and recall are given as follows:

$$precision = \frac{|\{RelevantDocuments\} \cap \{RetrievedDocuments\}|}{|\{RetrievedDocuments\}|} \tag{6.1}$$

$$recall = \frac{|\{RelevantDocuments\} \cap \{RetrievedDocuments\}|}{|\{RelevantDocuments\}|} \tag{6.2}$$

In our clustering model, the cluster number cannot be decided and some clusters may be duplicated, thus it is hard to evaluate the quality of cluster result. Thus, we need another evaluation method instead of standard precision measurement.

In the decision tree classification, it usually uses the classification error to evaluate the quality of each split of the records. It chooses the best split according to the classification error during

the training process and the classification error shown as follows:

$$Classification\_error(t) = 1 - \max_i [P(i|t)] \tag{6.3}$$

where

$P(i|t)$: the percentage that the element belonging to labeled cluster i are included in cluster t

We follow this concept and define the purity as the measurement of precision which can measure the quality of clusters even the cluster number is not predefined. By using purity measurement, we can evaluate the quality of the clusters by computing the rate of records that belong to the same topic. Let there are k clusters of the data set D and the size of cluster C_j is $|C_j|$. $|C_j|_{class=i}$ denotes number of items of valid class i assigned to cluster j. The purity of each cluster is given by:

$$purity(C_j) = \frac{1}{|C_j|} \cdot \max_i(|C_j|_{class=i}) \tag{6.4}$$

The precision is defined as the overall purity of a clustering solution which is expressed as a weighted sum of individual cluster purities:

$$precision = \sum_{j=1}^{k} \frac{|C_j|}{|D|} \cdot purity(C_j) \tag{6.5}$$

F-measure is defined as the harmonic mean of precision and recall as follows:

$$F - measure = \frac{2 \cdot (precision \cdot recall)}{precision + recall} \tag{6.6}$$

The purity applied in F-measure does not consider the split of clusters, so we apply another external measure of our experimental result known as entropy, which originally provides for evaluation of the clusters at one level of hierarchical clustering. The higher the homogeneity of a cluster, the lower the entropy is. We apply entropy for our reference that prevents our result data set split into too many unnecessary clusters. Given the number of documents $|T_i|$ in each cluster i where $1 \leq i \leq n$ and the total number of documents $|T|$, the entropy is defined as follows:

$$entropy = -\sum_{i=1}^{n} \frac{|T_i|}{|T|} \cdot \log_2 \frac{|T_i|}{|T|} \tag{6.7}$$

6.1.2 Evaluation Result

In order to test the effectiveness of our blog clustering method, we conduct several experiments with different document clustering methods: the traditional document clustering using DBSCAN, concept clustering using DBSCAN and concept clustering using Bounded Density-Based Clustering. The performance of the experimental results is discussed with both F-measure

and entropy.

In the experiment, the traditional document clustering method regards documents as base unit using VSM and measures their semantic distance by cosine similarity discussed in Chapter 4. Given the TF-IDF value of each term in document d_i and document d_j, the cosine similarity is measured by the equation shown in Equation 6.8. With the cosine similarity for distance measurement, DBSCAN can be applied for document clustering.

$$COSINE(d_i, d_j) = \frac{\sum_{t_l \in d_i \cap d_j} F(t_l, d_i) \cdot F(t_l, d_j)}{\sqrt{\left(\sum_{t_l \in d_i} F(t_l, d_i)\right)^2 \left(\sum_{t_l \in d_j} F(t_l, d_j)\right)^2}} \tag{6.8}$$

where

$TFIDF(t_l, d_i)$: the TF-IDF value of term t_l in document d_i

We choose DBSCAN for our blog clustering because it is a standard density-based clustering method that frequently discussed in previous work and it can effectively ignore the noise objects. Besides, we also apply the concept clustering using DBSCAN in order to reveal the difference of performance between concept clustering and traditional document clustering.

In our evaluation, we fix MinPts with well defined value and then tune Eps with different values in order to get the best result. Finally, we get the evaluation result shown in the following part.

Document Clustering Using DBSCAN							
MinPts	4	4	4	4	4	4	4
Eps	0.02	0.03	0.04	0.05	0.08	0.1	0.12
Cluster no	1	1	1	1	1	4	7
Precision	0.47	**0.48**	**0.49**	0.50	0.53	0.53	0.85
Recall	0.89	**0.88**	**0.85**	0.77	0.52	0.35	0.27
F-measure	0.61	**0.62**	**0.62**	0.61	0.52	0.42	0.41
entropy	0.00	0.00	0.00	0.00	0.00	1.25	1.86
max extropy	5.86	5.83	5.79	5.66	5.11	4.72	4.43
entropy ratio	0.00	0.00	0.00	0.00	0.00	0.27	0.42

Table 6.1: Evaluation Result of Document Clustering Using DBSCAN

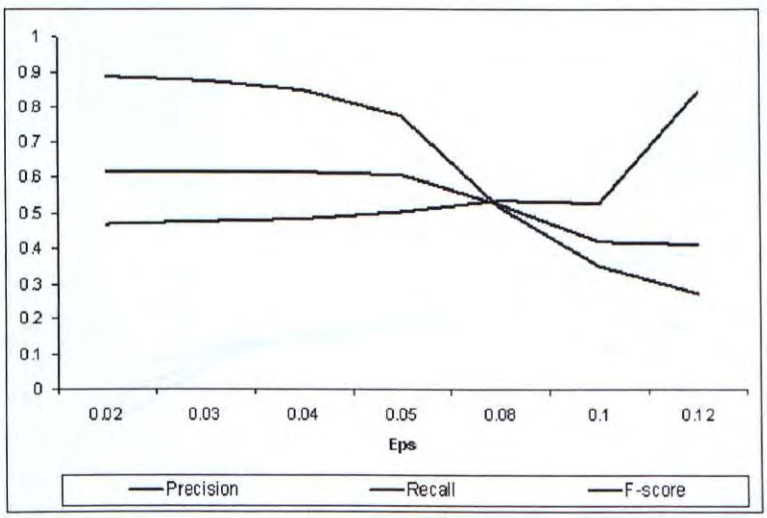

Figure 6.1: Evaluation Result of Document Clustering Using DBSCAN

Table 6.1 shows the evaluation result of the traditional document clustering using DBSCAN. The result reveals that the F-measure falls into the range from 41% to 62% and the highest F-measure is 62% as shown in Figure 6.1. The clustering result only obtains several clusters because the result is affected by the

elongation of clusters in DBSCAN. So that little information can be extracted from the result cluster.

Concept Clustering Using DBSCAN						
MinPts	4	4	4	4	4	4
Eps	0.1	0.15	0.18	0.2	0.22	0.25
Cluster no	1	12	12	15	15	15
Precision	0.49	0.56	0.56	**0.67**	**0.73**	0.78
Recall	0.36	0.53	0.60	**0.60**	**0.56**	0.47
F-measure	0.42	0.55	0.58	**0.64**	**0.64**	0.59
entropy	0.00	1.68	1.99	2.45	2.45	2.40
max extropy	4.89	5.38	5.54	5.55	5.44	5.25
entropy ratio	0.00	0.31	0.36	0.44	0.45	0.46

Table 6.2: Evaluation Result of Concept Clustering Using DBSCAN

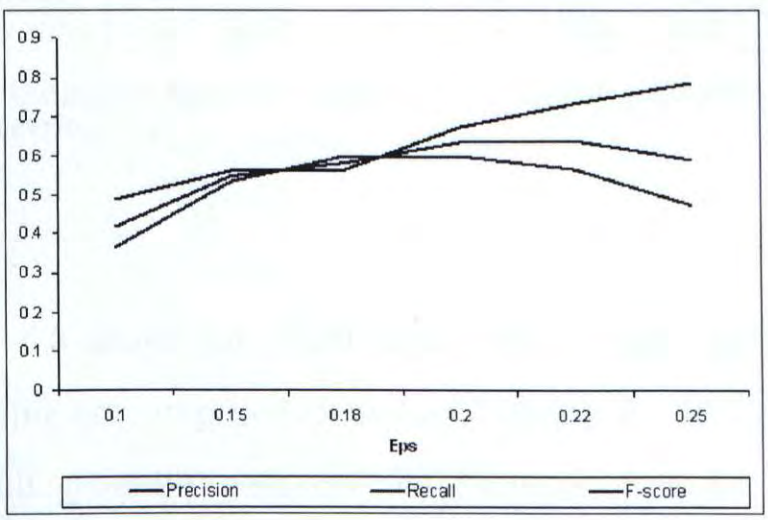

Figure 6.2: Evaluation Result of Concept Clustering Using DBSCAN

Table 6.2 shows the evaluation result of the concept clustering using DBSCAN. Compared with the traditional clustering, it contains much more clusters because the clusters are formed

by concept instead of documents. More information can be extracted in these result clusters. In addition, the evaluation result shows that the F-measure falls into the range from 42% to 64% as shown in Figure 6.2. Its performance is also slightly better than the traditional DBSCAN as presented in Table 6.4.

Concept Clustering Using Bounded Density-Based Clustering							
MinPts	4	4	4	4	4	4	4
Eps	0.08	0.09	0.1	0.12	0.15	0.18	0.2
Threshold	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Cluster no	187	140	109	65	36	24	22
Precision	0.65	0.65	**0.69**	0.70	0.72	0.68	0.73
Recall	0.89	0.89	**0.87**	0.83	0.76	0.72	0.65
F-measure	0.75	0.75	**0.77**	0.76	0.74	0.70	0.69
entropy	4.99	4.69	4.47	3.99	3.33	2.94	2.88
max extropy	8.23	8.03	7.73	7.32	6.65	6.26	6.21
entropy ratio	0.61	0.58	0.58	0.54	0.50	0.47	0.46

Table 6.3: Evaluation Result of Concept Clustering Using Boundary Density-Based Clustering

Table 6.3 shows the evaluation result of the concept clustering using our proposed Boundary Density-Based Clustering. The result generally contains lot of clusters because it forbid the elongation of clusters by using boundary. The highest F-measure in the Table is 77% as shown in Figure 6.3. Its performance is significantly better than the other two clustering methods as presented in Table 6.4. In addition, it can extract

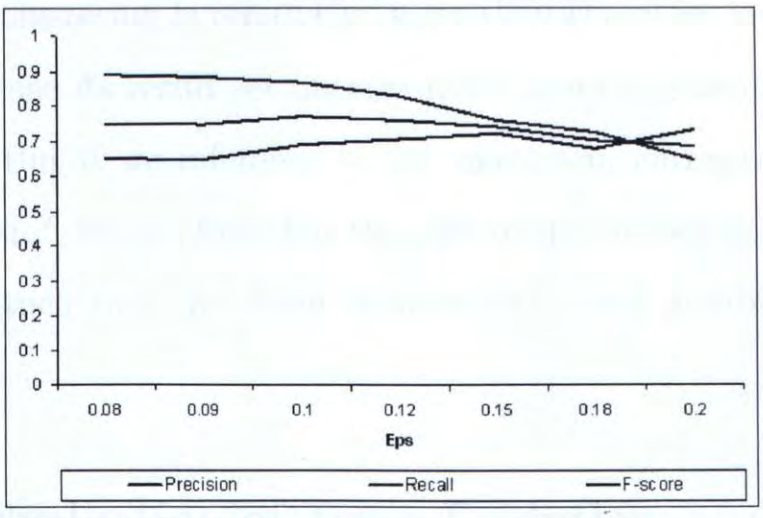

Figure 6.3: Evaluation Result of Concept Clustering Using Boundary Density-Based Clustering

more valid topic clusters and contain more information than the other clustering methods.

	Document Clustering Using DBSCAN	Concept Clustering Using DBSCAN	Concept Clustering Using Bounded Density-Based Clustering
MinPts	4	4	4
Eps	0.04	0.2	0.1
Initial Boundary	/	/	0.1
Cluster no	1	15	109
Precision	0.49	0.67	0.69
Recall	0.85	0.60	0.87
F-measure	0.62	0.64	0.77
entropy	0.00	2.45	4.47
max extropy	5.79	5.55	7.73
entropy ratio	0.00	0.44	0.58

Table 6.4: Evaluation Result

For the comparison of entropy in Table 6.4, the entropy of

concept clustering is relatively larger than the other two methods because its result set obtains much more clusters than the others. But if we reference to the maximum entropy and entropy ratio[2], we can find that the split of the clusters is not very obvious such that the result is informative and feasible to our needs.

6.2 Evaluation of Topic Evolution

One of the main contributions of Topic Detection and Document Clustering is the creation of Topic Evolution Graph. It helps to present the trends and capture topic evolutions by analyzing the evolutions of topic clusters. In this section, we present the selected parts of Topic Evolution Graph, then evaluate the topic evolution and present the significant topic evolution through a case study.

Before building the Topic Evolution Graph, we have to prepare the topic clusters extracted in different periods. In our blog corpus, it comprises the blog posts with the category of News, Politics and Sports extracted in MySpace from November 2008 to April 2009. For our experiment, we extract the blog posts divided by different periods: 2/19-2/21, 2/21-2/23, 2/23-2/25,

[2] $entropy\_ratio = \frac{entropy}{\max\_entorpy}$

2/25-2/27 and 2/27-3/01. There are around 400 blog posts obtained in each period and totally 2000 blog posts in the whole periods.

6.2.1 Results of Topic Evolution Graph

We apply our concept clustering model in each period and extract the significant topics. With the extracted topic clusters, we can measure the evolution relationship by the method suggested in Chapter 5 and build up the Topic Evolution Graph which is partly shown in Figure 6.4.

Figure 6.4 presents part of the Topic Evolution Graph related to "American's Economic and Financial Problem". In the Topic Evolution Graph, the topic clusters T_{ik} in different period i are involved and the evolution relationships $R(T_{ik}, T_{jl})$ are measured. There will be more than one evolution relationships about certain topic as shown in Figure 6.4 and the evolution relationship must be larger than certain threshold, otherwise it will not be analyzed. Only the topic clusters with related topic in different periods are connected.

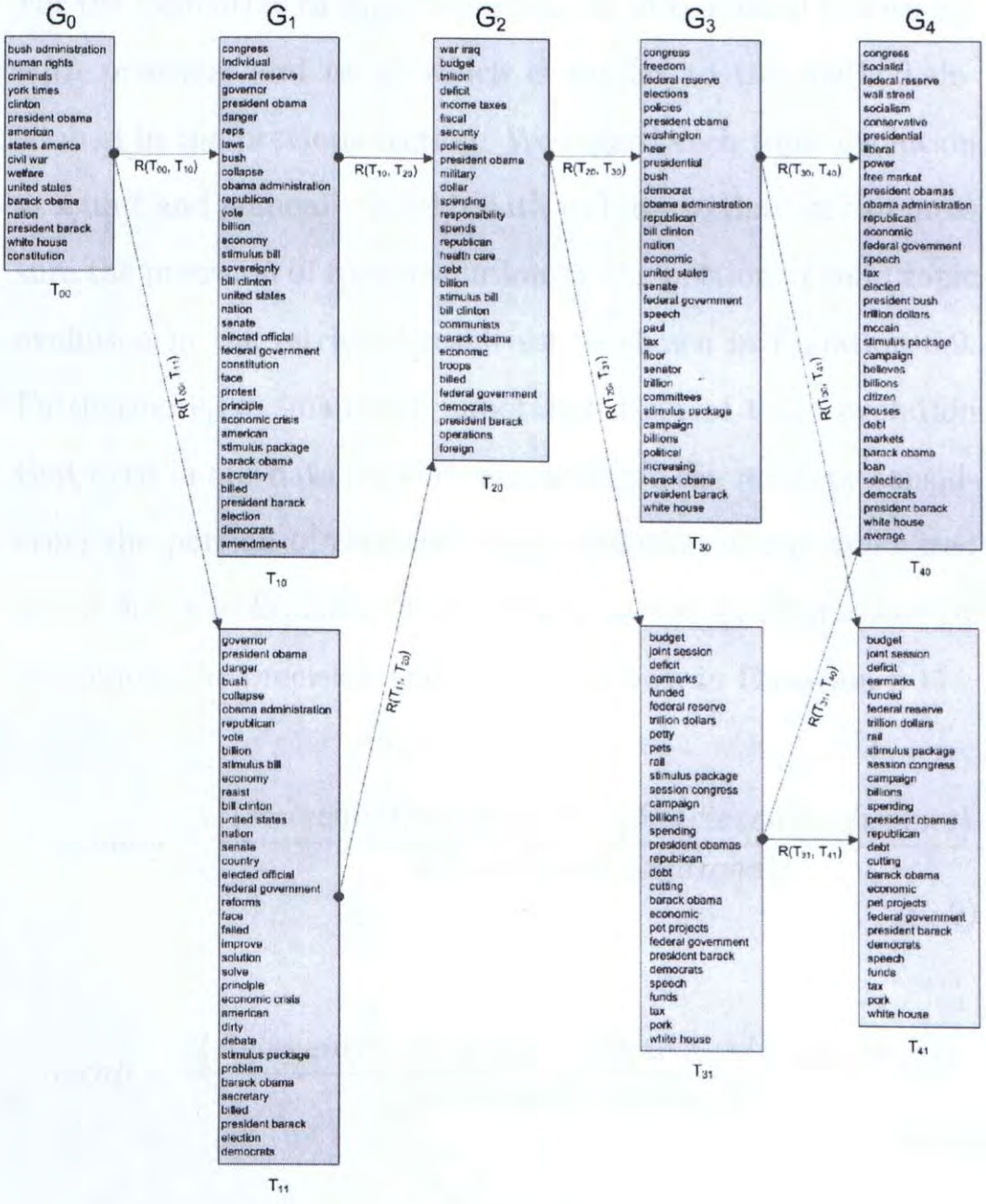

Figure 6.4: Selected part of Topic Evolution Graph

6.2.2 Evaluation Criteria

For the evaluation of topic evolution, we also applied F-measure with precision and recall which is similar to the method described in the previous section. We regard each topic evolution as a unit and manually label it with validity so that we can measure the precision of topic evolution by the portion of valid topic evolution in the retrieved result set as shown in Equation 6.9. Furthermore, we manually annotated a set of topic evolution that exist in our data set and then measure the recall by considering the portion of retrieved topic evolution in our annotated set as shown in Equation 6.10. Finally, we can get F-measure by combining the precision and recall as shown in Equation 6.11.

$$Precision = \frac{|\{RelevantEvolutions\} \cap \{RetrievedEvolutions\}|}{|\{RetrievedEvolutions\}|} \tag{6.9}$$

$$Recall = \frac{|\{RelevantEvolutions\} \cap \{RetrievedEvolutions\}|}{|\{RelevantEvolutions\}|} \tag{6.10}$$

$$F - measure = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall} \tag{6.11}$$

6.2.3 Evaluation of Topic Evolution

With the Topic Evolution Graph, we extract the topic evolution by the suggested algorithm shown in Chapter 5. Due to the nature of our tracking algorithm that all possible paths are considered, there may be some duplicated topic evolutions extracted in our model. Human effort will be needed to interpret the topic evolutions and eliminate the duplicated ones.

Though many topic evolutions are extracted in our model, most of them are easy to interpret, such as the topic about "OSCAR", "Financial problem in America", "Racing", etc. We can simply capture the main topic by identifying the shared terms in the topic clusters from different periods and measure the evolutions by analyzing the term difference between the topic clusters in different periods. More experimental results, such as the topic evolution about "Financial Problem in American" and "Racing", are presented in the Appendix.

In Table 6.5, the evaluation result shows that 44 topic evolutions are found in our tracking algorithm, such as "OSCAR", "IRAQ", "FINANCIAL PROBLEM", "RACING", "POLITIC PROBLEM", etc. There are 28 valid topic evolution within our

Number of Retrieved Topic Evolutions	44
Number of Relevant Topic Evolutions	39
Number of Relevant Topic Evolutions in the retrieved result set	28
Precision	0.63
Recall	0.71
F-measure	0.67

Table 6.5: Evaluation of Topic Evolution

result. Besides, we predefined that there are 39 topic evolutions obtained in our data set. With this information, we finally obtain precision = 0.63, recall = 0.71 and F-measure = 0.67.

Though there is not any existing evaluation about the blog topic evolution for our comparison, we evaluate the performance of our model by F-measure and analyze the result by its rationality. Thus some case studies will be presented in the following section.

6.2.4 Case Study

Besides the evaluation, we also present the topic evolution through a case study about OSCAR. Since OSCAR took place on 2/22, we can get the trend about OSCAR through analyzing the topic evolutions in this period.

We apply our concept clustering model on the blog entries

from 2/19 to 3/01 which is divided into five periods: 2/19-2/21, 2/21-2/23, 2/23-2/25, 2/25-2/27 and 2/27-3/01 and there are 400 blog entries obtained in each time period. With the topic clusters in these five periods, we build up a Topic Evolution Graph and capture the topic evolutions. Table 6.6 shows one of the valid topic evolution related to "OSCAR".

In the topic evolution shown in Table 6.6, some general terms, such as "academy awards", "actor" and "oscars", are occurred in each period. In this way, we can recognize that these topic clusters are discussing about OSCAR. Then we interpret the temporal change of the content by terms involved in different time periods as following:

In the period from 2/19 to 2/21 which is the eve of the Oscar ceremony, the topic terms are mainly about some popular and nominated move or actor name, such as "slumdog millionaire", "mickey rourke", "frost nixon", etc.

In the period from 2/21 to 2/23, more terms are shown up because the OSCAR ceremony took place on 2/22. The terms include the host of OSCAR ceremony "hugh jackman", the award winner "kate winslet", "lance black", "danny boyle", "heath

OSCAR				
2/19-2/21	2/21-2/23	2/23-2/25	2/25-2/27	2/27-3/01
academy awards	academy awards	academy awards	academy awards	dark knight
benjamin button	acceptance speech	anne hath-away	benjamin button	actor
case ben-jamin	anne hath-away	benjamin button	dark knight	awards
curious case	benjamin button	case ben-jamin	heath ledgers	millionaire
frost nixon	brad pitt	curious case	sean penn	oscar
mickey rourke	case ben-jamin	danny boyle	curious	predictions
sean penn	curious case	dark knight	actor	slumdog
slumdog millionaire	slumdog mil-lionaire	slumdog mil-lionaire	slumdog millionaire	
aaron	dark knight	hugh jackman	hollywood	
actors	dustin lance	kate winslet	kate	
adam	heath ledger	lance black	movie	
annual	hugh jackman	mickey rourke	nominated	
thunder	kate winslet	penelope cruz	oscars	
brad	lance black	heath ledger	recap	
danny	mickey rourke	sean penn		
fantastic	penelope cruz	actor		
film	red carpet	adapt		
kate	sean penn	batman		
milk	danny boyle	films		
movie	supporting actress	harvey		
nominated	actors	hollywood		
oscars	film	joker		
wrestler	frost	ledger		
winner	harvey	milk		
	hollywood	nominees		
	hughes	oscar		
	joker	reader		
	milk	screenplay		
	nominees	wrestler		
	oscar			
	reader			
	rourke			
	screenplay			
	winner			
	wrestler			

Table 6.6: Topic Evolution about "OSCAR"

ledger", "penelope cruz" and "sean penn". This shows that the content about OSCAR is updated after the ceremony took place.

In the periods from 2/23 to 3/01, the terms involved in the topic cluster about OSCAR are decreasing gradually because the discussion about the ceremony is becoming fewer and fewer. Only some general terms are left in these periods.

In Table 6.6, the topic evolution is meaningful. We can interpret the topic evolution easily and the result is consistent to our expectation that the topic terms are burst out as soon as the ceremony takes place and the terms are decreasing gradually afterward.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

This thesis is related to web content mining which is concerned with several areas, including data mining, text mining, pattern recognition, knowledge discovery, and artificial intelligence. Our approaches mainly address the problem of information categorization on blog posts. Blog posts are time-series documents that continually delivered with timestamps. Besides extracting and summarizing the hot topics from blog posts, we also evaluate both the content and temporal information, and capture the significant topic evolution from the selected periods.

Challenged by the issues that standard document clustering techniques are not feasible to blog content due to the special content nature, we introduce our framework for blog content

mining based on some novel ideas. We also present a method for solving the problem of extraction of hot topics and evolutions from the blog, and show our experimental results.

Our concept clustering model can be divided into three parts: blog analysis, blog clustering, and topic evolutions extraction. First, we introduce the blog analysis that identifies the blog structure and builds structured information representation out of the semi-structured web documents. Afterward, it comes to the main component of our work which presents the concept clustering and Boundary Density Based Clustering. Our proposed clustering model captures significant blog topics based on semantic relation between concepts and extracts the topics summarized with keywords and supporting blog entries. Finally, we develop Topic Evolution Graph and capture the significant topic evolution by our tracking algorithm.

The problem of document categorization generally does not have one "best" solution. The result will be different if we manually categorize the documents by different people. Therefore, categorizing the documents automatically will not always satisfy everyone. It only provides a structured way for us to analyze the content. However, we implement our proposed system, present the experimental result and evaluate it with other stan-

dard clustering model. The result showed that our clustering model outperforms the traditional ones and the result of topic evolution is also feasible to our actual needs.

7.2 Future Work

Our plan for future work involves the improvement of feature selection and extraction techniques.

The performance of topic cluster is linked up with the quality of feature selection. In several experiments, we found that some key features are not extracted in our model. We need some advanced techniques for text pre-processing, such as part-of-speech tagging, named entity recognition and specific parsing algorithm, that may result in further improvements for blog clustering.

From semantic view, it is difficult to find the topic's dynamic change only with a single individual topic development because the state change of topic is difficult to be described accurately. We plan to investigate more Topic evolutions with different factors which may help us understand the improvements. For example, we can achieve the communities discussing similar topics in their blog entries. In this way, more blog entries with same topic can be retrieved by traversing the blog entries posted by

the bloggers involved in related community.

In addition, lot of parameters are needed in our model, such as Eps and MinPts for the definition of density, the threshold function for the expansion boundary, the number of concepts in the feature selection process and the number of topic evolution in the tracking algorithm. Thus human effort is required in our model in order to achieve the best result. Our future improvement can be concentrate on how to simplify the clustering model and make the process more automatically.

Finally, what remains to consider is whether or not our model can be applied to our practical usage. We can conduct more experiments to see whether our model work in different corpora or not. Though our work is developed specific for blog content, we also can apply it to different kind of web content, such as web forum and news media to analyze its result.

Bibliography

[1] N. Agarwal, M. Galan, H. Liu, and S. Subramanya. Clustering with collective wisdom - a comparative study. In *Proceedings of ICWSM'08, Seattle, Washington, USA*, 2008.

[2] N. Agarwal and H. Liu. Blogosphere: research issues, tools, and applications. *ACM SIGKDD Explorations Newsletter*, 10(1), June 2008.

[3] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the ACM SIGMOD international conference on Management of data, Philadelphia, Pennsylvania, United States*, May 31-June 03.

[4] A. Aschenbrenner and S. Miksch. Blog mining in a corporate environment. *Technical Report ASGAARD-TR-2005-11, Smart Agent Technologies*, 2005.

[5] N. Bansal, F. Chiang, N. Koudas, and F. W. Tompa. Seeking stable clusters in the blogosphere. In *Proceedings of VLDB'07, Nienna, Austria*, Semptember 2007.

[6] M. Chau, J. Xu, J. Cao, P. Lam, and B. Shiu. A blog mining framework. *IT Professional*, 11(1):36–41, Jan./Feb. 2009.

[7] Y. Chen, F. S. Tsai, and K. L. Chan. Machine learning techniques for business blog search and mining. *Expert Systems with Applications: An International Journal*, 35(3):581–590, 2008.

[8] Y. Chi, S. Zhu, X. Song, J. Tatemura, and B. L. Tseng. Structural and temporal analysis of the blogosphere through community factorization. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, New York, USA*, pages 163–172, 2007.

[9] L. Ertoz, M. Steinbach, and V. Kumar. A new shared nearest neighbor clustering algorithm and its applications. In *Workshop on Clustering High Dimensional Data and its Applications, SIAM Data Mining 2002, Arlington, VA, USA*, 2002.

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the International Conference on Knowledge Discovery in Databases and Data Mining, AAAI Press, Portland, Oregon. KDD'96*, pages 226–231, 1996.

[11] L. Gonzaga, M. Grivet, and A. T. Vasconcelos. A simple and fast term selection procedure for text clustering. In *Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications. ISDA 2007*, pages 777–781, 2007.

[12] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. Clustering algorithms and validity measures. In *Proceedings of the Thirteenth International Conference on Scientific and Statistical Database Management, SSDBM 2001, Fairfax, VA, USA*, pages 3–22, November 2001.

[13] K. M. Hammouda. Web mining: Identifying document structure for web document clustering. *Master's Thesis, Department of Systems Design Engineering, University of Waterloo, Waterloo, Ontario, Canada*, 2002.

[14] M. A. Hearst, M. Hurst, and S. T. Dumais. What should blog search look like? In *Proceedings of SSM'08, Napa Valley, California, USA*, pages 95–98, October 2008.

[15] A. Hotho, A. Maedche, and S. Staab. Ontology-based text document clustering. In *Proceedings of the IJCAI'01 Workshop.*

[16] A. Hotho, A. Nurnberger, and G. Paas. A brief survey of text mining. *Zeitschrift fuer Computerlinguistik und Sprachtechnologie (GLDV-Journal for Computational Linguistics and Language Technologie)*, 20(1):19–62, 2005.

[17] M. Hu, A. Sun, and E.-P. Lim. Comments-oriented blog summarization by sentence extraction. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, Lisbon, Portugal*, pages 901–904, November 2007.

[18] L. Jing, L. Zhou, M. K. Ng, and J. Z. Huang. Ontology-based distance measure for text clustering. In *Proceedings of the Workshop on Text Mining, SIAM International Conference on Data Mining, Bethesda, MD*, 2006.

[19] R. Kosala and H. Blockeel. Web mining research: A survey. *SIGKDD Explorations*, 2(1):1–15, 2000.

[20] H.-P. Kriegel and M. Pfeifle. Density-based clustering of uncertain data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. KDD'05*, pages 672–677, 2005.

[21] R. Kumar, J. Novak, P. Raghvan, and A. Tomkins. On the bursty evolution of blogspace. In *Proceedings of WWW'03, Budapest, Hungary*, May 2003.

[22] J. G. W. Lai. Visualizing blogsphere using content based clusters. *wi-iat, International Conference on Web Intelligence and Intelligent Agent Technology*, 1:832–835, 2008.

[23] J. Lasica. Weblogs: A new source of news. *In J.Rodzville (ed.) We've Got Blog, Cambridge: Perseus Publishing*, pages 171–182, 2002.

[24] B. Li, S. Xu, and J. Zhang. Enhancing clustering blog documents by utilizing author/reader comments. In *Proceedings of the 45th ACM Southeast Conference (ACMSE 2007), Winston-Sale, North Carolina, USA*, March.

[25] L. LIU, J. KANG, J. YU, and Z. WANG. A comparative study on unsupervised feature selection methods for text clustering. In *Proceedings of Natural Language Pro-*

cessing and Knowledge Engineering'05, pages 597–601, October, November 2005.

[26] T. Liu, S. Liu, Z. Chen, and W.-Y. Ma. An evaluation on feature selection for text clustering. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washingon DC*, pages 213–219, 2003.

[27] J. Makkonen. Investigations on event evolution in tdt. In *Proceedings of HLT-NAACL 2003 Student Workshop*, pages 43–48, 2004.

[28] Q. Mei and C. Zhai. Discovering evolutionary theme patterns from text - an exploration of temoral text mining. In *Proceedings of KDD'05, Chicago, Illionis, USA*, pages 198–207, August 2005.

[29] B. A. Nardi, D. J. Schiano, M. Gumbrecht, and L. Swartz. Why we blog. *Communications of the ACM*, 47(9):41–46, Dec. 2004.

[30] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[31] R. Prabowo and M. Thelwall. A comparison of feature selection methods for an evolving rss feed corpus. *Informa-*

tion Processing and Management: an International Journal, 42(6):1491–1512, 2006.

[32] A. Qamra, B. Tseng, and E. Y. Chang. Mining blog stories using community-based and temporal clustering. In *Proceedings of the 15th ACM international conference on Information and knowledge management (Arlington, Virginia, USA)*, November.

[33] J. Qiu, C. Li, S. Qiao, T. Li, and J. Zhu. Timeline analysis of web news events. In *Proceedings of ADMA*, pages 123–134, 2008.

[34] D. R. Recupero. A new unsupervised method for document clustering by using wordnet lexical and conceptual relations. *Inf. Retr.*, 10(6):563–579, 2007.

[35] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, November 1975.

[36] Y. SEKIGUCHI, H. KAWASHIMA, H. OKUDA, and M. OKU. Topic detection from blog documents using users' interests. In *Proceedings of the 7th International Conference on Mobile Data Management MDM'06*, 2006.

[37] D. Sifry. Sifry's alerts. *State of Technorati*, Accessed on April 05 2007. http://www.sifry.com/alerts/archives/000493.html.

[38] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. In *KDD Workshop on Text Mining*, 2000.

[39] T. Tirapat, C. Espiritu, and E. Stroulia. Taking the community's pulse, one blog at a time. In *Proceedings of the 6th international conference on Web engineering, Palo Alto, California, USA*, pages 169–176, July 2006.

[40] H. Wang. Business blog mining based on hierarchical svm. In *Proceedings of 2008 International Symposium on Knowledge Acquisition and Modeling*, pages 837–841, 2008.

[41] J. Wang, X. Geng, K. Gao, and L. Li. Study on topic evolution based on text mining. In *Proceedings of the Fifth International Conference on Fuzzy Systems and Knowledge Discovery. FSKD'08*, volume 2, pages 509–513, October 2008.

[42] R. Xu and D. Wunsch. Survey of clustering algorithms. *EEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[43] X. Xu, M. Ester, H.-P. Kriegel, and J. Sander. A distribution-based clustering algorithm for mining in large spatial databases. In *Proceedings of the Fourteenth International Conference on Data Engineering*, pages 324–331, February 1998.

[44] C. C. Yang, X. Shi, and C.-P. Wei. Tracing the event evolution of terror attacks from on-line news. In *Proceedings of ISI'06*, pages 343–354, 2006.

[45] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning. ICML'97*, pages 412–420, 1997.

[46] O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *Proceedings of the 21st SIGIR'98, Melbourne, Australia*, pages 46–54, 1998.

Appendix A

Stop Word List

a	aside	comments	ever	hence	known
able	ask	concerning	every	her	knows
about	asking	consequently	everybody	here	l
above	associated	consider	everyone	hereafter	last
according	at	considering	everything	hereby	lately
accordingly	available	contain	everywhere	herein	later
account	away	containing	ex	hereupon	latter
across	awfully	contains	exactly	hers	latterly
actually	b	copyright	example	herself	least
add	be	corresponding	except	hi	less
added	became	could	f	him	lest
address	because	couldn	far	himself	let
addresses	become	couldnt	few	his	like
after	becomes	course	fifth	hither	liked
afterwards	becoming	currently	first	hopefully	likely
again	been	d	five	how	link
against	before	definitely	followed	howbeit	little
all	beforehand	described	following	however	ll
allow	behind	despite	follows	href	look
allows	being	did	for	i	looking
almost	believe	didn	former	ie	looks
alone	below	didnt	formerly	if	ltd
along	beside	different	forth	ignored	m
already	besides	do	four	img	mail
also	best	does	from	immediate	mainly
although	better	doesn	further	in	many
always	between	doesnt	furthermore	inasmuch	may
am	beyond	doing	g	inc	maybe
among	blog	don	get	indeed	me
amongst	both	done	gets	indicate	mean
an	br	dont	getting	indicated	meanwhile
and	brief	down	given	indicates	merely
another	but	downwards	gives	inner	might
any	by	during	go	insofar	more
anybody	c	e	goes	instead	moreover
anyhow	came	each	going	into	most
anyone	can	edu	gone	inward	mostly
anything	cannot	eg	got	is	much
anyway	cant	eight	gotten	isn	must
anyways	cause	either	greetings	isnt	my
anywhere	causes	else	h	it	myself
apart	certain	elsewhere	had	its	n
appear	certainly	email	happens	itself	name
appreciate	changes	enough	hardly	j	name
appropriate	clearly	entirely	has	just	namely
are	co	entry	have	k	nd
aren	com	especially	having	keep	near
arent	come	et	he	keeps	nearly

Table A.1: Stop Word List I

needs	password	selves	them	url	whom
neither	per	sensible	themselves	urls	whose
never	perhaps	sent	then	us	why
nevertheless	placed	serious	thence	use	will
new	please	seriously	there	used	willing
next	plus	seven	thereafter	useful	wish
nine	pm	several	thereby	uses	with
no	possible	shall	therefore	using	within
nobody	post	she	therein	usually	without
non	posted	should	theres	uucp	won
none	posts	since	thereupon	v	wonder
noone	presumably	site	these	value	wont
nor	probably	six	they	various	would
normally	provides	so	think	ve	would
not	q	some	third	very	x
nothing	que	somebody	this	via	y
novel	quite	somehow	thorough	viz	yes
now	qv	someone	thoroughly	vs	yet
nowhere	r	something	those	w	you
o	rather	sometime	though	want	your
obviously	rd	sometimes	three	wants	yours
of	re	somewhat	through	was	yourself
off	really	somewhere	throughout	way	yourselves
often	reasonably	soon	thru	we	z
oh	regarding	sorry	thus	web	zero
ok	regardless	specified	to	web	around
okay	regards	specify	together	website	as
old	relatively	specifying	too	welcome	comes
on	require	src	took	well	comment
once	required	still	toward	went	etc
one	requires	sub	towards	were	even
ones	respectively	such	trackback	what	hello
only	right	sup	trackbacks	whatever	help
onto	s	sure	tried	when	kept
or	said	t	tries	whence	know
other	same	tags	truly	whenever	necessary
others	saw	take	try	where	need
otherwise	say	taken	trying	whereafter	particular
ought	saying	tell	twice	whereas	particularly
our	says	tends	two	whereby	seen
ours	second	th	u	wherein	self
ourselves	secondly	than	un	whereupon	their
out	see	thank	under	wherever	theirs
outside	seeing	thanks	unfortunately	whether	up
over	seem	thanx	unless	which	upon
overall	seemed	that	unlikely	while	whoever
own	seeming	thats	until	whither	whole
p	seems	the	unto	who	

Table A.2: Stop Word List II

Appendix B

Feature Selection Comparison

χ^2	MI	Weight
president barack	joe bidens	improve
biden	low income	parties
infrastructure	wasteful	partly
expand	obama introduced	biden
obama	cell research	transportation
joe bidens	biden	ensures
low income	president barack	quot
opportunities	opportunities	strengthen
strengthen	president biden	worked
ensures	tax credits	joe bidens

Table B.1: the top 10 words with the closest semantic distance with "barack obama"

χ^2	MI	Weight
homeland	taliban leader	cyber
intelligence	homeland	nuclear weapons
civil liberties	civil liberties	improve
propaganda	propaganda	homeland
private sector	distribute	capacity
terrorist	private sector	parties
sector	sector	partly
distribute	afghan	nuclear
taliban leader	intelligence	strengthen
law enforcement	law enforcement	intelligence

Table B.2: the top 10 words with the closest semantic distance with "terrorism"

χ^2	MI	Weight
vice	tax incentives	biden
president biden	president biden	president biden
biden	obama introduced	parties
obama introduced	technologies	partly
president	vice	vice
president obama	tax credits	teacher
ensures	innovation	president obama
encourage	ensures	violence
quot	low income	domestic
opportunities	biden	percent

Table B.3: the top 10 words with the closest semantic distance with "vice president"

χ^2	MI	Weight
bankers	monetary	monetary
monetary	imf	bankers
imf	bankers	argues
argues	argues	financial
financial	toilet	draft
capital	governors	banks
draft	mob	summit
summit	carter	london
inter	anarchists	target
chief executive	wore	toilet

Table B.4: the top 10 words with the closest semantic distance with "financial crisis"

Appendix C

Topic Evolution

2/20-2/23	2/22-2/24	2/24-2/26	2/25-2/27
carl	acc	auto	kyle
edward	carl	common	las vegas
engine	daytona	driver	motor
flag	driver	engines	nascars
gordon	duke	jeff	racing
jeff	edwards	kyle	
kyle busch	engine	lap	
laps	espn	las vegas	
matt	forest	nascar	
racing	jeff gordon	racing	
	johnson	republic	
	kyle busch	restore	
	laps	stewart	
	las vegas		
	maryland		
	matt kenseth		
	nascar		
	north carolina		
	pit		
	racing		
	speedway		
	sprint		
	stewart		
	tony		
	truck		

Table C.1: Topic Evolution about "Racing"

2/20-2/23	2/22-2/24	2/24-2/26
barack obama	barack obama	banks
bill clinton	bill clinton	barack obama
billions	billion	billions
bush administration	bush	bush administration
collapse	collapse	clinton
danger	congress	conservative
democrats	danger	country
dirt	debate	debts
economic crisis	democrats	democrat
economy	dirty	economic
elections	economic	executive
face	economy	federal government
fannie mae	election	federal reserve
freddie	face	financial
obama administration	federal government	funds
president barack	federal reserve	george bush
president obama	governor	income
reading	laws	lending
ready	obama administration	loans
republican	president obama	members congress
stimulus bill	republican	nation
stimulus package	secretary	overseas
	senate	stimulus packages
	stimulus bill	president bushs
	stimulus package	president obama
	united states	republican
	vote	saving
		speech
		spend
		stimulus bill
		tanks
		tax cuts
		tax increases
		wall street

Table C.2: Topic Evolution about "Financial Problem in America"

Appendix D

Topic Cluster

OSCARS	
harvey	heath ledger
books	sean penn
milk	dark knight
wrestler	reader
rourke	hollywood
mickey rourke	actress
oscar	anne hathaway
benjamin button	joker
curious case	comic
chicken	penelope cruz
rice	boyle
vegetables	academy awards
case benjamin	hugh jackman
films	nominees
slumdog million-aire	screenplay
danny boyle	actor
kate winslet	nominated
lance black	batman

Table D.1: Topic Clusters I

FOOD	
burger	mcdonalds
flavors	mcnuggets
tastes	called 911
fast food	goodman
fries	refund
mcdonalds	nuggets
	misuse
	florida
	emergency
	chicken
	credit cards

Table D.2: Topic Clusters II

IRAQ and TERRORISM		
military	middle east	israel
operations	afghanistan	israelis
troops	iraq	palestinians
qaeda	taliban	middle east
iraqi	israel	prime minister
launch	terrorist	
security	moderate	

Table D.3: Topic Clusters III

MISCELLANEOUS	
pregnant	women
father	anti
boy	sexy
charges	culture
style	men women
charged	young
lover	hook
murder	
shotgun	
sleeping	

Table D.4: Topic Clusters IV

SPORT					
race	acc	champion	jerry	pace	match
espn	duke	wwe	nfl	marathon	champions
maryland	forest	raw	head coach	mile	wrestling
wake	north carolina	match	terrell owens	half marathon	main event
kyle busch	jeff gordon	championship	owens	runners	tag team
carl	johnson	wrestling	football	manhattan	heavyweight
edwards	laps	wrestlers	cowboys		
truck	carolina	ring	buffalo		
racing	kenseth	storyline	dallas		
las vegas	matt kenseth	main event			
engine	tony				
driver	daytona				
nascar	pit				
speedway	stewart				
sprint					

Table D.5: Topic Clusters V

FINANCIAL PROBLEM			
face	face	stimulus	income
governor	collapse	republican	loans
republican	economy	obama	financial system
stimulus package	danger	billion	wall street
billion	read	barack obama	government
barack obama	federal reserve	president	street
state	banks	democrats	tanks
collapse	banking system	stimulus bill	wall
president	reserve	bill clinton	
vote		billed	
economy			
danger			
federal government			
stimulus bill			
federal reserve			
economic			
bush			
democratic			
dollar			
secretary			
democrats			
bill clinton			
senate			
administration			
debate			
dirty			
reserve			
billed			

Table D.6: Topic Clusters VI