



THE HONG KONG
POLYTECHNIC UNIVERSITY

香港理工大學

Pao Yue-kong Library

包玉剛圖書館

Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

By reading and using the thesis, the reader understands and agrees to the following terms:

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact lbsys@polyu.edu.hk providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

The Hong Kong Polytechnic University

Department of Computing

Preprocessing Frameworks for Threaded Discussion Analysis by
Graphical Probabilistic Modeling

Donahue Chun-ming Sze

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy

March 2008

CERTIFICATE OF ORIGINALTY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____ (Signature)

Donahue Chun-ming, Sze (Name)

Abstract

User generated content (UGC) has become the fastest growing sector of the World Wide Web. Today, one major type of massive UGC data is generated from web forums. The web forum, similar to USENET, is a bulletin board commonly used by users to exchange ideas, publish topics, or simply send replies via the HTML based browser. Since almost all computers are equipped with the pre-installed browser and can be easily accessed, the web forum has become more popular, and is considered as a significant contributor of the UGC data. With the growing importance of such web forum data, there are increasing and compelling needs to develop techniques to help analyze such tons of data, for example, grouping them in a meaningful and an user-friendly manner.

Recently, Bayesian methods have grown from specialist niche to mainstream in the field of pattern recognition and machine learning. The graphical probabilistic model (GPM), induced by probability and graph theories, offers numerous useful properties to analyze data by using diagrammatic representations of probability distributions under the Bayesian perspective. By using effective algorithms like Gibbs Sampling, one may formulate topical problems (e.g. hot topics in a forum) in the latent variable model and obtains quality results in a tractable manner.

In addition, we may also infer the relationship between different textual type variables (e.g. author, entity, word, and sentiment) in the Markov random fields.

To analyze the web forum, one of the easiest ways is to directly convert a post or a thread as a bag of words (BOW) vector space representation and perform one of the graphical probabilistic modeling for instance latent variable modeling (for topical modeling) or Markov random fields (for non-topical modeling). However, the transformation of bag of words of threaded text may lead to a serious loss of important information, making the analysis or mining process ineffective. By using different graph models and inference techniques, we can develop a set of preprocessing frameworks to facilitate the analysis of web forum data. In topical modeling, we propose a framework for word-thread matrix formation. In order to provide more representative bag of words for latent variable modeling, our framework is designed to measure both implicit and explicit relationships between posts and replies. It consists two parts. In the first part, a threaded text is transformed to a directed acyclic graph (DAG) by a set of feature link generation functions. In the second part, different graph based ranking algorithms can be applied. Our framework, then, extracts a list of words by weighting the importance ranking value with traditional feature selection method. In non-topical modeling, on the other hand, we propose a distributional similarity model (DSM) to analyze the relationship between different textual type variables of a thread in the Markov random fields. This model is employed to measure not only the co-occurrence but also a distributional similarity in different type of distance level commonly found in threaded text. Empirical results obtained for the Hong Kong popular web forums show that the proposed methods are effective.

Acknowledgement

This thesis would not have been possible without the help and support of many friends and colleagues.

Foremost, I thank my advisor Dr. Chung Fu-lai (Korris). For the past two years, Korris has been an exemplary teacher, and mentor. I could not have imagined having a better advisor for my research degree, and without his knowledge, perceptiveness and appeal. I would never have finished.

I am fortunate to have a chance to go back to school and to pursue my M.Phil. degree that was ever my dream when I am still an undergraduate. I would very much like to thank you Dr. Fu Tak Chung and those management team of Well Synergy Limited to give me a chance to participate this Teaching Company Scheme.

I am grateful for Dr. Ken Sit (my dearest uncle) for his help to review some part of my thesis. Although he is not a technical IT person, he made a lot of valuable comments for my writing. I also thank all my friends, especially Kennis Yu, Cecilia Hon, Hugo Lam, C.Y. Ng and Avis Ku, their selfless supports and guidance.

Last but not least, special thanks my parents, brother and sister who have given me a lifetime of love and care.

Table of Contents

Abstract	3
Acknowledgement.....	5
Table of Contents	6
List of Figures	10
List of Tables.....	12
List of Algorithms	13
Chapter 1 Introduction	14
1.1 Problem and Motivation.....	14
1.2 Objectives.....	17
1.3 Contributions	17
1.4 Outline of the Thesis	18
Chapter 2 Graphical Probabilistic Modeling for Textual Data.....	19
2.1 Text Mining.....	19
2.2 Bag of Words (BOW) Representation.....	20
2.3 Latent Semantic Analysis (LSA).....	22

2.4	Probabilistic Latent Semantic Analysis (PLSA)	23
2.5	Graphical Probabilistic Model (GPM)	24
2.5.1	Bayesian Theorem.....	25
2.5.2	Graph Model	27
2.5.3	Bayesian Networks and Markov Random Fields.....	27
2.5.4	Textual Analysis.....	28
2.5.5	Approximate Inference.....	29
2.6	Conclusion.....	30
Chapter 3 Topical Modeling in Threaded Discussion Analysis.....		32
3.1	Introduction	32
3.2	Conversation Focus Detection Framework	34
3.2.1	Thread Representation by Directed Graph.....	36
3.2.2	Feature-oriented Link Generation	36
3.2.3	Graph-based Ranking Algorithms	40
3.2.4	Algorithmic Framework.....	42
3.3	Latent Variable Modeling	44
3.3.1	Latent Dirichlet Allocation	46
3.3.2	Topic-Entity Modeling.....	48
3.3.3	Topic-Time Modeling.....	49
3.4	Experimental Results.....	50

3.4.1	Datasets	51
3.4.2	Basic Analysis of Threaded Text	53
3.4.3	Golden Data Set	55
3.4.4	Mean Square Error	56
3.4.5	Evaluation of Different Feature-oriented Link Generation Methods.....	57
3.4.6	Evaluation of Different Graph-based Ranking Algorithms	60
3.4.7	Topic Modeling Results	63
3.5	Conclusion.....	65
Chapter 4 Non-Topical Modeling in Threaded Discussion Analysis		66
4.1	Introduction	66
4.2	Distributional Similarity Model	68
4.2.1	The Nature of UGC Data	68
4.2.2	Distributional Similarity	70
4.3	Multi-Modality Clustering Algorithm.....	72
4.4	Experimental Results.....	73
4.5	Conclusion.....	76
Chapter 5 Conclusion.....		77
5.1	Contribution.....	77
5.2	Future works.....	78
Appendices		83

- A. Some results of bag of words extracted by graph-based ranking algorithms..... 83
- B. Some results of LDA approximated results by using Gibbs sampling..... 90
- Bibliography..... 99

List of Figures

Figure 2.1 Latent Semantic Analysis Decomposition.....	22
Figure 2.2 Probabilistic Latent Semantic Analysis Graphical Notation	24
Figure 2.3 Topic Model Decomposition	28
Figure 3.1 Conversation Focus Detection Framework	35
Figure 3.2 Direct Link and Quote	37
Figure 3.3 Lexical Similarity	38
Figure 3.4 Illustration of the generative process and the problem of statistical inference underlying topic models	45
Figure 3.5 The graphical model for the topic modeling in plate notation.....	46
Figure 3.6 Non-Markov continuous – time topic model.....	49
Figure 3.7 Number of Daily Posts from 1/9/2006 to 31/12/2006	51
Figure 3.8 An Interesting Board in the Forum (Digital Camera).....	52
Figure 3.9 A Post in a Forum (Digital Camera).....	52
Figure 3.10 Number of Threads in different Thread Length	53
Figure 3.11 Number of Threads vs Different Number of Quotes	54
Figure 3.12 Number of Threads vs Different Number of Hours.....	54
Figure 3.13 Number of Posts in the First 100 different Number of Words	55

Figure 3.14 Experimental Flow.....	56
Figure 3.15 Sample of different Link Generated in Weighted Directed Graph.....	59
Figure 3.16 Sample of Extracted Words and Generated Graph.....	62
Figure 3.17 Example 1 of LDA and its variants by approximated using Gibbs Sampling.....	63
Figure 3.18 Example 2 of LDA and its variants by approximated using Gibbs Sampling.....	64
Figure 4.1 General Structure of a Discussion Thread.....	69
Figure 4.2 Schema of a Document Post.....	69
Figure 4.3 Internal Structure of Document Post Content.....	70
Figure 4.4 A Sample Pair-wise Interaction Graph, Variable D=Document, A=Author, S=Sentiment, M=Mood.....	72
Figure 4.5 Flow of the DSM experiments.....	73
Figure 5.1 Positivity of Mobile Phone Brands.....	80
Figure 5.2 Positivity of Tsang and Leung in Chief Executive Election.....	80

List of Tables

Table 2.1 Sample of document-word matrix.....	21
Table 3.1 Example of Weighting Value assigned by Voter.....	57
Table 3.2 Example of Squared Difference of Weighting Value	57
Table 3.3 Results of the Mean Square Error of different Feature-oriented Link Generation	58
Table 3.4 Results of the Mean Square Error of different Graph-based Ranking Algorithm	60
Table 4.1 Number of Features Summary	74
Table 4.2 The Weighting of Distance Level between features, where $b = 2$ in $w_j = bx_{ji}$	75
Table 4.3 The Precision on Two Domains of Data.....	75

List of Algorithms

Algorithm 3.1 Pseudo Code of Bag of Words Extraction (Main Function)	43
Algorithm 3.2 Pseudo Code of Bag of Words Extraction (Graph Ranking Function)	43
Algorithm 3.3 Pseudo Code of Bag of Words Extraction (Construct Graph Function)	44
Algorithm 3.4 Latent Dirichlet Allocation Generative Process	47
Algorithm 3.5 CorrLDA2 Generative Process	48
Algorithm 3.6 Topic-time Model Generative Process	50

Chapter 1

Introduction

1.1 Problem and Motivation

With the invention of the second generation of Internet-based service (Web 2.0, which is coined by O'Reilly Media in 2004), the amount of social media such as blog, forum and newsgroup is increasing dramatically. This fast growing thread is now a plentiful resource for investigations. It offers an unprecedented opportunities and challenges to researchers of many different work sectors. For example, marketing analyst may concern about what consumers say in the web regarding the products and services. These valuable word of mouth resources provide a wholly new ways for analysis.

The nature of social media is semi-structured and written by a human readable language. If a marketing analyst conducts a survey on a certain brand of mobile phone, the web cannot answer questions like what type of mobile phone topic consumers talk the most. This is because freeform text cannot be processed easily by machines. They can only be understood effectively by humans. To finish this task, browsing a ton of data is necessary. However, it is hardly to be completed manually as the amount of data is huge. They can only be processed efficiently by machines[1].

To minimize this paradox, scholars believe we can adopt those heuristics techniques derived from information retrieval, machine learning and data mining. In the case of information retrieval, a set of techniques has been proposed to process textual data. From the basic Bag of Words (BOW) representation (70's)[2] to the generative based Probabilistic Latent Semantic Analysis (PLSA) (90's)[3], the study of the problems of textual analysis has apparently moved to more powerful and sophisticated statistical learning framework.

In contrast, a recent surge of research on machine learning over the last decade has been accompanied by many important developments in the underlying algorithms and techniques. For example, Bayesian methods have grown from a specialist niche to become mainstream[4]. Meanwhile, Graphical Probabilistic Model (GPM) has emerged as a general framework for describing, visualizing and explaining complex problems. In addition, the applicability of Bayesian methods has been greatly enhanced by the development of a set of approximate inference algorithms such as variational Bayes[5] and Markov chain Monte Carlo[6]. When applying these techniques in textual data, we may formulate topical problems (e.g. hot topics in a forum) with the latent variable model and obtain quality results in tractable manner. We may also infer the relationships between different textual type variables (e.g. author, entity, word, and sentiment) with the Markov random fields.

Characterized by the diversity of the Internet usage, textual data is no longer pure flatted text in the World Wide Web today. Machine readable structural text like the XML or RSS feed, Webpage formatted semi-structured text like the HTML, and threaded discussion formatted web forum text are those common textual data. On the one hand, the XML and HTML texts have

already been made a great attention both in academic and industrial interests. On the other hand, one of the less concerns but massive data in today is generated from web forums.

The web forum, similar to the USENET, is a bulletin board commonly used by users to exchange ideas, publish topics, or simply send replies via the HTML based browser. Since almost all computers are equipped with the pre-installed browser and can be easily accessed, the web forum has become more popular, and is considered as a significant contributor of user generated content.

To analyze the web forum, one of the easiest ways is to directly convert a post or a thread as bag of words (BOW) vector space representation and perform one of the graphical probabilistic modeling e.g. latent variables modeling for topical modeling or Markov random fields (for non-topical modeling). However, the transformation of bag of words of a threaded text may lead to a serious loss of important information, making the analysis and mining process ineffective. There are at least two problems:

- 1) First, the transformation totally ignores the relationships between posts and replies. In threaded text, not all the contents (posts) are equally important. Threaded text usually involves two or more parties, discussing an interesting topic, and each party conveys certain information to the topic during the turn by turn interaction. Each turn does not contribute equal important information to the topic. In other words, the importance measure of each word is different from pure flattened text.
- 2) Second, vector space representation ignores the distributional similarity between different textual type variables (e.g. entity, word, and sentiment). It is not robust to represent the information entropy, in terms of co-occurrence, between two variables in threaded text.

For instance, a replying post contains a positive sentiment is correlated with a product name entity which only appears in the master thread of current post. Such kind of correlation cannot be counted in co-occurrence because two variables are placed in two different documents independently.

1.2 Objectives

In view of the two problems stated in Section 1.1, we have come up with following two objectives. The first one is:

- 1) to propose a set of preprocessing frameworks to facilitate threaded discussion analysis with graphical probabilistic modeling

To facilitate the analysis, we need to solve the above mentioned problems. That is a concrete threaded text representation model is needed. The implicit and explicit relationship between posts and replies should be well captured and measurable. With respect to the second problem, we need to develop a new similarity model to capture the intra-post relationship. In view of the recent development of different graphical probabilistic models for textual analysis, our second objective is

- 2) to analyze different graphical probabilistic models with threaded discussion textual data. For instance, Latent Dirichlet Allocation[7], its variants (topic-entity[8], topic-time model[9], and etc.) and Markov random fields[10] are considered

1.3 Contributions

We can summarize the contributions as below:

In this work, we can classify the solutions of threaded textual problems into topical and non-topical modeling. In topical problem, this is the first work to propose a preprocessing framework to select bag of words by conversation focus detection. The framework is composed of two parts – constructing a directed acyclic graph (DAG) by a set of feature link generation functions to represent a threaded text, and performing different graph based ranking algorithms to extract a list of words (word-thread matrix) for latent variable modeling. In non-topical modeling, a distributional similarity model is first introduced to extend the multi-modality clustering with positional and link information for the unique characteristic of user generated data.

1.4 Outline of the Thesis

In this thesis, a set of preprocessing frameworks with graphical probabilistic models and its application to extracted social media data of Hong Kong web forums are reported. The thesis is organized into five chapters. A literature review of text mining, Bag of Words (BOW), Latent Semantic Analysis (LSA) representation, and emerged framework of Graphical Probabilistic Model (GPM) is provided in Chapter 2. Chapter 3 introduces a conversation detection algorithm for weighing the importance of bag of word in topical modeling. In Chapter 4, the Distributional Similarity Model (DSM) is proposed to facilitate the relationship analysis of different textual type variables in non-topical modeling. The final chapter gives the conclusions and future works.

Chapter 2

Graphical Probabilistic Modeling for Textual Data

2.1 Text Mining

Text mining is an interdisciplinary field which induces on information retrieval, data mining, machine learning, statistics and computational linguistics. It is the process of deriving high quality information from text though the detection of patterns and trends. Typically, text mining tasks include text categorization, text clustering, concept/entity extraction, sentiment analysis, document summarization and entity relation modeling. In practical applications, the world largest text mining project is the Echelon surveillance system owned by the governments of Australia, Canada, New Zealand, the United Kingdom and the United States. It is capable of interception and content analysis of fax, email and other data traffic globally through the interception of communication bearers including satellite transmission, public switched telephone networks and the Internet[11].

With the advanced development of the Internet, the amount of textual data has been increasing dramatically. The goal to find short descriptions of the members of such textual collection that enable efficient processing and preserving the essential statistical relationships has increasing

and compelling needs. Therefore, developing text mining techniques are the most popular research work in recent years.

We may think that text mining problems can be solved in use of conventional data mining techniques. However, the answer is partially correct. The fundamental difference between text mining and conventional data mining is that text is not structural tabulate data which generally to be assumed before. Curse of dimensionality in text mining makes most of the traditional data mining techniques ineffectively and inefficiently. Although, scholars propose a set of subspace mining algorithms [12, 13] to cope with this problem, the maze nature of natural language, for instance, polysemy and stop words, makes the mining results hard to interpret. Consequently, we are still looking for a set of techniques to obtain high quality information from textual data with tractable computational time.

In this chapter, we go through a review which scholars proposed in text mining: bag of words representation, latent semantic analysis, probabilistic latent semantic analysis and finally the generative graphical probabilistic modeling for textual analysis.

2.2 Bag of Words (BOW) Representation

Bags of words representation[2] (also called vector space representation), the basic methodology proposed by IR researchers for representing text corpora, is a simple and easy way to transform unstructured text to structured tabulated data by representing a text as an unordered collection of words, disregarding its grammar and the word order information. In the transformation, the text is converted to a document-word matrix, in which the row is the document id and the column is the word id. Commonly, the value shows how many times a word appears in the particular document. For instance,

D1 = "I like apple", D2= "I hate apple apple"

Then the document-word matrix would be:

Table 2.1 Sample of document-word matrix

	I	like	hate	apple
D1	1	1	0	1
D2	1	0	1	2

More sophisticated weights have been proposed by different researcher. One typical example would be the Term Frequency - inversed Document Frequency (tf-idf)[2]. This is a statistical measure used to evaluate how important a word is to a document in a collection. The idea is that, the importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the text. The formulation is as follows:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \quad [2.1]$$

where $n_{i,j}$ is the number of occurrences of the considered term in document d_j , and the denominator is the number of occurrences of all terms in document d_j .

$$idf_i = \log \frac{|D|}{|\{d_j: t_i \in d_j\}|} \quad [2.2]$$

with $|D|$ is total number of document in the corpus and $|\{d_j: t_i \in d_j\}|$ is the number of documents where the term t_i appears.

Then,

$$tfidf_{i,j} = tf_{i,j} \times idf_i \quad [2.3]$$

2.3 Latent Semantic Analysis (LSA)

One of the problems in bag of words representation is that it does not consider synonymy and polysemy which generally being concerned in natural language processing. To address these shortcomings, IR researchers have proposed latent semantic analysis[14]. LSA uses a singular value decomposition of the document-word matrix to identify a linear subspace in the space of tf-idf features. In other words, it produces a set of concepts related to the documents and terms. LSA transforms the document-word matrix into a relation between the words and some concepts, and a relation between those concepts and the documents. Thus, the words and documents are now indirectly related through the concepts. The formulation is as follows:

$$A = USV^T \quad [2.4]$$

where S is the diagonal matrix of singular values and U, V are matrices of left and right singular vectors.

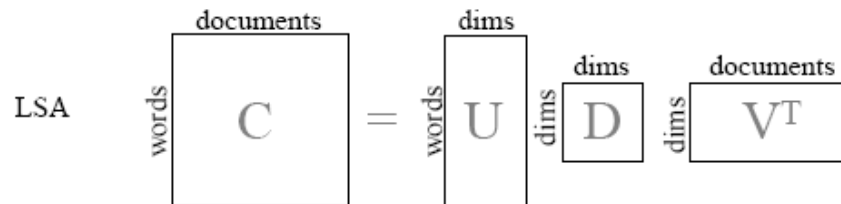


Figure 2.1 Latent Semantic Analysis Decomposition

The advantage of introducing the concept space is that it can be used to measure semantic similarity between documents, which is quite useful in text classification and clustering. First, it

captures the relations between words, e.g. synonymy and polysemy. Second, it reduces the dimensionality of the matrix e.g. combining some features. Thus, some mining techniques in conventional data mining can perform much efficient with LSA representation.

2.4 Probabilistic Latent Semantic Analysis (PLSA)

Probabilistic latent semantic analysis[3], also known as aspect model, is a statistical technique for the analysis of two-mode and co-occurrence data. PLSA evolved from latent semantic analysis by adding a sound probabilistic model. Compared to standard LSA which stems from linear algebra and downsizes the occurrence tables in using singular value decomposition, PLSA is based on a mixture decomposition derived from a latent variable model.

Considering observations in the form of co-occurrences document-word matrix, PLSA models the probability of each co-occurrence as a mixture of conditionally independent multinomial distributions as follows:

$$P(w, d) = \sum_c P(c)P(d|c)P(w|c) = P(d) \sum_c P(c|d)P(w|c) \quad [2.5]$$

The first formulation is symmetric, where w and d are both generated from the latent class c in similar ways, whereas the second formulation is asymmetric, in which each document d , a latent class is chosen conditionally to the document according to $P(c|d)$, and a word is then generated from that class according to $P(w|c)$.

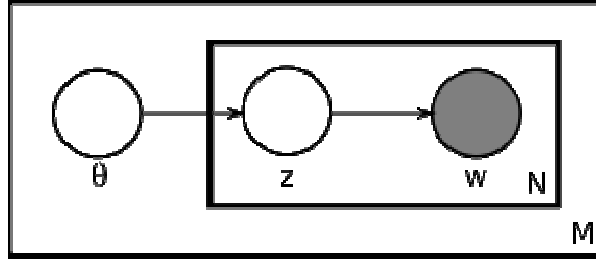


Figure 2.2 Probabilistic Latent Semantic Analysis Graphical Notation

The above graph model represents the PLSA model. θ_i is the topic distribution for document i , z_{ij} is the topic for the j th word in document i , and w_{ij} is the specific word. The w_{ij} are the only observable variables. Thus, a standard statistical inference technique can be used to infer the topics which describe the data.

Compared with LSA, PLSA is more directly to cope with the problems. It has important theoretical advantages over LSA. It is a generative data model. Thus, it directly minimizes word perplexity. It can also take advantage of statistical standard methods for model fitting, overfitting control, and model combination.

2.5 Graphical Probabilistic Model (GPM)

From the basic bag of words representation to the generative based PLSA, the study of the problems of textual analysis has apparently moved to more powerful and sophisticated statistical learning framework. In this part, we describe a recent very hot model in the field of machine learning and pattern recognition - Graphical Probabilistic Modeling (GPM)[4]. Before talking its functionalities, we review the Bayesian theorem as a background study.

2.5.1 Bayesian Theorem

Graphical probabilistic modeling is a combined field of probability theory and graph theory. The model is based on the generative power of probability, which come originally from Bayesian theorem[15].

The basis of probability theory is sum rule and product rule. When combining of them, it allows us to solve complex problems include some degree of uncertainty. In probability theory, there are two views of probabilities - frequentist view and Bayesian view. Frequentist view is to define an event's probability as the limit of its relative frequency in a large number of trials. For example, the probability of the coin landing heads is 0.53. In contrast, Bayesian view is to interpret the concept of probability as a measure of a state of knowledge. In other words, it is a quantification of degree of belief. For example, the probability that it will rain tomorrow is 0.2. It is because it is not possible to repeat tomorrow. The probabilities estimation is subjective and dependent on prior knowledge.

Recently, Bayesian methods have grown from specialist niche to become the mainstream. It is quite a popular method for pattern recognition and machine learning. Based on Bayesian theorem, we can train a classifier from a set of prior knowledge, or a set of patterns for clustering by Bayesian inference techniques. Below is the description:

Sum Rule:

$$P(A) = \sum_B P(A, B) \quad [2.6]$$

Product Rule:

$$P(A, B) = P(B|A)P(A) \quad [2.7]$$

From the Product Rule, we have

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad [2.8]$$

From the Sum Rule, the denominator can be written as

$$P(A) = \sum_B P(A|B)P(B) \quad [2.9]$$

In the above equations:

- $P(A)$ is the prior probability or marginal probability of A. It is prior in the sense that it does not take into account any information about B.
- $P(A|B)$ is the conditional probability of A, given B. It is also called posterior probability because it is derived from or depends upon the specified value of B.
- $P(B|A)$ is the conditional probability of B given A.
- $P(B)$ is the prior or marginal probability of B, and acts as a normalizing constant.

Thus, If A and B denote two events, $P(A|B)$ denotes the conditional probability of A occurring, given that B occurs. It is noted that the two conditional probabilities $P(A|B)$ and $P(B|A)$ are generally different. Bayes theorem gives a relations between $P(A|B)$ and $P(B|A)$ in [2.8]. As a result, we can use this rule to revise the beliefs (prior) to infer the posteriori.

In words, the posterior probability is proportional to the product of the prior probability and the likelihood. So, we can use the observed data B (prior) to maximize the likelihood $P(A|B)$ to infer the posterior $P(B|A)$. It is actually what we did in classification and clustering in the mathematical point of view.

2.5.2 Graph Model

Another part of graphical probabilistic modeling is graph. In GPM, a graph comprises vertexes connected by links. Each vertex represents a random variable (or group of random variables), and the links express probabilistic relationship between these variables. Thus, the graph captures the way of the joint distribution over all of the random variables.

By combining graph model with probability theory, we can use both functionalities to have new insights into the problems. Graphical probabilistic modeling offers several useful properties. It can allow us to visualize the structure of a probabilistic model and to design and motivate new model. Also, complex computations in terms of graphical manipulations can be easily expressed, for instance, perform inference and learning in sophisticated models.

2.5.3 Bayesian Networks and Markov Random Fields

There are two types of graph models. The first type is Bayesian networks, also known as directed graphical models in which the links of the graphs have a particular directionality indicated by arrows. Bayesian networks are useful for expressing causal relationships between random variables. For example, how document is generated from a set of words. The other major class of graphical models is Markov random fields, also known as undirected graphical models, in which the links do not carry arrows and have no directional significance. Markov random fields are

suited to expressing soft constraints between random variables. For example, how a person name is correlated with a sentiment expression.

2.5.4 Textual Analysis

Based on the description above, we now know that graphical probabilistic modeling provides a simple but powerful framework to represent independencies among random variables.

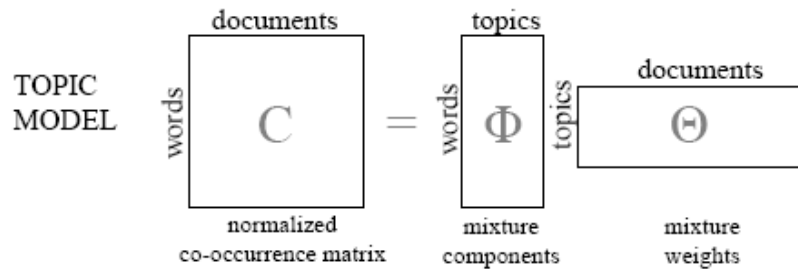


Figure 2.3 Topic Model Decomposition

In the recent development of textual analysis techniques, some scholars start to use GPM for modeling of text. One of the famous models is Latent Dirichlet Allocation (LDA)[7]. LDA is a graphical model that allows set of observations to be explained by unobserved groups which explain why some parts of the data are similar. For example, if observations are words collected into documents, it posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics.

In LDA, each document may be viewed as a mixture of various topics. This is similar to PLSA, except that in LDA the topic distribution is assumed to have a Dirichlet prior (Conjugate Prior). This prior, have a same functional form as the posterior, will make the inference easier and faster. In fact, PLSA is incomplete in that it provides no probabilistic model at the level of documents. In addition, LDA solves some problems existed in PLSA. First, in PLSA, the number

of parameters grows linearly with the size of the text, which leads to serious problems with overfitting. Second, it is not clear how to assign probability to a document outside of the training set. It means PLSA is not generative for a new document. The details of LDA and its variants for textual analysis will be discussed in the next chapter – Topical modeling.

Next, we will review a set of inference technique usually used in graphical model.

2.5.5 Approximate Inference

If we have a graphical model in hand, next issue is to calculate the conditional distribution what values some of unobservable variables take by inferring from observable variables.

Given a graphical model, we can answer all possible inference queries by marginalization. However, a graphical model has size $O(2^n)$, where n is the number of vertexes, and we have assumed each vertex can have 2 states. Thus, many statistical method is mathematical possible but not computational tractable. Below are some alternative solutions:

For a directed graphical model, we can use variable elimination to do marginalization efficiently[16]. The key idea is to push sums in as far as possible when summing out irrelevant terms. In addition, if we wish to compute several marginal at the same time, we can use Dynamic Programming (DP)[17] to avoid the redundant computation that would be involved if we used variable elimination repeatedly. However, not all the problems can be solved by variable elimination with Dynamic Programming.

In fact, many models of interest have large induced width, which makes exact inference very slow. Thus, we can resort to approximation techniques. Below are the two popular techniques:

1) Variational methods [5] -The simplest example is the mean-field approximation, which exploits the law of large numbers to approximate large sums of random variables by their means. In particular, it is to decouple all the vertexes and to introduce a new parameter, called variational parameter, for each vertex. Then, it iteratively update these parameters so as to minimize the cross-entropy (KL distance) between the approximate and true probability distributions.

2) Sampling (Monte Carlo) methods [6] - The simplest kind is importance sampling, where it draw random samples x from $P(X)$, the unconditional distribution on the hidden variables, and then weight the samples by their likelihood, $P(y|x)$, where y is the evidence. A more efficient approach in high dimensions is called Monte Carlo Markov Chain (MCMC), and includes as special cases Gibbs sampling which is very popular inference technique used in Latent Dirichlet Allocation.

2.6 Conclusion

In this chapter, we reviewed a list of techniques which try to provide sound statistical solution in textual analysis. From the basic assumption of Bag of Words (BOW) representation, scholars proposed a singular decomposition, Latent Semantic Analysis (LSA), to transform document-word matrix to document-concept and concept-word matrices. The emergence of Probabilistic Latent Semantic Analysis (PLSA) reveals the generative and inference power of Bayesian theorem. With the advanced development of the probabilistic model, scholars summarize and create a new framework to handle complex problems in a sophisticated and simple way – Graphical Probabilistic Model (GPM). GPM provides two types of model to solve casual relationship and soft constraints between random variables. With approximate inference techniques, we possess the ability to cope with exponential problems in tractable time. Based on

the revision of the GPM, the following two chapters, topical and non-topical modeling in threaded discussion analysis, describe several preprocessing frameworks and algorithms to better process threaded text in use of the graphical model.

Chapter 3

Topical Modeling in Threaded Discussion Analysis

3.1 Introduction

Recent popularity of latent variables modeling, like Latent Dirichlet Allocation (LDA)[7], makes a great attraction in research community. LDA is a graphical probabilistic model that formulates topical problem in massive textual data. The idea is that documents are mixtures of topics, where a topic is a probability distribution over words. This model is a generative model for documents. It specifies a simple probabilistic procedure in which documents can be generated. For example, to make a new document, one chooses a distribution over topics. Then, for each word in that document, one chooses a topic at random according to this distribution, and draws a word from that topic. Standard statistical techniques can be used to invert this process, inferring the set of topics that were responsible for generating a collection of documents. Besides, several LDA variants and extensions [7-9, 18-20] like topic-entity modeling, and non-Markov topic-time modeling provide us a complete and sophisticated framework to formulate many problems in textual data.

Today, one of the massive data generated from the Internet is web forums. The web forum, similar to the USENET, is a bulletin board commonly used by users to exchange ideas, publish

topics, or simply send replies via the HTML based browser. Since almost all computers are equipped with the pre-installed browser and can be easily accessed, the web forum has become more popular, and is considered as a significant contributor of the user generated content.

Compared with flatted discussion like instant chat, threaded discussion, as an electronic conversation method, is now a standard way to facilitate the communication between producer and consumer in the web forum. According to the conversation analysis theory[21], turn by turn asynchronized conversation between two or more members can allow conversation topic traceable and is not out of the original topic intention. Gradually, the extensive use of threaded discussion method in the web forum has generated tons of data in World Wide Web today.

To analyze the web forum in use of latent variable modeling, one of the easiest ways is to treat it as a flatted text. We can directly convert post or thread to a bag of words (BOW) vector space representation. Different level of analysis can be conducted, e.g., cross-forums, inter-threads or intra-thread. We can group extract similar topic in a set of forums in cross-forums analysis, extract the hottest topic inside a board by inter-threads analysis, and infer different topic in a large thread by intra-thread analysis.

However, this transformation of bag of words may lead to a serious loss of important information, making mining process ineffective. It totally ignores the relationships between posts and replies. In threaded text, not all the contents (posts) are equally important. Threaded text usually involves two or more parties, discussing an interesting topic, and each party conveys certain information to the topic during the turn by turn interaction. Each turn does not contribute equally important information to the topic. For example, one post may be quoted by many others replies and

another post may just be a reply to the previous turn and without other replies quoting it. In other words, the importance measure of each word is different from pure flattened text.

Based on this unique characteristic, we should use more concrete model to represent importance distributions between posts and replies. In fact, threaded text possesses an implicit structure like tree or even a graph between posts and replies. We argue that it is too simplified to use TFIDF or mutual information techniques to construct bag of words representation for threaded text topic modeling. Instead, we should rank posts by its importance and to extract important clues in the thread. Therefore, this chapter is to propose a preprocessing framework to facilitate feature selection process in topic modeling of threaded textual data.

In this chapter, we first describe the framework and algorithm to select bag of words by conversation focus detection. Second, a range of latent variables model for textual analysis are introduced. Next, basic analysis of our extracted Hong Kong web forum data is reported. Finally, some empirical results of conversation focus detection are evaluated and discussed.

3.2 Conversation Focus Detection Framework

A threaded text consists of a set of posts arranged in chronological order. The most informative or important one in this sequence is referred to as the conversation focus in some question answering domain. Different from common assumption of flattened text in most IR research, the relationships between posts and replies may differ in use of feature importance measurement method, e.g. TFIDF or mutual information. In this aspect, we describe a framework for word-thread matrix formation, where idea comes from [22-24].

In order to provide more representative bag of words for latent variables modeling, our framework is designed to measure both implicit and explicit relationships between posts and replies. The framework is composed of two parts in Figure 3.1. In the first part, a threaded text is converted to a directed acyclic graph (DAG) by a set of feature link generation functions, for instance direct link and quote, lexical similarity and authority. In the second part, different graph based ranking algorithms are performed to boost important keywords such as degree centrality, betweenness centrality[25], PageRank[26], and original HITS [27] used by Feng[24]. Our framework, then, extracts a list of words by weighting the importance ranking value with traditional feature selection method.

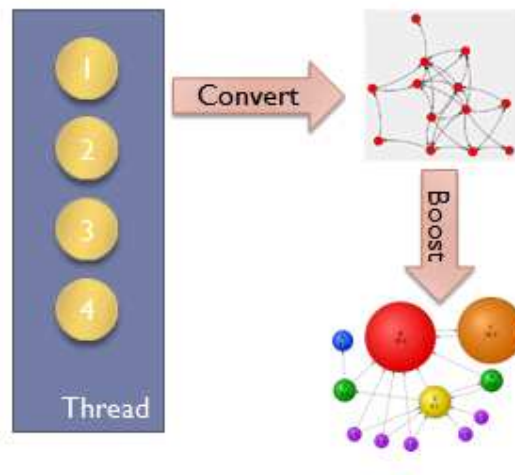


Figure 3.1 Conversation Focus Detection Framework

In the following part, we first define the thread representation by directed graph. Second, we introduce the feature-oriented link generations. Then, we describe a set of ranking algorithms to measure the importance value between each vertex. And, finally, we combine all of them as a pseudo code.

3.2.1 Thread Representation by Directed Graph

A threaded discussion consists of a set message posted in chronological order. Let each message represent by $m_i, i = 1, 2, \dots, n$. Then the entire thread is a directed graph that can be represented by $G = (V, E)$, where V is the set of vertexes (posts), $V = \{m_i, i = 1, \dots, n\}$, and E is the set of directed edges. The set V is automatically constructed as each message joins in the discussion. E is a subset of $V \times V$. We will discuss the feature-oriented link generation functions that construct the set E later.

We make use of lexical similarity and other similarity measurements in generating the links. Once a relation is identified between two posts, links will be generated using generation functions. When m_i is a message vertex in the thread graph, $F(m_i) \in V$ represents the set of vertexes that vertex m_i points to (i.e. children of m_i), and $B(m_i) \in V$ represents the set of vertexes that points to m_i (i.e. parents of m_i) [24].

3.2.2 Feature-oriented Link Generation

Conversation structure contains both explicit and implicit relationships between posts and replies. In linguistic research community, it has received a lot of attention to study of that such as discourse structure analysis, speech act analysis and etc.[28]. Feature-oriented link generation is a measurable function to generate explicit relationships between each post. The edges, thus, can be created with a certain value of weight to indicate its strength of such relationship. To measure the relationships between posts, we propose three functions – direct link and quote, lexical similarity and authority.

3.2.2.1 Direct Link and Quote

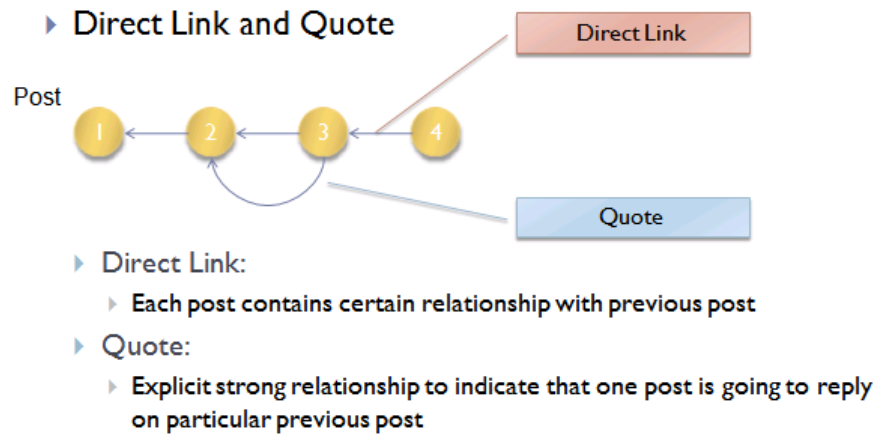


Figure 3.2 Direct Link and Quote

Direct link and quote is a function to generate relationship with original thread structure extracted from the forum. Two types belong to this class. The first is direct link in which we assume each post must have certain relationship, except the first post, with previous posts no matter how the relationship may be comparably weak. The second is quote in which the explicit quoting sometimes exists. This is a strong relationship to indicate that one post is going to reply on particular previous post. In example Figure 3.2, there are four posts. The label 1 is the first one posted in the thread and the label 4 is the last one. In the direct link generation, label 4 points to label 3, label 3 posts to label 2 and label 2 posts to label 1. Therefore, it forms a natural sequence indicating that latter posts go to reply the former posts. If explicit quotation exists like label 3 points to label 2, an additional link is generated to describe such relationship.

3.2.2.2 Lexical Similarity

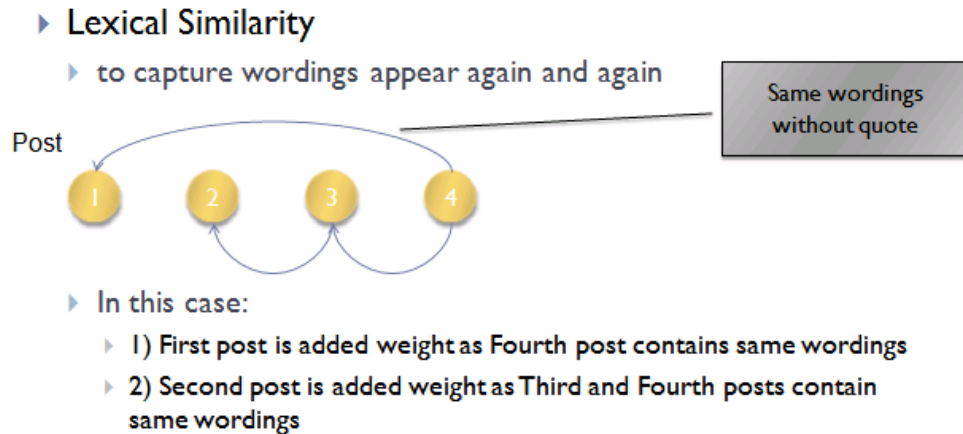


Figure 3.3 Lexical Similarity

Another link generation function is lexicon similarity between posts. In a discussion thread, people usually mention some wording appeared in previous posts. Such wordings appear again and again that usually gain its chance to become a member of bag of word representative. To capture this nature, we generate an edge to the corresponding first post which wordings appear. Thus, the centrality of that post will become higher. It means it conveys an indirect weighting to make words in first post to become more important. In example Figure 3.3, the label 1 is added weight as label 4 contains same wordings. And the label 2 is added weight as label 3 and 4 contain same wordings. Therefore, label 1 and label 2 posts become more important in the thread sequence.

To measure the lexical similarity of any pair of posts, two problems exist. First, it is computationally expensive. But we argue that post in a thread is relatively short compared with formal self-contained document. We will show our dataset analysis in later part to confirm this assumption. Second, word segmentation problem may exist in some language likes Chinese, Korean or Japanese. Some solutions for longest common substring problem[29] and hidden

Markov model [30, 31] for segmentation are effective solutions. The similarity ratio can be formulated as:

$$\text{Similarity Ratio} = \frac{\text{Same Words between Posts}}{\text{No. of Words in First Post}} \quad [3.1]$$

3.2.2.3 Authority

The third function is related to author relationship. To reflect the value of the post, poster active rate can be used. We can use whole forum or just intra-thread statistic to measure the relative active rate of a poster. The below equation shows the simplest way to measure poster active rate by intra-thread statistic:

$$\text{Poster Active Rate} = \frac{\text{No. of Posts of the Poster}}{\text{Total No. of Posts in the Thread}} \quad [3.2]$$

3.2.2.4 Weighted Directed Graph Generation

To generate a graph $G = (V, E)$ for a thread, as mentioned before, the set V is automatically constructed as each message joins in the discussion. The second step is to execute the feature-oriented link generation. E can be generated with direct link and quote, lexical similarity and authority functions discussed before. For the direct link and quote function, edges are pointed from the newer post to older post with an arrow. Similarly, if a newer post contains lexical similarity with older post, an edge will be added. The edge will also be weighted by the relative poster active rate according to [3.2]. If more than one links exist between two vertexes, the link will be joined by adding up the weighting values.

3.2.3 Graph-based Ranking Algorithms

In graph theory, there are various measures of the ranking within a graph that determine the relative importance of a vertex. Ranking algorithms assign values to each vertex or edge according to a set of criteria that reflect the structural properties of the graph. These criteria are generally intended to measure the influence, authority, centrality of a given vertex or edge. In our system, we used four measurements that are widely used in graph analysis: degree centrality, betweenness centrality, PageRank, and HITS.

3.2.3.1 Degree Centrality

The first, and simplest, is degree centrality. Degree centrality is defined as the number of links incident upon a vertex. If the network is directed, two separate measures of degree centrality are defined, namely indegree and outdegree. Indegree is a count of the number of ties directed to the vertex, and outdegree is the number of ties that the vertex directs to others. In threaded discussion, high indegree means there are many quotes refer to or similar with other posts[25].

3.2.3.2 Betweenness Centrality

Betweenness centrality is a measure of a vertex within a graph[25]. Vertices that occur in many shortest paths between other vertices have higher betweenness than those that do not.

For a graph $G: = (V, E)$ with n vertices, the betweenness $C_B(v)$ for vertex v is:

$$C_B(v) = \sum_{\substack{s \neq v \neq t \in V \\ s \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad [3.3]$$

where σ_{st} is the number of shortest geodesic paths from s to t , and $\sigma_{st}(v)$ is the number of shortest geodesic paths from s to t that pass through a vertex v . This may be normalised by dividing

through by the number of pairs of vertices not including v , which is $(n - 1)(n - 2)$. Calculating the betweenness centralities of all the vertices in a graph involves calculating the shortest paths between all pairs of vertices on a graph. That is, it is very costly to compute. A faster algorithm for betweenness centrality is introduced by Brandes in [32]. It requires $O(n + m)$ space and runs in $O(nm + n^2 \log n)$ time on weighted graph, where n is the number of actors and m is the number of links.

3.2.3.3 PageRank

PageRank[26] is originally a link analysis algorithm that assigns a numerical weighting to each element of a hyperlinked set of documents, with the purpose of measuring its relative importance within the set. We adjust that the link is generated by our feature-oriented generation functions. Thus, PageRank algorithm can help us to rank the importance of a set of posts within a thread.

PageRank algorithm ranks each vertex in a Bayesian network according to its stationary probability. It is a variant of the eigenvector centrality measure. Eigenvector centrality is also a measure of the importance of a vertex in a graph. By using the adjacency matrix to represent connection strengths, eigenvector centrality can be found.

3.2.3.4 HITS

HITS [27] is a link analysis algorithm. Different from PageRank, it rates web pages for their authority and hub values. Authority value estimates the value of the content of the page. Hub value estimates the value of its links to other pages. Finally, the combined values can be used to rank web page importance. The weighted iterative updating computations are recalled as follows:

$$\text{hub}^{r+1}(m_i) = \sum_{m_j \in B(m_i)} W_{ij} * \text{authority}^r(m_j) \quad [3.4]$$

$$authority^{r+1}(m_i) = \sum_{m_j \in B(m_i)} W_{ji} * ub^r(m_j) \quad [3.5]$$

where r and $r + 1$ are the numbers of iterations.

With the above graph-based ranking algorithms, we can rank a set of posts in an importance order. The importance ranking value obtained is used to weight the bag of word extracted in particular post. We can also combine this value with the traditional feature selection measurements to calculate the bag of word representative for a discussion thread.

3.2.4 Algorithmic Framework

With conversation focus detection framework as a preprocessing step, we can take into account importance ranking value of a particular post. Two approaches can be used to extract bag of words when introducing this step. One is simply to use the importance ranking value of the ranked post as the weighting value of its words. The other is to combine the importance ranking value with the traditional feature selection measurements like TFIDF or mutual information.

Below is the Pseudo code of Bag of Words Extraction with Conversation Focus Detection. The inputs are a list of threads and a boolean value to indicate whether the importance ranking value is combined with traditional feature selection measurements. After the processing of graph construction and graph ranking subroutines, a list of bag of words with different score can be obtained and the word-thread formation is constructed.

```

Function Main(){
Input:
  A List of Threads
  Boolean Combine with TFIDF
Output:
  A List of Bag of Words <String, Score> per Thread
Begin:
  For each Thread{
    Graph = Call Construct_Graph(Thread);
    List of Ranking Score <Post, Score> = Call Graph_Ranking();
    If (Combine with TFIDF){
      // TFIDF Calculation
      All Content = Join all the post content;
      Bag of Word = Segmentation(All Content);
      A List of TFIDF Bag of Words <String, TFIDF Score> =
        TFIDF(Bag of Word);
      Combine TFIDF Scored Bag of Words with Ranking Score in
        each Post;
    } else {
      For each Post {
        Bag of Word = Segmentation(Post Content);
        Combine Bag of Word with Ranking Score;
      }
    }
  }
  Sort Bag of Word with Combined Score;
End;
}

```

Algorithm 3.1 Pseudo Code of Bag of Words Extraction (Main Function)

```

Function Graph_Ranking (){
Input:
  Weighted Directed Graph,
  Type of Algorithm
Output:
  List of Ranking Score <Post, Score>
Begin:
  Switch (Type of Algorithm)
  Case (Degree Centrality){
    Run Degree Centrality Ranking;
    Break;
  }
  Case (Betweenness Centrality){
    Run Betweenness Centrality Ranking;
    Break;
  }
  Case (HITS){
    Run HITS Ranking;
    Break;
  }
  Case (PageRank){
    Run PageRank Ranking;
    Break;
  }
End;
}

```

Algorithm 3.2 Pseudo Code of Bag of Words Extraction (Graph Ranking Function)

```

Function Construct_Graph(){
Input:
  Array of Posts,
  Boolean withDirectLinkEdge,
  Boolean withQuoteEdge,
  Boolean withLexicalSimilarityEdge,
  Boolean includeAuthorty
Output:
  A Weighted Directed Graph
Begin:
  Generate Vertex for each post;
  If (withDirectLinkEdge){
    For Each Vertex{
      Assign the Authority value to the Vertex
    }
  }
  If (withDirectLinkEdge){
    For Each Vertex except the first post{
      Generate Edge with previous Vertex;
    }
  }
  If (withQuoteEdge){
    For Each Vertex except the first post{
      If Quote exist in the Vertex{
        For Each Vertex previously{
          If the content is same with Quote{
            Generate Edge with that post;
          }
        }
      }
    }
  }
  If (withLexicalSimilarityEdge){
    For Each Vertex except the first post{
      Extract all the words;
      For Each Vertex previously{
        If same words exist{
          Generate Edge with that post with certain weighting;
        }
      }
    }
  }
End;
}

```

Algorithm 3.3 Pseudo Code of Bag of Words Extraction (Construct Graph Function)

3.3 Latent Variable Modeling

With the word-thread formation constructed by the proposed conversation focus detection framework, we introduce some latent variables modeling to extract high level topical information from the web forum in this part.

A latent variable model is a model that relates a set of variables to a set of latent variables which are defined as variables that are not directly observed but are rather inferred from other variables that are observed and directly measured. One advantage of using latent variables modeling is that

it reduces the dimensionality of data. A large number of observable variables can be aggregated in a model to represent the underlying concept, making it easier for humans to understand the data.

In recent development of text mining, generative topic modeling attracts a great attention in machine learning community. In fact, generative topic modeling is a kind of latent variables modeling[3, 7]. A generative model for documents is based on simple probabilistic sampling rules that describe how words in documents might be generated on the basis of latent (random) variables. When fitting a generative model, the goal is to find the best set of latent variables that can explain the observed data (for instance, observed words in documents), assuming that the model actually generated the data.

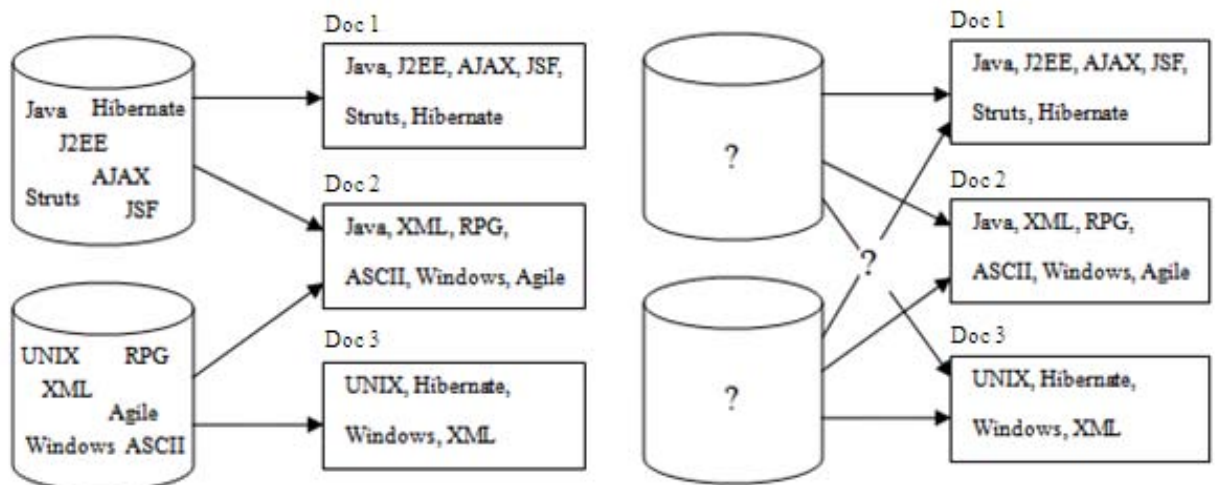


Figure 3.4 Illustration of the generative process and the problem of statistical inference underlying topic models

Above figure illustrates the topic modeling approach in two distinct ways: as a generative model and as a problem of statistical inference. On the left, the generative process is illustrated with two topics. Topics 1 and 2 are thematically related to Java and XML and are illustrated as bags

containing different distributions over words. Different documents can be produced by picking words from a topic depending on the weight given to the topic. For example, documents 1 and 3 were generated by sampling only from topic 1 and 2 respectively while document 2 was generated by an equal mixture of the two topics.

The right panel illustrates the problem of statistical inference. Given the observed words in a set of documents, we would like to know what topic model is most likely to have generated the data. This involves inferring the probability distribution over words associated with each topic, the distribution over topics for each document, and, often, the topic responsible for generating each word.

3.3.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [7] is one of the latent variable models for topical problem. In LDA, each document may be viewed as a mixture of various topics. It is different from probabilistic Latent Semantic Analysis (PLSA) [3] that the topic distribution is assumed to have a Dirichlet prior. As discussed in Section 2.5.4, this prior have a same functional form as the posterior, will make the inference easier and faster.

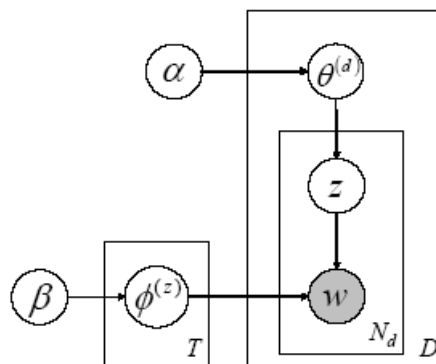


Figure 3.5 The graphical model for the topic modeling in plate notation

In the graphical model depicted in Figure 3.5, LDA is used for topic extraction in a collection of threaded text.

The generative process can be described as:

- For all d threads sample $\theta_d \sim \text{Dir}(\alpha)$
- For all t topics sample $\phi_t \sim \text{Dir}(\beta)$
- For each of the N_d words w_i in thread d
 - Sample a topic $z_i \sim \text{Mult}(\theta_d)$
 - Sample a word $w_i \sim \text{Mult}(\phi_{z_i})$

where D is the number of documents, T is the number of topics, N_d is the number of extracted words in thread d . α and β are Dirichlet smoothing parameters, θ is the topic-thread distribution, ϕ is the word-topic distribution, z_i is a topic, w_i is a word.

Algorithm 3.4 Latent Dirichlet Allocation Generative Process

To infer the generative process, we are actually to solve the following equation by computing the posterior distribution of the hidden variables given a document:

$$p(\theta, z | \mathbf{w}, \alpha, \beta) = \frac{p(\theta, z, \mathbf{w} | \alpha, \beta)}{p(\mathbf{w} | \alpha, \beta)}. \quad [3.6]$$

However, the distribution is intractable to compute:

$$p(\mathbf{w} | \alpha, \beta) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^k \theta_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^N \sum_{i=1}^k \prod_{j=1}^V (\theta_i \beta_{ij})^{w_n^j} \right) d\theta, \quad [3.7]$$

It is because it faces the coupling between θ and β in Equation [3.7]. Exact inference to reverse this generated process is intractable. Therefore, we need to resort approximate inference techniques, e.g. mean-field approximation (a variational method) [5] or Gibbs sampling (a

Markov chain Monte Carlo method)[6]. In our system, we used Gibbs sampling which is outperformed in many studies.

3.3.2 Topic-Entity Modeling

The LDA model is highly modular and can therefore be easily extended. One of that is topic-entity modeling[8]. The key idea is that, documents usually convey information about who, what, when and where. If we want to learn and summarize this entity-topic relationship in a set of documents, we can extend the LDA model with entity variables.

There are at least four models to formulate this problem: conditionally-independent LDA (CI-LDA), SwitchLDA, CorrLDA1, and CorrLDA2 model. The following is the CorrLDA2 generative process:

CorrLDA2:

- For all d threads sample $\theta_d \sim \text{Dir}(\alpha)$
- For all $t = 1 \dots T$ word topics sample $\theta_t \sim \text{Dir}(\beta)$ and $\varphi_t \sim \text{Dir}(\gamma)$
- For all $t = 1 \dots T$ entity topics sample $\tilde{\theta}_t \sim \text{Dir}(\tilde{\beta})$
- For each of the N_{w_d} words w_i in thread d
 - Sample a topic $z_i \sim \text{Mult}(\theta_d)$
 - Sample a word $w_i \sim \text{Mult}(\theta_{z_i})$
- For each of the $N_{\tilde{w}_d}$ entities \tilde{w}_i in thread d
 - Sample a supertopic $x_i \sim \text{Unif}(z_{w_1} \dots z_{w_{N_{w_d}}})$
 - Sample a topic $\tilde{z}_i \sim \text{Mult}(\varphi_{x_i})$
 - Sample an entity $\tilde{w}_i \sim \text{Mult}(\tilde{\theta}_{\tilde{z}_i})$

where v_i is a word or entity, while w_i is a word, \tilde{w}_i is an entity, T are word topics and \tilde{T} are entity topics.

Algorithm 3.5 CorrLDA2 Generative Process

With these four models using Gibbs sampling approach, we can directly learn the relationship between topics discussed in threaded discussion and entities mentioned in each thread.

3.3.3 Topic-Time Modeling

To extract topics by using LDA, one of the problems is that it cannot capture the topic structure over time. To overcome this problem, one may use Markov assumption on state dynamics or discretization of time. However, this makes a risk of inappropriately dividing a topic into two when there is a brief gap in its appearance in Markov model and being undecidable to find out proper value for discretization of data by time. Wang proposed to simply introduce a random variable to address this problem, i.e. a non-Markov continuous –time topic model[9] as shown in Figure 3.6 where Ψ_z is the beta distribution of time specific and t_{d_i} is the timestamp associated with the i -th token in the document d .

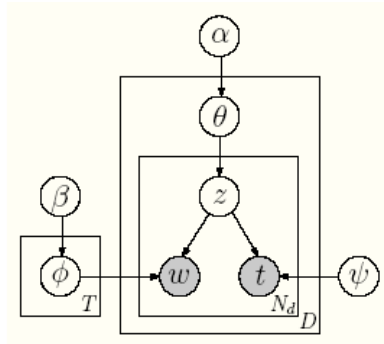


Figure 3.6 Non-Markov continuous – time topic model

The generative process is for such a non-Markov continuous – time topic model can be summarized by below:

- Draw T multinomials θ_z from a Dirichlet prior β , one for each topic z
- For each thread d , draw a multinomial θ_d from a Dirichlet prior α , then for each word w_{d_i} in thread d
 - Draw a topic z_{d_i} from multinomial θ_d
 - Draw a word w_{d_i} from multinomial $\theta_{z_{d_i}}$
 - Draw a timestamp t_{d_i} from $\mathbf{Beta}(\Psi_{z_{d_i}})$

Algorithm 3.6 Topic-time Model Generative Process

When using this topic-time model, we can discover topics that simultaneously capture word co-occurrences and locality of those patterns in time. It means that we can avoid to carelessly group topic only based on word co-occurrence. And it is able for us to discover the topic relationship with time, for instance, topic thread.

3.4 Experimental Results

In this section, an evaluation of our proposed methods is reported. First, we have conducted a basic analysis of our datasets to reveal the unique nature of threaded text. We measure the average thread length, number of quotes, number of words, and life span of our testing thread. Second, we analyze the performance of different feature-oriented link generations and estimate the weighting values. Third, we test the quality of extracted words by the four graph-based ranking algorithms and the baseline TFIDF method with our manually labeled data. Finally, we will show some results of LDA, topic-entity modeling and topic-time modeling.

3.4.1 Datasets

We worked on 12 Hong Kong web forums, including phonehk.com, uwants.com, discuss.com.hk, between 2006-09-01 and 2006-12-31. Numbers of daily posts are shown in Figure 3.7. They all use Discuz! engine developed by Comsenz Inc. in PRC. We developed a set of crawlers to extract the data from its archive.

Basically, for each of the post, we can extract the following information: title, publish time, author, content, forum name, and board name as shown in Figure 3.8. We assigned a unique document id and its master document id (the first document id of a thread) to reserve the post and replies relationships of a thread.

Inside the content, a semi-structured text may include quote, discuz! code, smilies code, and even html code, e.g. in Figure 3.9. Thus, images and links can also be embedded in the document.

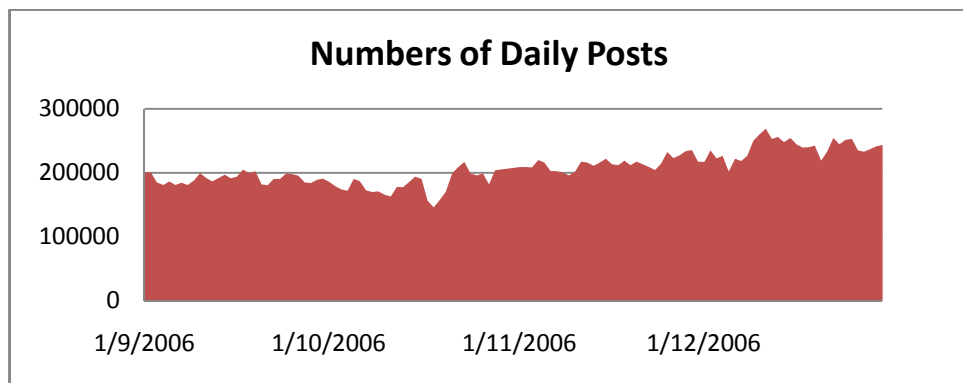



Figure 3.7 Number of Daily Posts from 1/9/2006 to 31/12/2006

Topic Name

Time of Last Post

論壇主題	作者	回覆 / 查看	最後發表
[諮詢意見] 女仔想用半專業機  1 2 3 4 5 6 	 ka266 2007-9-28	87 / 15703	2008-3-30 03:32 PM by wintere
[諮詢意見] 最新 28mm dc 的 投票 	 yochau 2008-3-23	13 / 2303	2008-3-30 03:29 PM by gdele
[諮詢意見] canon ixus 960is VS canon ixus 970is  NEW!	 initial_d@jay 2008-3-30	0 / 2	2008-3-30 03:23 PM by initial_d@jay
[諮詢意見] 我最最最最喜愛DC  1 2 3 	 佐中 2008-1-5	36 / 7298	2008-3-30 03:14 PM by 14號
[諮詢意見] 買邊部相機好~~比d意見啊~~~  1 2 3 4 5 6 	 darkmo 2007-7-23	79 / 20969	2008-3-30 03:13 PM by sony_boy
[諮詢意見] Canon 860IS, Fujifilm F100fd 	 ppc0101 2008-3-27	7 / 768	2008-3-30 02:57 PM by takakoh
[諮詢意見] 請問各位認  ?  1 2 	 undepong88 2008-2-21	19 / 406	2008-3-30 02:55 PM by chinmandk
[諮詢意見] gX100 VS 	 kit2_5 2007-7-22	59 / 16230	2008-3-30 02:55 PM by ch


Figure 3.8 An Interesting Board in the Forum (Digital Camera)

nicholas1793
原始人


積分 223
帖子 99
現金 5200 U幣
存款 5232 U幣
報價次數 0 次
閱讀權限 10
註冊 2007-1-18

發表於 2007-7-25 10:03 AM 資料 文集 短消息 #7

QUOTE:

原帖由 OneZero 於 2007-7-24 11:21 PM 發表
係咪我out得滯,f47咩泥ga? 

jifilm FinePix F47 fd 產品簡介
基本規格 BASIC SPECIFICATION

推出日期 :	約 2007 年 06 月
感光元件像素 :	903 萬像素

Figure 3.9 A Post in a Forum (Digital Camera)

3.4.2 Basic Analysis of Threaded Text

From the extracted information, the average thread length, number of quotes, number of words, and life span of our testing thread can be easily measured. The results are reported below:

3.4.2.1 Number of Threads vs Different Thread Length

Data spanned period: 2006-09-01 – 2006-12-31

Number of posts: 6,552,906

Number of threads: 366,550

Average thread length: $6,552,906/366,550=17.87$ posts per thread

Range: 0 to 35,284 posts in a thread (Thread Length)

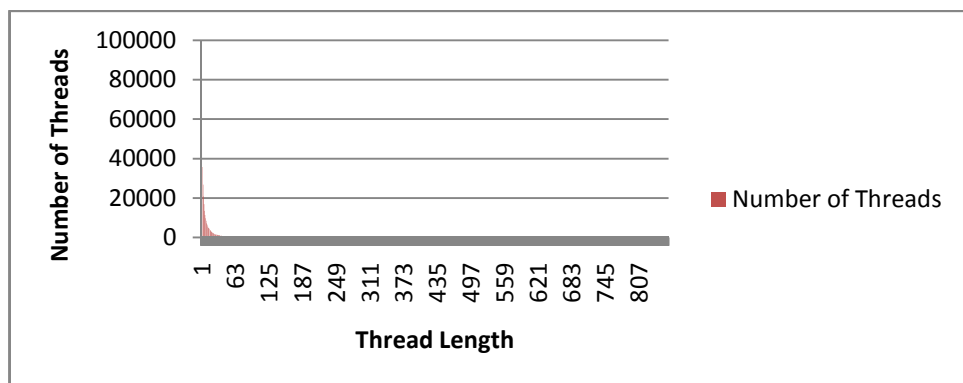


Figure 3.10 Number of Threads in different Thread Length

3.4.2.2 Number of Threads vs Different Number of Quotes

Number of quotes in total 366,550 threads: 2,876,162

Average number of quotes inside a thread: $2,876,162/366,559=7.84$ quotes per thread

Range: 0 to 31,344 quotes in a thread

Number of posts per quote: $17.87/7.84=2.28$ posts

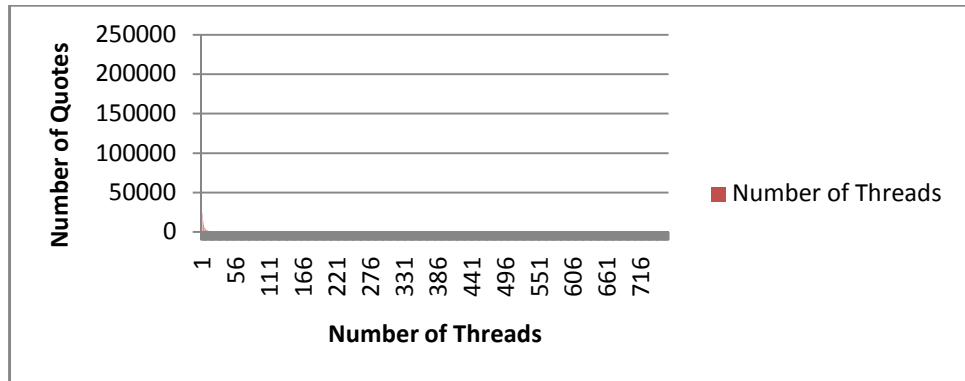


Figure 3.11 Number of Threads vs Different Number of Quotes

3.4.2.3 Number of Threads vs Different Number of Hours

Average number of hours of a thread: 319.86 (13.32days)

Range: 0 to 5806 hours (0 to more than 240 days) last in a thread

Average number of hours per post: $319.86/17.87=17.90$ hours per post

Average number of hours per quote: $319.86/7.84=40.80$ hours per quote

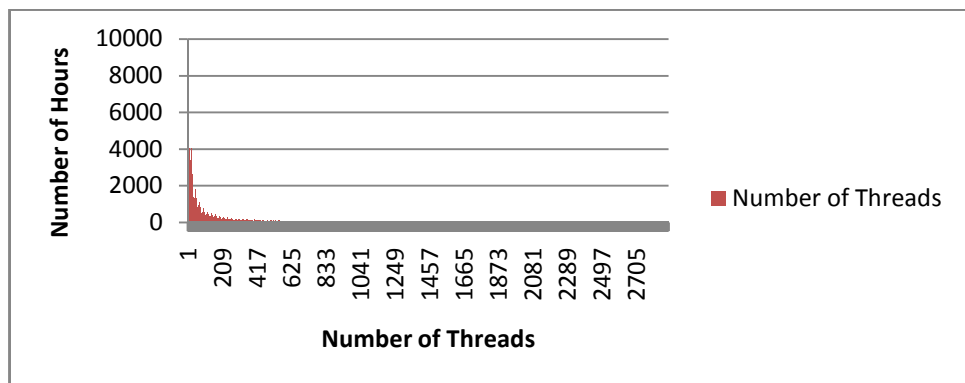


Figure 3.12 Number of Threads vs Different Number of Hours

3.4.2.4 Number of Posts vs Number of Words and Characters

Average number of words of a post: 5.47 words

Average number of characters of a post: 67.27 characters

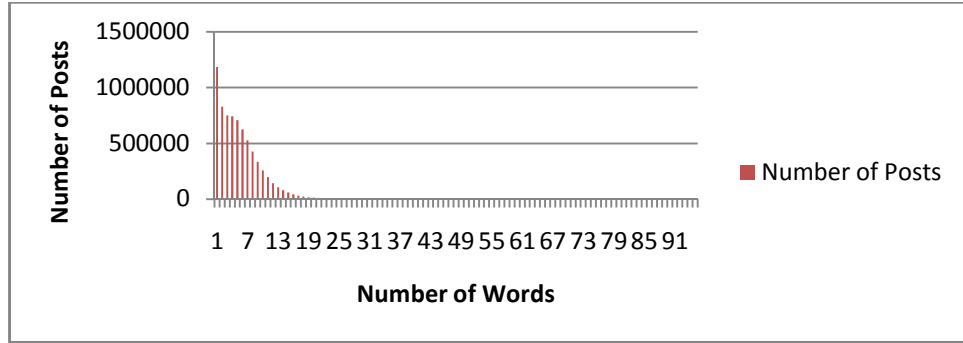


Figure 3.13 Number of Posts in the First 100 different Number of Words

3.4.2.5 Nature of Threaded Text

As shown in the above Figure 3.10 - Figure 3.13, most of the threads in the web forum are composed by a set of posts (not a single document) with a life span around 13.32 days. Many of them have under 30 posts / replies and the average is 17.87 (where 7.84 are quotes). Post is generally short, just 5.47 words (in term of Chinese) on average or under 20 words.

This basic analysis of the data supports what we have stated in the previous sections. First, post in a thread is relatively short (5.47 words) compared with traditional textual document. Second, quotes appears frequently (a quote appears in every 2.28 posts in average), it means that the relationships between posts / replies is very strong. That supports why we need a new thread model with feature-oriented link generations. Third, the life span of a thread is generally short (13.32 days). Thus, topic-time modeling is necessary so that we can avoid to irrelevantly group of topics only based on word co-occurrence when we analyze a period of data.

3.4.3 Golden Data Set

To conduct our experiments, we have prepared a golden data set of data to compare different feature-oriented link generation functions and the graph-based ranking algorithms. First, 53,276 threads (in 693,308 posts) were selected randomly (10%) in our Dataset. Second, formatting

removal and a compression-based Chinese word segmentation algorithm [30] were applied to extract a set of potential words. As this algorithm was applied before the experiment, training and testing data will have the same base error due to the potential incorrect segmentation. It will not affect the subsequent result. Then, we asked several voters to assign an importance value to each word that is most representative to the thread's topic. The weighting scale is 1 (less important) to 10 (most important) as shown in Figure 3.14.

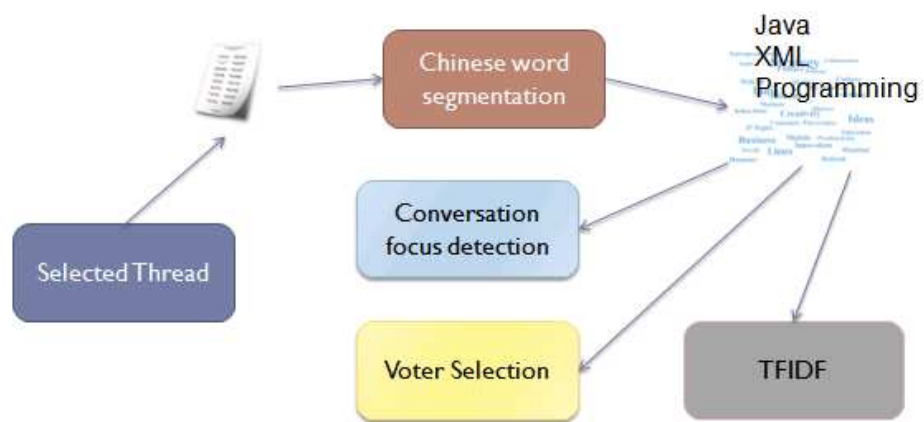


Figure 3.14 Experimental Flow

3.4.4 Mean Square Error

To evaluate different feature-oriented link generation functions and the graph-based ranking algorithms, we used the Mean Square Error (MSE) measure. First, we normalized all the weighting value (0 to 10) of our manually labeled words to sum of 1. A set of <word, score> list was obtained. We assume that this list is used to indicate the real ratio to represent a thread.

For instance, the voter assigns the following weights in a thread. Then, the rightmost column is the ratio:

Table 3.1 Example of Weighting Value assigned by Voter

Word	Weight (1 to 10)	Ratio (Sum to 1)
Java	8	8/15 = 0.53
Programming	5	5/15 = 0.33
XML	2	2/15 = 0.13

Second, all the importance ranking of words extracted by algorithms perform the same way. Then, the difference of them for each word is squared and all of the squared differences are equalized to become mean square error value. Thus, the lower the error value, the better the algorithm's performance. In the example above, we use PageRank with TFIDF to weight the word:

Table 3.2 Example of Squared Difference of Weighting Value

Word	Weight calculated by PageRank with TFIDF	Calculated Ratio (Sum to 1)	Squared Difference with respect to Voter
Java	0.6448	0.6448/1.2169 = 0.8218	$(0.53-0.8218)^2 = 0.08515$
Programming	0.4323	0.4323/1.2169 = 0.3552	$(0.33-0.3552)^2 = 0.000635$
XML	0.1398	0.1398/1.2169 = 0.1149	$(0.13-0.1149)^2 = 0.000228$

Thus, the mean square error is: $\frac{1}{3} \times 0.08515 + 0.000635 + 0.000228 = 0.028671$

3.4.5 Evaluation of Different Feature-oriented Link Generation Methods

First, we conduct an experiment to evaluate the performance of different feature-oriented link generation methods. We use PageRank graph-based ranking algorithm as the control method, and check different link generation functions.

In fact, different link generation functions can have different weighting based on its importance. To evaluate different link generation functions, we first set all the generation functions' weighting to 1, i.e., no weighting. Then, we test our data with the golden data set by measuring the mean square error and counting which link generation function is outperformed in each thread.

Table 3.3 Results of the Mean Square Error of different Feature-oriented Link Generation

Feature-oriented link generation function	Average MSE	MSE S.D.
TFIDF	0.776821	0.31321
4) all generation functions	0.27288	0.268547
1) Lexical Similarity Only	0.29293	0.246477
1) Quote Only	0.261614	0.237645
1) Direct Link Only	0.262492	0.237568
1) Authority Only	0.273232	0.239012
2) Direct Link and Quote	0.259325	0.235892
2) Direct Link and Authority	0.268299	0.247840
2) Direct Link and Lexical Similarity	0.27892	0.247232
2) Quote and Authority	0.269827	0.237849
2) Quote and Lexical Similarity	0.281232	0.246721
2) Authority and Lexical Similarity	0.289254	0.234493
3) Direct Link, Quote and Lexical Similarity	0.280012	0.248901
3) Direct Link, Quote and Authority	0.258732	0.235490
3) Direct Link, Lexical Similarity and Authority	0.285321	0.236484
3) Quote, Lexical Similarity and Authority	0.279463	0.248549

In table above, different combination tests were conducted in 53276 Threads. The average MSE and its standard deviation are reported. TFIDF is just for baseline comparison. In this experiment, the works show that Quote is the most significant feature-oriented link generation functions in one on one comparison. As talked in 3.4.2, quotes appear frequently (2.28 posts in

average). The relationships between posts / replies are very strong. If one post was quoted by many others replies, the content of that post is much representative of the thread. That supports why we need a new thread model with different feature-oriented link generations. Also, we found that Lexical Similarity makes some noise to the result. When Lexical Similarity was used, the MSE will be bigger. The reason of that is some similarity is because of stop word removal in Chinese is not well performed. Some stop words appear again and again between each post and incorrectly boost the vertex importance. And the results imply that different link generation functions contribute to the generation of the thread model and prove that word-thread matrix formation is more effective than original word-document co-occurrence matrix for threaded text.

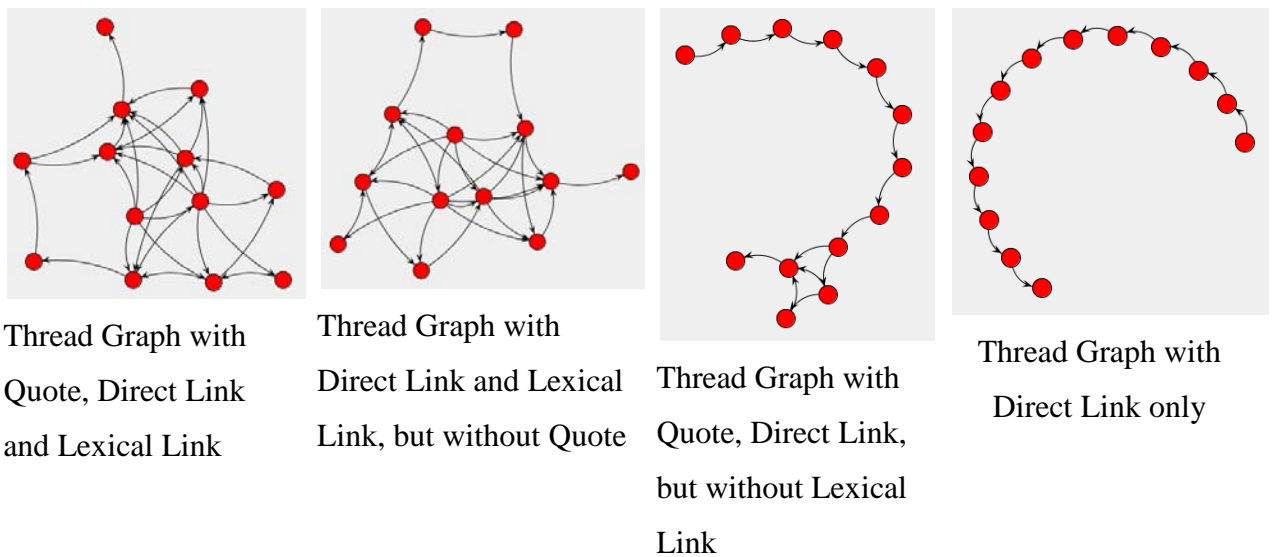


Figure 3.15 Sample of different Link Generated in Weighted Directed Graph

In Figure 3.15, four directed acyclic graphs are presented. In each graph, vertex is the post / reply of a thread. Link is the relationship between post and replies. With different link generation functions, vertex with more in-link arrows means more importance in its thread. By applying different link generation functions, a more concrete model for threaded text representation can be obtained.

3.4.6 Evaluation of Different Graph-based Ranking Algorithms

Next, we conduct another experiment to evaluate the performance of different graph-based ranking algorithms. In order to show the performance of different graph-based ranking algorithms, we also simulate the solely TFIDF method as a baseline comparison. The following table is the result of the Mean Square Error (MSE) of different graph-based ranking algorithms and TFIDF approach:

Table 3.4 Results of the Mean Square Error of different Graph-based Ranking Algorithm

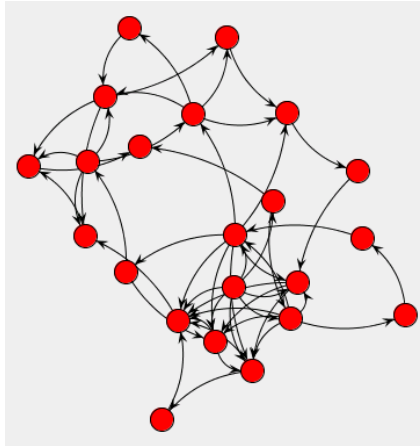
Graph-based Ranking Algorithms	Average MSE	MSE S.D.
TFIDF	0.804549721	0.341108675
Degree	0.474503026	0.25219906
Betweenness	0.270371452	0.265691907
HITS	0.495848664	0.289949148
PageRank	0.26917844	0.264098845

In the Table 3.4, the PageRank and Betweenness algorithm is the two better methods for extracting bag of words compared with all the graph-based ranking algorithms. When compared with TFIDF, it is noted that all the graph-based ranking algorithms are outperformed. It means that our proposed conversation focus detection framework is very useful for extracting important bag of words for topical modeling.

Figure 3.16 shows three samples of extracted words and its generated directed acyclic graph. In these real samples, the vertex relationships are very complicated. Different ranking algorithms are used to measure the vertex weighting and finally provide a list of important words to represent the thread. By comparing our golden data set, each MSE were obtained and the minimum error one was used to extract represented words. In the first figure, a thread with 22

posts with the topic of “how many size of memory stick used in your digital camera?” was used. In this example, PageRank algorithm is the best performed one with the smallest MSE. Based on this algorithm, we can extract a list of potential keywords to represent this thread.

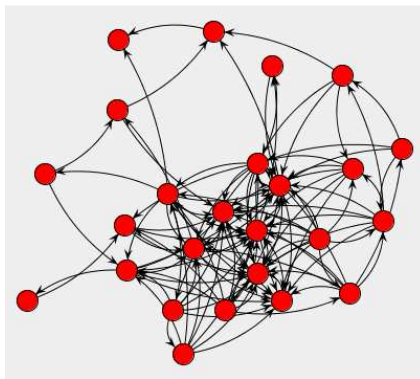
Based on this evaluation, it supports that a more concrete model to represent importance distributions between posts and replies is needed. In the result, it is clear that a threaded text possesses an implicit structure like tree or even a graph between posts and replies. If the proposed preprocessing framework was used to construct bag of words representation for threaded text topic modeling, a better result can be obtained.



Forum: Uwants.數碼相機 DC
Title: 大家用幾大記憶卡?
Publish Time: Sat Sep 02 11:34:00 CST 2006
Thread size: 22

MSE(tfidf): 0.5846950630308869
MSE(hits): 0.39486198365876396
MSE(degree): 0.3463538923384369
MSE(betweenness): 0.3250321546195029
MSE(pagerank): 0.3240806776817077

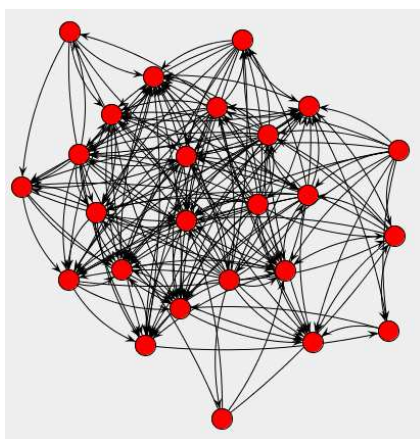
Extracted words by PageRank:
gb, 256, x1, mb, gb+60, 512 mb, 2gb, sd, cf



Forum: Uwants.娛樂圈動態
Title: **不如一起討論下台灣D歌手牙";}
Publish Time: Fri Sep 01 12:16:00 CST 2006
Thread size: 25

MSE(tfidf): 0.6479394856319005
MSE(hits): 0.20118469464563998
MSE(degree): 0.2302969990988608
MSE(betweenness): 0.2767283102032347
MSE(pagerank): 0.2767222936706863

Extracted words by HITS:
香港, 台灣, 孫燕姿, 國語, 唱功, 歌手



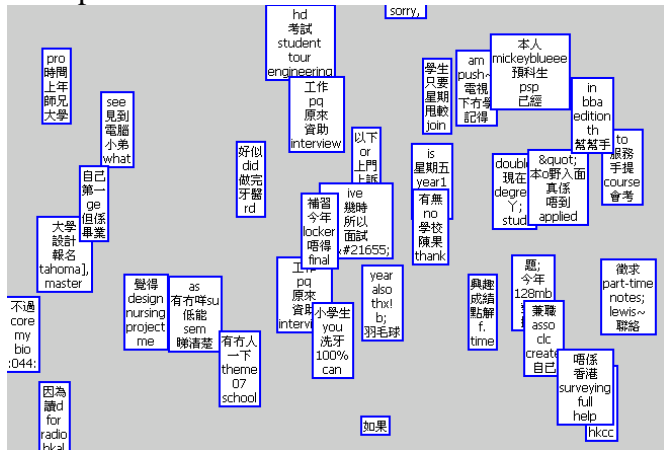
Forum: Uwants.時事及政治討論
Title: 大家覺唔覺得爭取子女居港權係無理要求??
Publish Time: Fri Sep 01 15:04:00 CST 2006
Thread size: 25

MSE(tfidf): 0.5684820054497931
MSE(hits): 0.18904808889572663
MSE(degree): 0.28890567929590455
MSE(betweenness): 0.18751862031624733
MSE(pagerank): 0.18751301527136843

Extracted words by PageRank:
大陸, 女人, 大陸人, 團聚, 香港, 子女, 結婚, 移民, 貢獻, 爭取

Figure 3.16 Sample of Extracted Words and Generated Graph

Example 2



Uwants.理工大學 PolyU
 Number of Posts: 3311
 Number of Threads: 530
 From 2006-09-01 to 2006-12-31

TOPIC_46	0.01721	TOPIC_47	0.02662
double	0.10133	and	0.32731
現在	0.10133	唔該	0.15714
degree	0.0912	polyu	0.12442
Y;	0.08108	check	0.03934
stud	0.05071	讀 nursing	0.03934
屋企	0.05071	trans	0.02625
工程	0.04059	optometry	0.0197
fitness	0.04059	門口	0.0197
fd~	0.03047	lecturer	0.0197
mr	0.02035	外觀	0.01316

TOPIC_46	Entity
理工大學	0.32188
香港	0.08231

TOPIC_47	Entity
理工大學	0.27634

Time Span of Topic 46: 2006-09-01 to 2006-09-04
 Time Span of Topic 47: 2006-09-10 to 2006-09-16

Figure 3.18 Example 2 of LDA and its variants by approximated using Gibbs Sampling

In the above Figure 3.18 and Figure 3.18, two examples of LDA modeling are shown. The left top image is the visualization of extracted topics by matlab. In each box insides the image, it represents a latent topic containing a set of words. At the right top side, two sample topics are shown. The latter value is the probability of the topic appears and the word distribution of that topic.

At the button of the examples, the corresponding topic-entity and topic-time modeling results are shown. By applied these two extended LDA models, entity and time information can also be extracted in the modeling.

3.5 Conclusion

In this chapter, we proposed a framework and algorithms to select bag of words by conversation focus detection. Conversation focus detection is a method based on the idea of well-developed graph-based ranking algorithms for extracting bag of words in threaded text. In threaded text, not all the contents (posts) are equally important. Threaded text usually involves two or more parties, discussing an interesting topic, and each party conveys certain information to the topic during the turn by turn interaction. Each turn does not contribute equal important information to the topic. In other words, the importance measure of each word is different from pure flattened text. With ranking algorithms, one may easily score different importance of each post and extract more representative bag of words in a discussion thread.

In addition, we briefly introduce a set of latent variables modeling for textual data analysis in topical problem. We conducted several experiments to evaluate different ranking algorithms and feature-oriented link generation methods. The performance of our methods, some sample from bag of words extraction and topical modeling are presented. Empirical results of the Hong Kong popular web forums show that our proposed methods are proved to be more meaningful and effective.

Chapter 4

Non-Topical Modeling in Threaded Discussion Analysis

4.1 Introduction

The User Generated Content (UGC) has become the fastest growing sector of the World Wide Web. Individuals can share opinions, experiences and expertise by simply clicking a button in the paradigm of the Web 2.0 platform. This media, called Social Media, has had an increasingly important role in today's marketing, journalism and opinion polling. With the growing importance of UGC, there are increasing and compelling needs to develop techniques for analyzing such tons of data, for example by grouping them into a meaningful manner.

Data mining from UGC presents challenges not typically found in text mining from documents. UGC, such as newsgroup posts, blogs and discussion forum threads, can be semi-structured, and can contain links and images represented by embedding HTML tags or some proprietary codes like BBCode and DiscuzzCode. The content of UGC can be very short and informal, containing relatively little content similar to a chat or an email conversation. In contrast, some blogs contain substantial contents like news articles or personal diaries [33]. Besides the content, UGC can be viewed as multi-modal data. Such kind of data has various type variables like document, word, author, title, publishing time, entity, sentiment, mood and even genre [34]. Its categorization can

be handled in terms of multiple modalities, not just grouping similar topics which share common “bag of words” in a set of documents. For instance, documents can be clustered as groups authors who share similar sentiment expressions in a kind of entity which relates to a particular consumer electronic product. These characteristics and new needs have posed big challenges and research questions for scholars to cope with.

Traditionally, clustering documents are based on their “bag of words” vector space representation which forms a document-word matrix of similarity. Then, hierarchical or k -means algorithm can be employed to group the documents automatically in a one-way fashion. In two-way clustering, scholars try to cluster documents based on the common words that appear in them and to cluster words based on the common documents that they appear in at the same time. Surprisingly, this approach can work well with sparse and high-dimensional data in “bag of words” representation of documents[35]. Recently, multi-modality clustering has become a popular topic in machine learning community. In [10], Bekkerman presents a multi-way distributional clustering (MDC) algorithm based on the pair-wise interaction between multiple type variables. The idea of this algorithm is that simultaneous clustering of different type of textual variables such that the one clustering in two variables can bootstrap clustering in the other two different variables. Empirical evidence shows that the overall clustering quality of documents can be improved when adding more additional types of data.

To cluster UGC data, similar to conventional document-word based clustering. We can construct various contingency tables (e.g. document-word, author-entity and sentiment-entity tables) of textual data types and then employ the MDC algorithm. However, by considering a contingency table which summarizes the co-occurrence statistics of two textual type variables in a document, it is not robust to represent the information entropy between two variables in UGC data. For

instance, a replying post contains a positive sentiment is correlated with a product name entity which only appears in the master thread of current post. Such kind of correlation cannot be counted in co-occurrence because two types of features are placed in two different documents independently. Because of this limitation, we would like to propose a novel similarity measurement, called Distributional Similarity Model (DSM), to solidify the graph model proposed by Bekkerman to cope with the unique characteristics of UGC data.

The remainder of this chapter is organized as follows. First, we will describe the idea of the Distributional Similarity Model for matrix construction in two textual type variables. Then, we will introduce the Multi-modality clustering algorithm and how our data can be fed into such algorithm. We will also present the experimental results to show the improvement of our model from baseline. Finally, we will sum up and discuss some future works.

4.2 Distributional Similarity Model

In this section, we will first introduce the nature of UGC data. Then, a detail description of the model of distributional similarity and its distance measure using positional and link information between features will be presented.

4.2.1 The Nature of UGC Data

UGC data is semi-structured text. It is temporal in nature and contains tree-like linking between documents. In some message boards, such as the discussion forums and the Usenet newsgroups, you can even see the existence of “Quote” inside the content. In some discussion forums, users can leave their messages using a rich content editor to embed a predefined list of proprietary style codes for formatting, and insert hyperlinks or mood icons to decorate their emotions in text.



Figure 4.1 General Structure of a Discussion Thread

Meta Information

- Document ID, Parent Document ID, Master Document ID
- Forum/Usenet/Blog Hosting ID, Board ID

Document Information

- Title, Content, Author, Published Time, Tag(s)

Figure 4.2 Schema of a Document Post

In Figure 4.1, a thread is a particular topic in a board. Board is an interest group for bulletin, e.g. mobile phone user board. A thread may contain one or even thousands of reply posts for a topic. The first post, called master post, starts the conversation. In Usenet newsgroup, a post can be linked with a particular reply post, called parent post. Thus, Figure 4.2 shows that the schema of a post contains several fields to represent the linking and meta.

Inside a post, titles, authors, published time and contents are included. In its content, it can be pure text or semi-structured text embedded with HTML tags or style codes. Some posts may contain a nested quote for replying a particular message appeared in the previous post. In general, a text contains several paragraphs and each paragraph is composed of a list of sentences as shown in Figure 4.3.

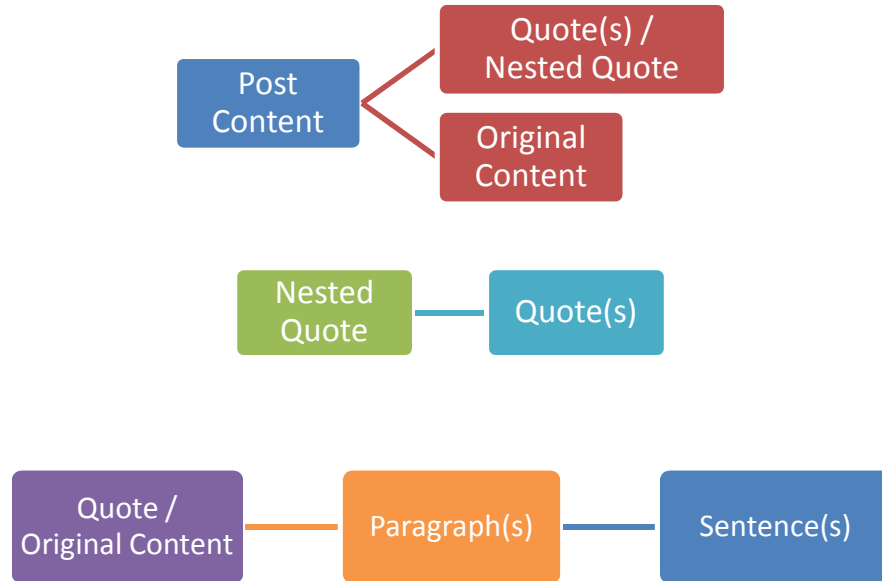


Figure 4.3 Internal Structure of Document Post Content

4.2.2 Distributional Similarity

Distributional similarity is a method to measure two features' distributional similarity based on their contexts. Originally, the method is used to determine the semantic similarity between two words appearing in a similar context [36]. Based on the original idea, we propose a method for measuring distributional similarity in UGC data.

As mentioned before, documents can be viewed from different angles, e.g. a set of topics or non-topical angles such as sentiment, mood and genre. Those angles or modalities are projected based on particular textual type features. Some features may be a keyword, a set of words, a phrase, a pattern, a concept or even a complex structure of clues embedded inside a document. A feature possesses not only its type and value but also positional information to describe where it is located inside a document. In addition, the master, parent and reply positions of a document provide link information to describe the inter-document relationship within a thread. Based on

this positional and link information of a feature, we can then calculate the distance between the two features inside the same document or under the same thread (i.e. the same context).

The relationship between two textual type variables is typically formulated as a matrix for clustering algorithms. The count between two features in the matrix is calculated as follows:

$$Count = \sum_{i=0}^m b^{x_{f(i)}} - \frac{(b^{x_{f(i)}} - b^{x_{f(i-1)}}) \times dist_i}{total_dist_i} \quad [4.1]$$

or decomposed as:

$$Count = \sum_{i=0}^m d_i$$

$$d_i = W_j - df_i$$

$$df_i = (W_j - W_{j-1}) \times \frac{dist_i}{total_dist_i}$$

$$W_j = b^{x_{j_i}}$$

$$j_i = f(i)$$

where m is the number of co-occurrence of two different textual type features under the same thread. The feature distance d_i is calculated by standard weighting W_j minus the decay factor df_i [37]. The standard weighting is an exponential function of x_{j_i} mapping from $f(i)$ and its base is variable b , which is a predefined constant. Therefore, the level of distance weighting between posts, paragraphs and sentences can be defined by the value of x_{j_i} (level of distance) and b (predefined constant). In contrast, it is necessary to introduce the decay factor that can be used to

adjust the weighting by the relative distance $\frac{dist_i}{total_dist_i}$ between two features, e.g. the relative distance between feature A in sentence 1 and feature B in sentence 7 is 0.6, in total of 10 sentences. Then, the decay factor is calculated by the relative distance multiplying with the unit weighting between the current weighting W_j and the upper weighting W_{j-1} . For example, when the current weighting is sentence weight, the upper weighting is paragraph weight. In other words, if feature A is in the first sentence and feature B is in the last sentence of the same paragraph, the final weighting nearly equals its upper weighting (i.e. paragraph weighting). In contrast, the weighting is maximized if two features are located in their neighboring sentences. Therefore, to measure the distance between two features, the level of distance such as the position of post, paragraph, sentence or character are used to model the distributional similarity in an exponential function with a decay factor for adjustment.

By adopting this model, the contingency table is extended by introducing the distributional weighting with the original occurrence counting between two textual type features.

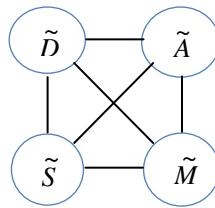


Figure 4.4 A Sample Pair-wise Interaction Graph, Variable D =Document, A =Author,

S =Sentiment, M =Mood

4.3 Multi-Modality Clustering Algorithm

In [10], Bekkerman et al. introduced the pair-wise interaction graph to model the problem of multi-modality clustering. In the following, the idea of this model is highlighted.

Let $G = (V, E)$ be a pair-wise interaction graph over different textual type variables $\tilde{X}_i, i = 1, \dots, m$ where $\tilde{X}_i = V$, m is the number of different textual type variables and \tilde{X}_i is the partitions of variable i . For each $e_{ij} \in E$, it is given by a contingency table T_{ij} determined by our DSM calculation. To execute the clustering algorithm, the input is the graph G , the tables T_{ij} and a clustering schedule. Based on the given schedule, clusters \tilde{X} are determined by maximizing the mutual information $I(\tilde{X}, \tilde{Y})$ which indicates that the amount of information clusters \tilde{X} are provided by clusters \tilde{Y} , in every edge linking from \tilde{X} . Therefore, the objective function is:

$$\max_{\{\tilde{X}_i\}} \sum_{e_{ij} \in E} w_{ij} I(\tilde{X}_i; \tilde{X}_j) \quad [4.2]$$

where w_{ij} is the augment edges in E to weight the relationship between two textual types.

Finally, multiple textual type variables are simultaneously clustered based on the schedule by choosing maximum information gain between textual types in each step.

4.4 Experimental Results

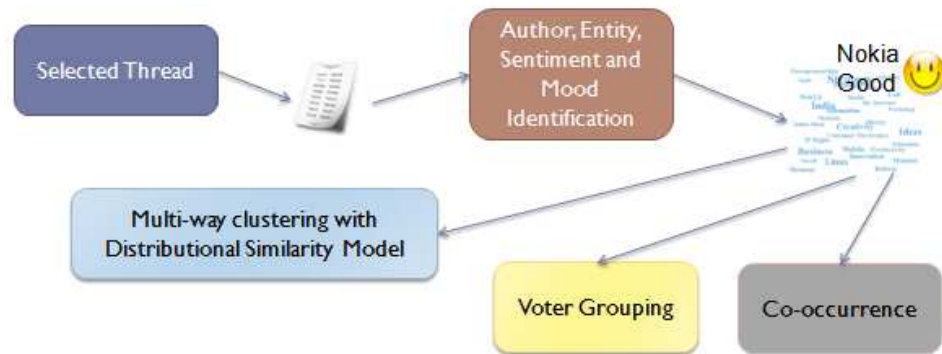


Figure 4.5 Flow of the DSM experiments

In our experiment in Figure 4.5, we focused on the effectiveness of using our DSM in constructing the contingency tables for multi-way distributed clustering algorithm. The

evaluation was conducted by using a set of labeled collections of documents selected from our UGC dataset which was collected by a tailor-made grasping engine from 24 newsgroups sets, 12 discussion forums and 7 blog hostings, and around 65 millions posts mainly coming from Hong Kong online social communities were obtained. The voters were asked to group two domains of documents: consumer electronics, mobile phone – 8000 posts from 2006 Apr 27 to 2006 Dec 06 (Dataset 1) and The Hong Kong Third Term Chief Executive Election - 3000 posts from 2006 Nov 1 to 2007 Mar 31 (Dataset 2).

Several variables for the MDC algorithm were chosen in our experiment: 1) Document ID, 2) Author, 3) Sentiment, 4) Mood and 5) Entity. Document ID and Author are directly captured from the document file attributes. The features of sentiment are obtained by our SEN algorithm which is a similar work of Theresa Wilson et. al. in [38]. Mood is expressed by smile codes, e.g. 😡, 📺 and 😊, which are normally dedicated from particular forums. So we summarized them to form a list of unified smile codes by grouping similar codes together. The person name, location and organization name are the features classified as entity. The Table 4.1 below shows the statistics of the features set extracted from selected datasets.

Table 4.1 Number of Features Summary

Dataset	Labeled Class	Author	Sentiment	Mood	Entity
1	1073	5547	879	37	118
2	578	1873	327	21	59

We compared the performance of our DSM scheme in three tests. The first one used the fully distributional similarity measurements including inter and intra post weighting of feature set pair.

The second one only includes intra post weighting. And the last one is to count the original co-occurrence as the baseline comparison to evaluate our method.

Table 4.2 The Weighting of Distance Level between features, where $b = 2$ in $w_j = b^{x_{ji}}$

	Character	Sentence	Paragraph	Type	Parent	Post
x_{ji}	3	2	1	0	-1	-2
w_j	8	4	2	1	0.5	0.25

In Table 4.2, different weighting of distance level between features are shown. The Character, Sentence, Paragraph and Type level distance measures are intra post weighting functions which are used in first two tests. Type is the distance level used for comparing features located in different titles, quotes and contents in the same document. In order to evaluate from baseline, the Type weighting is set to one which means no weighting. Furthermore, Parent and Post distance level are inter-post weightings to measure the inter correlation between posts within the same thread. In this experiment setting, Type and Parent distance level do not have decay factor. The Post level is measured by post index under the same thread.

Table 4.3 The Precision on Two Domains of Data

Dataset	Test 1 (Fully DSM)	Test 2 (Intra DSM)	Test 3 (Co-occurrence)
1	71.2%	68.9%	66.1%
2	75.8%	70.1%	67.2%

The results based on comparison between testing and training, Table 4.3 show that our DSM scheme improves 5-7% from baseline. And there is an enhancement of 3-5% when inter-post distance level measurements are considered.

In this experiment, we can see that the overall precision of multi-modalities clustering is increased. In contrast to using simply co-occurrence of features solely, including the distributional features, it requires a little additional cost, while the performance can be improved.

4.5 Conclusion

This chapter has presented a Distributional Similarity Model (DSM) for Multi-Modality Clustering in social media. Based on the unique inter and intra structure of User Generated Content (UGC), the clustering quality can be improved by considering both positional and link information when applying feature extraction with a little additional cost. This chapter was published in [P1].

Chapter 5

Conclusion

5.1 Contribution

In this thesis, we have proposed a preprocessing framework for topical and non-topical threaded discussion analysis. The proposed models and algorithms have been simulated and tested on the most popular Hong Kong web forums. As demonstrated by the experimental results, they are effective and yet efficient.

To the best of our knowledge, the work of this thesis represents a first attempt (to the best of my knowledge and belief) to handle semi-structured threaded discussion text by graphical probabilistic modeling. The contributions of this thesis are:

- 1) To provide a framework to analyze the web forum with the recent emerging statistical learning techniques, for instance, latent variable models or Markov random fields. In topical modeling, we have proposed a conversation focus detection method to select an appropriate bag of words to represent threaded text. This scheme is able to measure the importance of particular word by considering the relationships between posts and replies. In non-topical modeling, we have proposed a Distributional Similarity Model (DSM) to solidify the similarity measure

between different textual type variables. This model allows us to measure not only co-occurrence but also distributional similarity in different types of distance level commonly existed in threaded text.

2) To provide an empirical evidence for developing an online buzz surveillance and analysis systems. With the growing importance of web forum data, there are increasing and compelling needs to develop sophisticated system to help analyzing such tons of data. With the recent applicability of graphical probabilistic modeling, an in-depth study is required.

5.2 Future works

Among the many topics to be explored in future research, some important ones can be listed as follows:

In topical modeling, the following two have been identified

1) Speech act analysis between posts and replies for feature-oriented link generation

Pragmatic knowledge is quite important in conversation focus analysis. In [24], Feng argues that we can adopt the theory of Speech Acts and define a set of speech acts (SAs) that relate every pair of messages in the corpus. Based on their analysis, three categories of speech acts can be grouped. Messages may involve a request (REQ), provide information (INF), or fall into the category of interpersonal (INTP) relationship. Categories can be further divided into several single speech acts.

A speech act may represent a positive, negative or neutral response to a previous message depending on its attitude and recommendation. Then, the strength of each speech act is calculated as:

$$W^S(SA) = \text{sign}(dir) \sum_{person_k} \frac{\text{count}(SA_{person_k})}{\text{count}(SA)} W^P(person_k) \quad [5.1]$$

where the sign function of direction is defined with

$$\text{sign}(dir) = \begin{cases} -1 & \text{if } dir \text{ is } NEGATIVE \\ 1 & \text{otherwise} \end{cases} \quad [5.2]$$

However, speech act analysis is computationally intensive and it is probably not easy to design in Chinese environment.

2) Sentiment analysis in latent variable modeling

Today, users usually express their opinions in the web forums. Sentiment classification is a technique to classify reviews into positive and negative based on the overall sentiment expressed by authors[38-41]. Besides the research in this thesis, we have developed a preliminary sentiment analysis algorithm and applied to two set of data namely, Mobile Phone and the Hong Kong Third Term Chief Executive Election. Some interesting results are shown in Figure 5.1 and Figure 5.2:

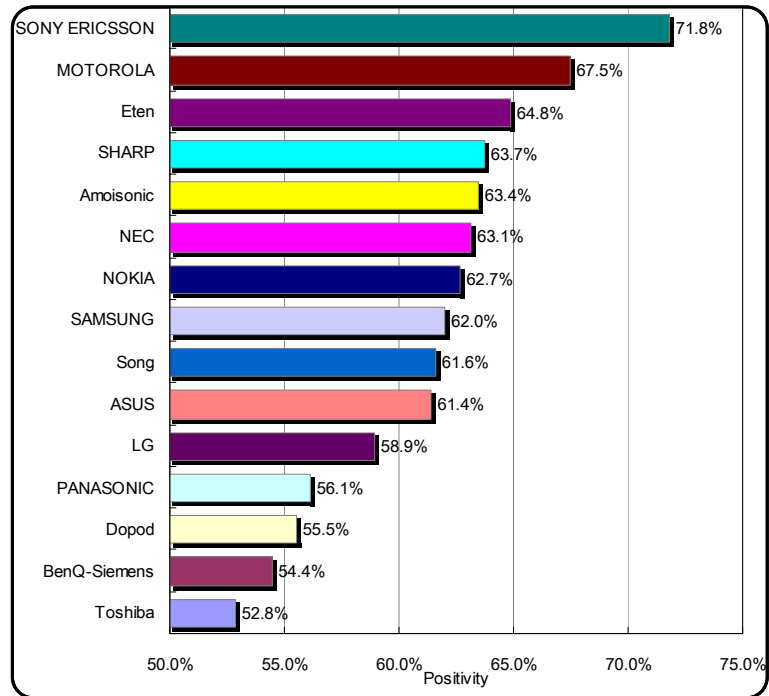


Figure 5.1 Positivity of Mobile Phone Brands

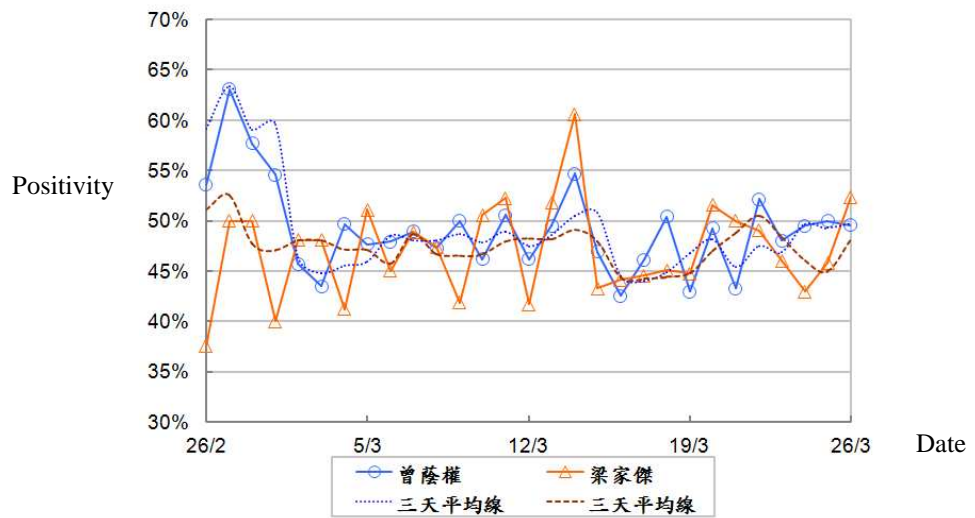


Figure 5.2 Positivity of Tsang and Leung in Chief Executive Election

In our future works, we would like to develop a sentiment modeling with topic, author, entity and time. For instance, in the case of sentiment-author-entity modeling, we want to know what sentiment orientation a group of author expresses in a particular mobile phone brand entity.

In non-topical modeling, the following three research directions can be considered:

3) Automatic learning of weight value in distributional similarity model

We can further improve the weighting function by applying automatic learning from overall distance measurements of features in dataset. Our idea is that the current approach has a limitation of manually predefined weighting value for positional and link distributional distance. We would like to develop a more scientific approach to assign these values.

One possible approach is to use statistical methods, like TFIDF, to estimate the relative importance of positional and link distance with the whole dataset. For instance, distributional distance is measured by the total number of paragraphs in a thread with inversed average number of paragraphs in the whole dataset.

4) Temporal models for multi-way distributional clustering

Current multi-way distributional clustering based graph model is a standard Markov random fields. Similar to LDA based topic time modeling, temporal information can be introduced into our MDC based graph model for pattern mining.

5) Markov-logic network (MLN) for textual analysis

Besides Bayesian and Markov networks introduced in this thesis, Markov-logic network is a new graphical model proposed in [42]. It is a first-order knowledgebase with a real number, attached

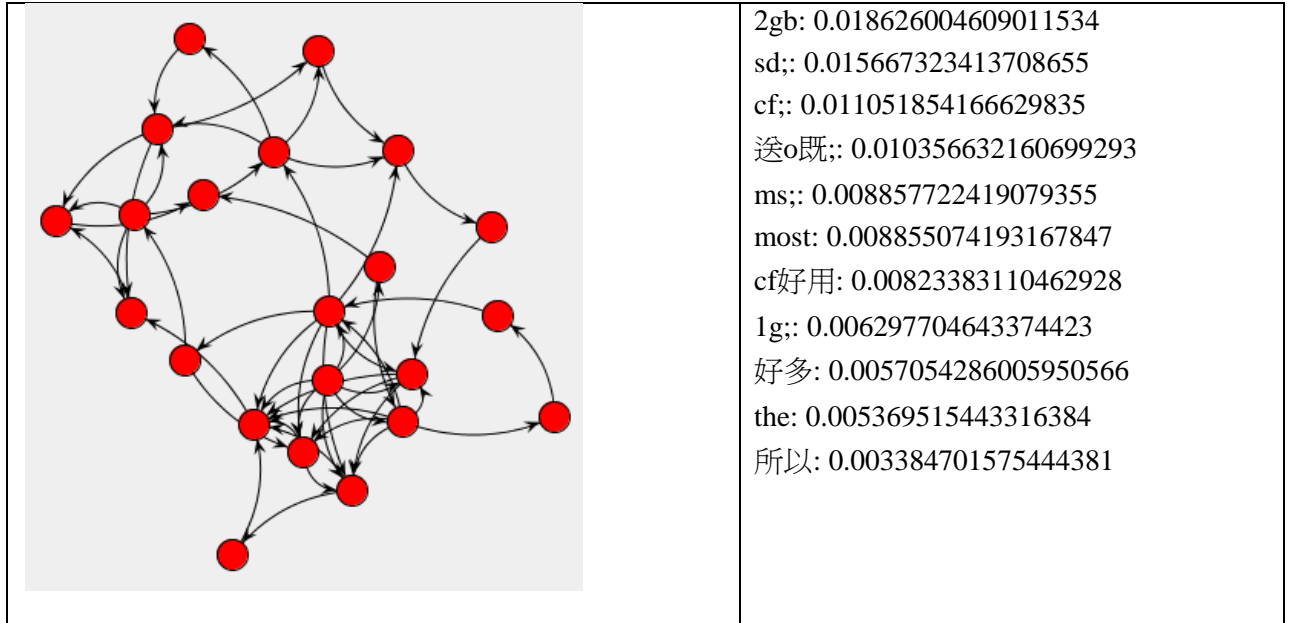
to each formula, and implements a probabilistic logic. With this model, we may be able to extract more high-level knowledge from the textual data.

Appendices

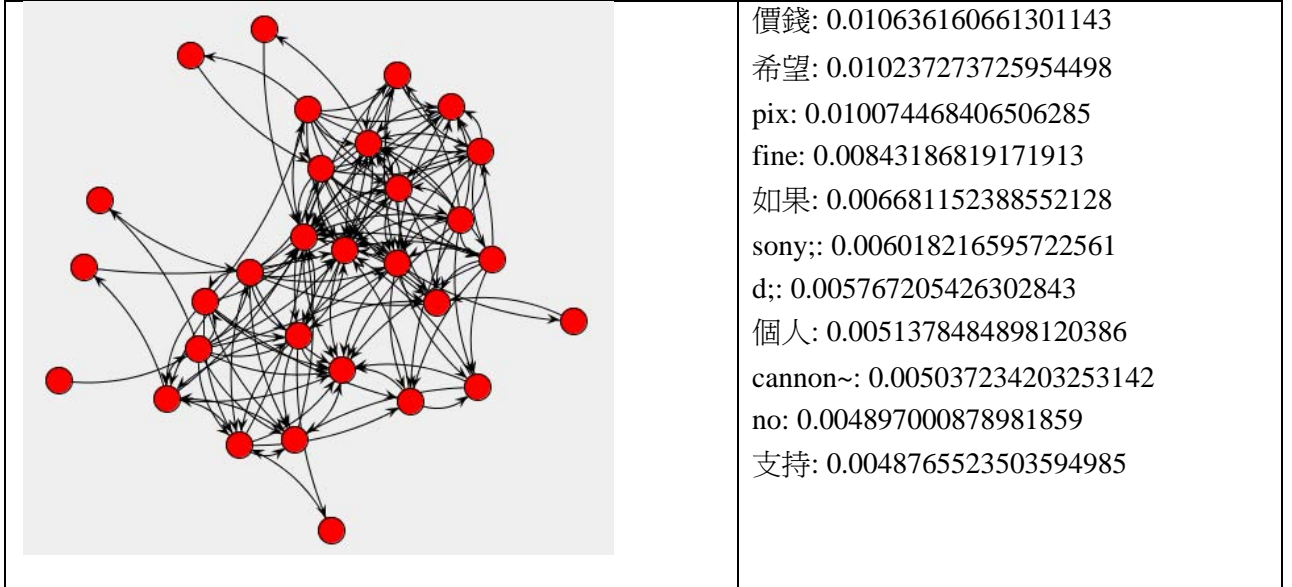
A. Some results of bag of words extracted by graph-based ranking algorithms

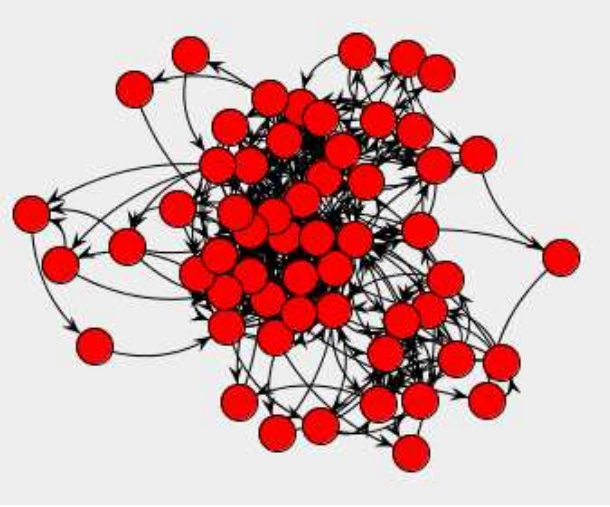
Below shows several samples of extracted words and its generated directed acyclic graph. Different ranking algorithms are used to measure the vertex weighting and finally provide a list of highest words to represent the thread. By comparing our golden data set, each MSE were obtained and the minimum error one was used to extract represented words.

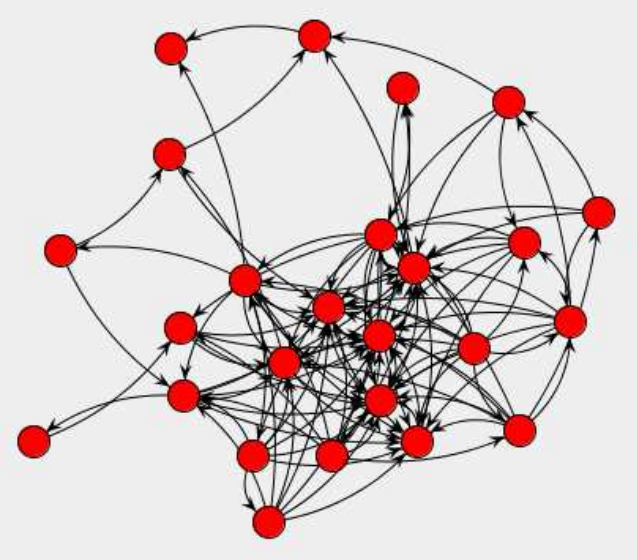
Forum: Uwants.數碼相機 DC Title: 大家用幾大記憶卡? Publish Time: Sat Sep 02 11:34:00 CST 2006 Thread size: 22 MSE(tfidf): 0.5846950630308869 MSE(hits): 0.39486198365876396 MSE(degree): 0.3463538923384369 MSE(betweenness): 0.3240806776817077 MSE(pagerank): 0.3240806776817077	Result of PageRank algorithm gb: 0.04420741666651934 256: 0.03530153648822384 gb,: 0.032479347050939314 下,: 0.029584795149286962 x1: 0.028002740034506216 mb: 0.027993386675692008 gb+60: 0.02210472703224985 512: 0.021834664883672035 mb,: 0.020713264321398587
--	--

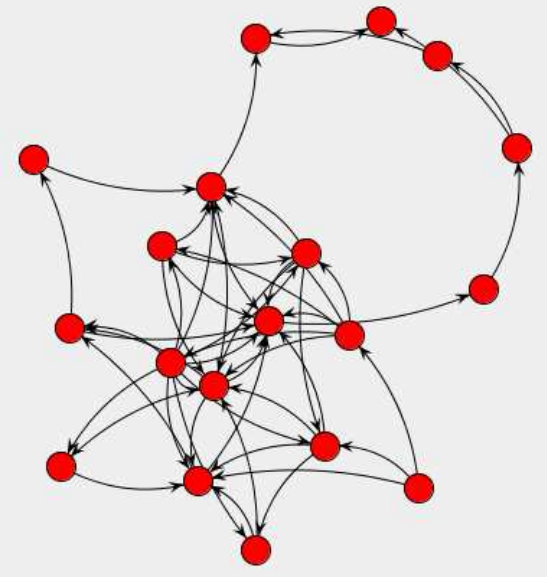


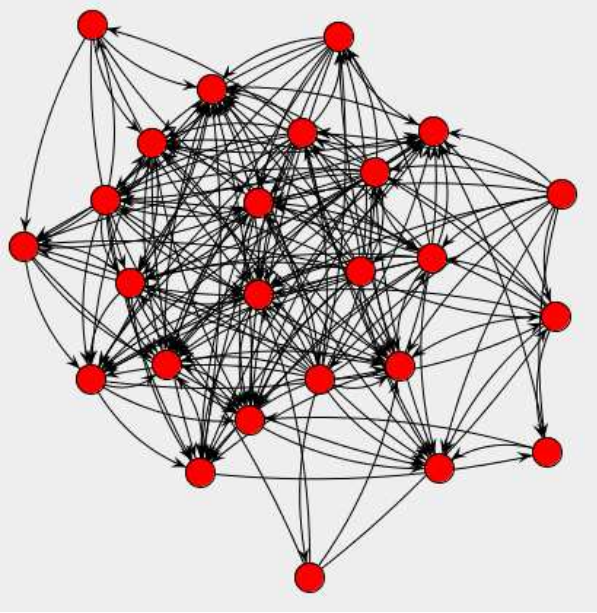
<p>Forum: Uwants.數碼相機 DC Title: 買機的決擇 ... 15/16 ... 幫幫手投下 la!! Publish Time: Sat Sep 02 20:08:00 CST 2006 Thread size: 30</p> <p>MSE(tfidf): 0.5632081370860988 MSE(hits): 0.15449032056563014 MSE(degree): 0.1904077566391642 MSE(betweenness): 0.18480040594368774 MSE(pagerank): 0.18479067642541605</p>	<p>Result of HITS algorithm</p> <p>canon: 0.05059120915031473 cannon: 0.04533510782927828 sony: 0.03781458112401838 isux: 0.031911362884244066 800: 0.025278606286215893 ixus: 0.023634113202511436 is:: 0.02028466089991071 相機: 0.01890729056200919 影像: 0.012413341356425718</p>
--	---



<p>Forum: Uwants.數碼相機 DC Title: 終極 DC 機王(只限機仔)選舉 Publish Time: Fri Sep 01 17:26:00 CST 2006 Thread size: 57</p> <p>MSE(tfidf): 0.5513923479709684 MSE(hits): 0.1958763044480109 MSE(degree): 0.24550170537231425 MSE(betweenness): 0.09511450115809486 MSE(pagerank): 0.09511382090650225</p>	<p>Result of Betweenness algorithm</p> <p>fujifilm: 0.0527670382450511 f30: 0.03925179323516158 800: 0.016612695809346045 canon: 0.015102450735769126 pix: 0.015066747663070541 ixus: 0.013592205662192231 fine: 0.01208196058861531 to: 0.011726008498900247 digital: 0.010571715515038407 br: 0.00960333875226785 /&: 0.00941671728941909 is: 0.0067791257157837 is: 0.006625659393547859 f30: 0.005688756433678084 win: 0.005650030373651453 lt: 0.00533518819570436 分別: 0.004753120611326741 xd: 0.004669985710378191 ricoh: 0.004609571584857352 &lt: 0.004572888536189531</p>
	

<p>Forum: Uwants.娛樂圈動態 Title: **不如一起討論下台灣D歌手牙&quot;;} Publish Time: Fri Sep 01 12:16:00 CST 2006 Thread size: 25</p> <p>MSE(tfidf): 0.6479394856319005 MSE(hits): 0.20118469464563998 MSE(degree): 0.2302969990988608 MSE(betweenness): 0.2767283102032347 MSE(pagerank): 0.2767222936706863</p>	<p>Result of HITS algorithm</p> <p>香港: 0.06108038470875428 台灣: 0.0443531366070928 好多: 0.02545749118550147 五月: 0.02417857104507957 孫燕姿: 0.022131666850147443 lee: 0.017406928212629205 跳舞: 0.017368555497811623 好過: 0.014623616243496918 s.: 0.013489287471043597 唔好: 0.0129161715793447 不過: 0.01145431801224458 國語: 0.009907196257079873 王心: 0.00974907749566461 歌手: 0.009614130667285524 david: 0.008994264134786793 當然: 0.008684277748905811 唱功: 0.008355163094698742 h.: 0.008324555512982421 唔得: 0.008199554496558901 喇!: 0.006812625250965764</p>
	

<p>Forum: Uwants.時事及政治討論 Title: 你覺得香港應否出錢買大陸運動員?? Publish Time: Fri Sep 01 16:17:00 CST 2006 Thread size: 19</p> <p>MSE(tfidf): 0.6496882879929213 MSE(hits): 0.1472363386066301 MSE(degree): 0.1596485210735216 MSE(betweenness): 0.1883725243441088 MSE(pagerank): 0.18835135377392115</p>	<p>Result of HITS algorithm</p> <p>運動員: 0.026342382941954644 支持: 0.01976340578791075 唔係: 0.019756787206465985 香港: 0.01789070604632433 &#36130;: 0.013171191470977322 不少: 0.00908137814328936 香港人: 0.008945353023162165 增光: 0.007654805883107952 金牌: 0.007577058385069946 因為: 0.0073730795362518696 好似: 0.007156282418529687 大气粗: 0.006585595735488661 o岩: 0.0064895301043064145 能力: 0.005922597737934392 中國: 0.005367211813897266 有咩支格: 0.005316966013787715 不如: 0.005051372256713291 :046:: 0.004936850617110694 世界: 0.004904125115859402 出錢: 0.004290429663851755</p>
	

<p>Forum: Uwants.時事及政治討論 Title: 大家覺唔覺得爭取子女居港權係無理要求?? Publish Time: Fri Sep 01 15:04:00 CST 2006 Thread size: 25</p> <p>MSE(tfidf): 0.5684820054497931 MSE(hits): 0.18904808889572663 MSE(degree): 0.28890567929590455 MSE(betweenness): 0.18751862031624733 MSE(pagerank): 0.18751301527136843</p>	<p>Result of PageRank algorithm</p> <p>應該: 0.02696451261977939 大陸: 0.02426156265148186 女人: 0.01557706793565232 大陸人: 0.013881535764868256 團聚: 0.01387807830940674 香港: 0.013704046698431692 人士: 0.011556219694191166 可以: 0.010703630581536103 子女: 0.009963694514959032 申請: 0.009855622529065907 結婚: 0.00867595985304266 無理: 0.008522534071887102 好似: 0.007846299866290318 家人: 0.007662346144245195 居民: 0.00759490248466033 the: 0.007155549482272416 移民: 0.006943755885931822 貢獻: 0.006940767882434128 唔係: 0.006870644651194775 爭取: 0.006818027257509679</p>
	

B. Some results of LDA approximated results by using Gibbs sampling

Below is two set of LDA results. In each result, 50 topics are extracted. Each topic contains a set of words. The latter value is the probability of the topic appears and the word distribution of that topic.

1. Uwants.數碼相機 DC,

Number of Posts: 9214, Number of Threads: 1353, From 2006-09-01 to 2006-12-31

TOPIC_1	0.02063	TOPIC_2	0.0256	TOPIC_3	0.02027	TOPIC_4	0.01847
f30	0.31043	可以	0.77871	影相	0.13492	f31	0.13247
鏡頭	0.20929	適合	0.01689	真係	0.11717	350	0.12858
岩岩	0.04537	80	0.01408	價錢	0.10297	題	0.07794
之後	0.04189	少少	0.00846	you	0.06038	thank	0.06626
850is	0.03142	相信	0.00846	mm	0.03553	cansorry00	0.04289
thx.	0.02794	pm	0.00846	price	0.03553	:098:	0.0351
:006:	0.02445	s9600	0.00565	普通	0.03198	中文	0.0312
變焦	0.01747	可能	0.00565	w70	0.02843	震.	0.0273
日常	0.01747	銀色	0.00565	arial,	0.02133	星期六	0.02341
影響	0.01399	無限	0.00565	can	0.01778	a700	0.02341
TOPIC_5	0.02221	TOPIC_6	0.01855	TOPIC_7	0.01797	TOPIC_8	0.0184
唔知	0.31098	防震	0.13584	各位	0.25632	謝謝	0.10953
其實	0.20409	800	0.09704	牌子	0.10415	&	0.06652
腳架	0.10044	:028:	0.09704	lens	0.07612	no	0.06652
光學	0.03242	>	0.0466	師兄	0.04008	of	0.05087
機友	0.01947	wide	0.03884	face	0.03608	見到	0.04696
ccd	0.01947	其實	0.03108	建議	0.03608	特別	0.03523
so	0.01623	is	0.0272	d200	0.03207	:023:	0.03523
意思	0.01623	唔係	0.02332	係咪	0.02407	20010	0.03132
software	0.01299	有咩牌子=	0.02332	電量	0.02407	仲有	0.03132
指點	0.01299	s5	0.02332	一&#	0.02407	28mm	0.03132
TOPIC_9	0.0207	TOPIC_10	0.01905	TOPIC_11	0.01855	TOPIC_12	0.02272
fujifilm	0.22247	比較	0.13225	dc	0.27552	數碼	0.33259
kit	0.06954	all	0.05292	旅行	0.06212	維修	0.1964
第一	0.06954	kodak	0.05292	日期	0.03884	但係	0.11405
二手	0.03479	pls	0.04914	唔係	0.03496	鑑於	0.06654
leica	0.03479	搵機	0.04537	一般	0.03496	會友	0.02854
分享	0.03479	s6500	0.04159	好~	0.03496	jpeg	0.0222

300	0.03131	how	0.04159	手機	0.03108	大大	0.00953
not	0.02784	質素	0.03781	唔駛	0.03108	鐘頭	0.00953
thx	0.02784	2500	0.03026	he	0.0272	brought	0.00953
aa	0.02089	星際	0.02648	> <	0.0272	學"	0.00953
TOPIC_13	0.02135	TOPIC_14	0.01811	TOPIC_15	0.01905	TOPIC_16	0.03581
問題	0.21235	水貨	0.29003	應該	0.16247	canon	0.91417
最好	0.11799	百老匯	0.11127	g7	0.14358	thkz	0.00605
and	0.0944	想要	0.0636	對焦	0.0567	牌子	0.00404
意見	0.08429	保養	0.05565	try	0.05292	hong	0.00404
谢	0.04721	sigma	0.03579	60	0.02648	大 mon	0.00404
已經	0.04047	日本	0.02785	賣	0.02648	200	0.00203
個人	0.03373	don't	0.02387	豐澤	0.0227	一樣	0.00203
a640	0.01688	各界	0.02387	有无	0.0227	mph	0.00203
用品	0.01688	f31fd	0.0199	買唔買	0.0227	印像	0.00203
请	0.01351	a16	0.0199	永成	0.0227	plan 多	0.00203
TOPIC_17	0.01833	TOPIC_18	0.02049	TOPIC_19	0.01797	TOPIC_20	0.01826
olympus	0.16099	功能	0.28452	本人	0.14019	有無	0.23651
900	0.12959	旺角	0.14052	push	0.10015	casio	0.14586
a710	0.09818	選擇	0.05623	t30	0.08413	需要	0.05127
唔係	0.09033	分別	0.0492	buy	0.08013	delete	0.03157
家用	0.02359	as	0.0492	希望	0.04008	konica	0.03157
諗左好	0.02359	power	0.03867	t10	0.03608	全新	0.03157
搵部	0.01967	國美	0.03516	anybod	0.03207	有	0.02763
ge	0.01967	修理	0.02111	around	0.02807	picture	0.02369
million	0.01574	感測器	0.02111	thanks	0.02407	noise	0.01975
有咩機	0.01574	arial]	0.0176	效果	0.02006	:023:	0.01975
TOPIC_21	0.01962	TOPIC_22	0.02034	TOPIC_23	0.01761	TOPIC_24	0.01991
快門	0.1247	sony	0.56241	ricoh	0.14306	fx07	0.11567
手動	0.08437	card	0.05663	請教	0.09402	幾多	0.08676
it	0.06237	究竟	0.02833	所以	0.07768	<	0.07592
f,	0.04037	like	0.01772	am	0.05725	a640	0.07231
先決	0.03304	既	0.01772	iso 好	0.0409	for	0.06147
fx01	0.03304	t-50	0.01418	出現	0.03273	s3	0.04701
!	0.02937	歷史	0.01065	mju	0.03273	多謝	0.03617
some	0.02937	價格	0.01065	auto	0.02864	可能	0.02895
買	0.0257	仲有 d 咩	0.01065	边	0.02456	影片	0.02895
good	0.0257	sc-t30	0.01065	r5	0.02456	緊要	0.02533
TOPIC_25	0.01919	TOPIC_26	0.01768	TOPIC_27	0.0166	TOPIC_28	0.02085
最近	0.09377	有冇人	0.1791	in	0.09106	因為	0.30031

參考	0.09377	"	0.12213	30	0.06939	之前	0.11393
睇下	0.07877	dcfever	0.04481	證據	0.04338	where	0.08632
大約	0.07502	興趣	0.04074	聖誕	0.03472	help	0.0449
支援	0.05627	check	0.03667	記得	0.03472	just	0.03455
mode	0.04503	官方	0.0326	d?	0.03472	casioz57	0.02074
感光	0.04503	wanchai	0.0326	hello	0.03038	hkd	0.01729
方法	0.04128	入門	0.02446	using	0.02605	would	0.01384
买	0.03378	pebtax	0.02446	機~	0.02605	dc 仔,	0.01384
正常	0.02253	不如	0.02446	但是	0.02605	細部	0.01384
TOPIC_29	0.02099	TOPIC_30	0.01948	TOPIC_31	0.02013	TOPIC_32	0.02128
如果	0.52792	朋友	0.12562	400	0.39327	panasonic	0.46329
sd	0.09259	白色	0.05914	點解	0.10371	唔好	0.09471
今天	0.03088	當然	0.05175	d40	0.05723	made	0.05414
速度	0.02746	有得	0.04806	bod	0.04293	1	0.04399
樓主	0.02746	原裝	0.04067	ok	0.02506	昨日	0.0237
fx-50	0.0206	拍片	0.04067	如果	0.02149	f31	0.0237
s9	0.0206	memory	0.03697	about	0.02149	環境	0.02032
借問	0.01375	thx	0.03328	好好	0.01791	n2	0.02032
買咗	0.01375	有冇	0.02959	max	0.01791	:032:	0.02032
battery	0.01375	以下	0.0222	可信	0.01434	dc.	0.01694
TOPIC_33	0.01862	TOPIC_34	0.01934	TOPIC_35	0.01876	TOPIC_36	0.02121
pentax	0.22421	不過	0.23819	左右	0.12277	is	0.35969
自己	0.12758	唔該	0.16005	有咩	0.10743	今日	0.17986
咁多	0.06574	題~	0.09679	can	0.04606	fuji	0.14254
lcd	0.05415	差額	0.03725	push!	0.04606	一下	0.05093
攝影	0.04255	通常	0.03353	主要	0.04223	what	0.03057
冇人	0.03096	介紹	0.02981	dc	0.04223	唔錯	0.01021
ifc	0.03096	相機	0.02236	以前	0.02689	啦	0.01021
15	0.02323	放 wor	0.01492	夜晚	0.02689	think	0.01021
品牌	0.01936	上網	0.01492	2000	0.02689	喺	0.01021
d70s	0.0155	有冇花 lo!	0.01492	push	0.02689	fd31-how	0.01021
TOPIC_37	0.01883	TOPIC_38	0.02063	TOPIC_39	0.02502	TOPIC_40	0.01984
nikon	0.32098	the	0.26509	請問	0.67584	覺得	0.2938
d80	0.11084	to	0.14303	mon	0.06905	唔會	0.15961
電腦	0.08409	digital	0.09769	高手	0.04029	現在	0.11247
10	0.04207	or	0.0663	sorry	0.02016	full	0.03268
下~	0.03825	緊,	0.03142	mmm.	0.01153	with	0.03268
kit	0.02678	if	0.02096	爱	0.00866	電池	0.02542
其他	0.02296	mk	0.02096	老婆	0.00866	原裝袋	0.01817
鏡	0.01914	wp-content	0.02096	相對	0.00866	小妹	0.01817
保證	0.01532	photo	0.01747	咩先	0.00866	1cm	0.01817
5%	0.01532	you	0.01747	有冇得救 ga 部	0.00866	d 用	0.01454

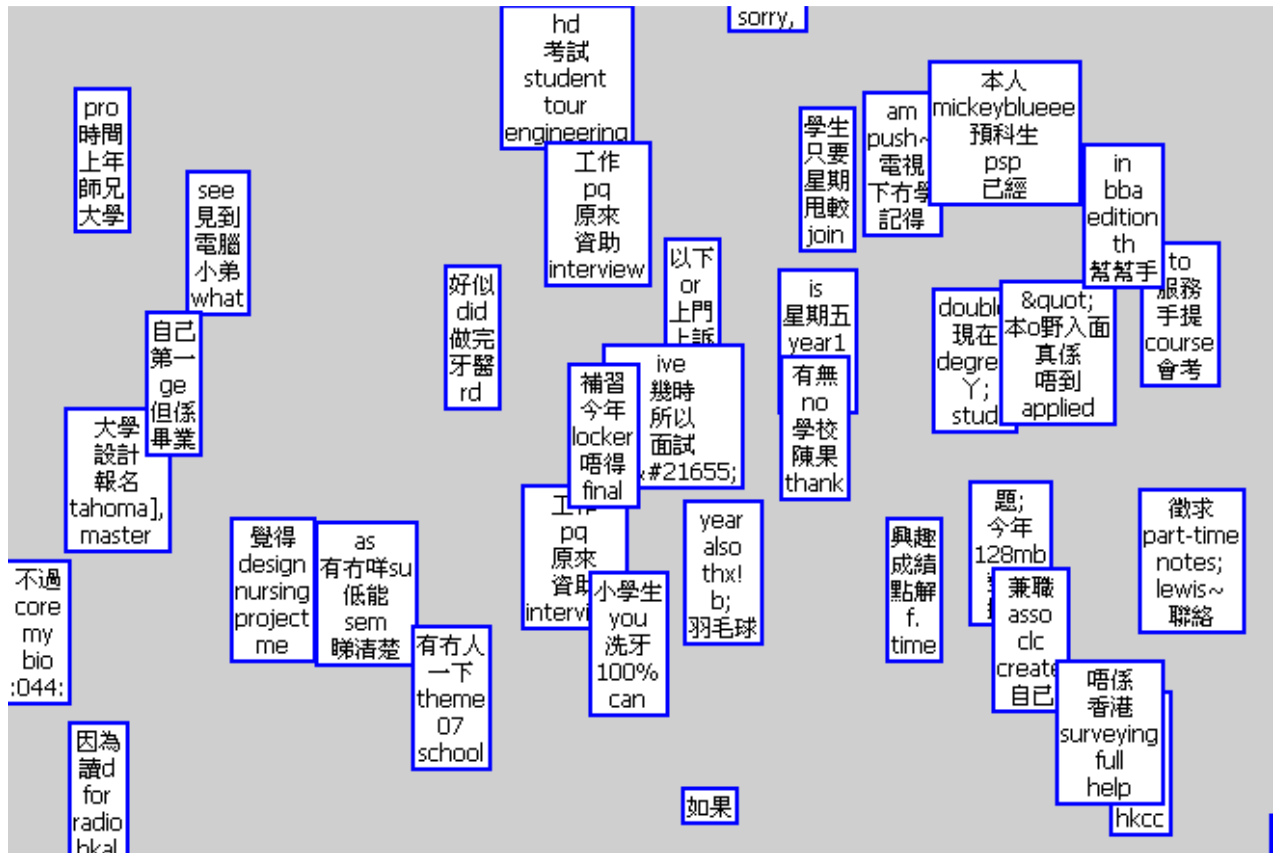
TOPIC_41	0.01718	TOPIC_42	0.01905	TOPIC_43	0.01732	TOPIC_44	0.02114
模式	0.1299	大家	0.44577	850	0.32821	ixus	0.25878
像素	0.12571	cannon	0.06803	xx	0.02912	唔係	0.09536
iso	0.0922	fx50	0.06048	600	0.02912	說明書	0.07834
拍攝	0.0545	a 系	0.02648	景深	0.02497	閃光燈	0.05451
:014:	0.0545	t-10	0.02648	dc,	0.02497	黑色	0.04429
好	0.03774	實用	0.0227	褲袋	0.02497	鴨記	0.02387
center]	0.03774	放電	0.01892	中文版	0.02497	有冇-3000	0.02387
帮	0.03355	是否	0.01892	誠信	0.02081	香港	0.02046
下邊	0.02517	完全	0.01515	d,	0.02081	揾	0.02046
dv	0.02517	45	0.01137	o 地	0.02081	屏幕	0.02046
TOPIC_45	0.01905	TOPIC_46	0.02481	TOPIC_47	0.01819	TOPIC_48	0.01991
好似	0.26068	相機	0.6179	小弟	0.28097	夜景	0.15181
好多	0.12092	短片	0.03774	要求	0.07522	光圈	0.14819
t50	0.04537	閃燈	0.03194	富士	0.07126	都係	0.1229
want	0.04537	唔到	0.02324	以上	0.05939	&#	0.0687
have	0.04537	相較	0.01743	相片	0.03961	邊間	0.03617
	0.02648	耐,	0.01743	我等	0.02378	2 千幾~	0.03617
跟住	0.0227	方面	0.01453	唔同	0.02378	問問	0.02533
黃金	0.01892	waste	0.01163	試機	0.02378	知道	0.02533
this	0.01515	佳能	0.00873	fz50	0.01982	gp	0.02533
時間	0.01515	thank	0.00873	几时	0.01587	睇唔到~	0.0181
TOPIC_49	0.01667	TOPIC_50	0.01826				
which	0.12087	新手	0.13798				
lx2	0.10361	samsung	0.08675				
一定	0.07772	廣角	0.07492				
gb	0.06909	ok	0.05916				
注意	0.03888	什麼	0.05127				
唔好意思	0.02594	ok 咖.	0.03945				
thx!	0.02594	a80	0.03157				
有冇高人	0.02594	考慮	0.03157				
430	0.02162	d 咩好	0.03157				
聲音	0.02162	睇中	0.02369				

60	0.06962	電腦	0.07572	我們	0.07482	洗牙	0.05681
p o l y	0.0418	小弟	0.07572	bu 讀的	0.04492	100%	0.05681
幫忙	0.0418	what	0.05681	緊 computer	0.04492	can	0.05681
quota	0.0418	go	0.05681	左嚟~	0.02997	pe,	0.05681
出讓	0.0418	題~	0.05681	hihi,	0.01502	咩係 light	0.04736
training	0.03484	arial]急	0.05681	7号	0.01502	product	0.03791
credit	0.02789	push.	0.04736	es;	0.01502	地點	0.02845
ielts	0.02789	shaw	0.04736	^^同學	0.01502	medical	0.02845
TOPIC_9	0.01721	TOPIC_10	0.01843	TOPIC_11	0.01634	TOPIC_12	0.02
year	0.12157	to	0.40656	工作	0.12805	poly	0.61861
also	0.10133	服務	0.03791	pq	0.10673	book	0.04364
thx!	0.07096	手提	0.03791	原來	0.0854	lab	0.02622
b;	0.05071	course	0.02845	資助	0.06408	your	0.01751
羽毛球	0.04059	會考	0.02845	interview	0.05342	ymca~	0.01751
rail	0.04059	公司	0.02845	core 果	0.04276	排係	0.01751
・	0.04059	how	0.02845	其他	0.03209	勁便 ok~	0.01751
籃球	0.03047	ppc	0.02845	=;	0.03209	正門	0.01751
va7 仔	0.03047	護士	0.019	明年	0.03209	義工	0.0088
civil	0.03047	bc;	0.019	bba 讀	0.02143	咁我	0.0088
TOPIC_13	0.01913	TOPIC_14	0.02226	TOPIC_15	0.02296	TOPIC_16	0.01773
有冇人	0.12761	唔係	0.31309	如果	0.55399	題;	0.18676
一下	0.05474	香港	0.07833	辛苦	0.08354	今年	0.09834
theme	0.05474	surveying	0.07051	邊間	0.05319	128mb	0.05904
7	0.05474	full	0.05486	gym	0.03802	對面	0.04922
school	0.03652	help	0.04703	例如	0.03043	擺明	0.03939
good	0.03652	好似	0.03138	介紹	0.03043	沒有	0.03939
雙面	0.03652	***	0.03138	think	0.01525	嚟 mini	0.03939
推推~	0.03652	land	0.02355	withdrawal	0.00766	nursing~	0.02957
ee?	0.02742	take	0.02355	like	0.00766	上午	0.02957
occupancy	0.02742	網頁	0.02355	借問	0.00766	where	0.01975
TOPIC_17	0.01721	TOPIC_18	0.01825	TOPIC_19	0.02766	TOPIC_20	0.01947
as	0.11145	徵求	0.11461	可以	0.6991	係咪	0.4563
有冇咩 su	0.10133	part-time	0.0669	please	0.05044	之後	0.14321
低能	0.10133	notes;	0.03827	唔駛	0.01896	唔點名;	0.03587
sem	0.07096	lewis~	0.03827	正在	0.01896	應該	0.02693
睇清楚	0.04059	聯絡	0.03827	ma	0.01266	咩書	0.02693
d;	0.04059	個個	0.02872	ga	0.01266	幾多	0.01798
電子	0.04059	then	0.02872	測量學	0.01266	地下	0.01798
if	0.03047	飛機	0.02872	丫~	0.00636	notes	0.01798
自己	0.03047	we	0.02872	咋;	0.00636	今天	0.01798
終於	0.03047	dun	0.02872	鋪;	0.00636	football;	0.01798
TOPIC_21	0.0186	TOPIC_22	0.01913	TOPIC_23	0.02139	TOPIC_24	0.02209

thx	0.09374	"	0.29156	camp	0.12224	補習	0.26035
poly 係有 pe!	0.09374	本 o 野入面	0.09117	同學	0.1141	今年	0.1105
any	0.07501	真係	0.06385	申請	0.08152	locker	0.0474
仲裁	0.05628	唔到	0.06385	thx.	0.04895	唔得	0.03951
讀完	0.05628	applied	0.03652	:098:	0.0408	final	0.03951
晚上	0.05628	-"	0.02742	下年	0.0408	下~	0.03163
職業	0.03755	出路	0.02742	tutorial	0.03266	jockey	0.03163
打鼓	0.03755	what	0.01831	零五年	0.03266	急聘	0.02374
have	0.03755	cool	0.01831	緊要	0.03266	phy	0.02374
唔好;	0.03755	睇番	0.01831	左學;	0.03266	xddd;	0.02374
TOPIC_25	0.02278	TOPIC_26	0.02069	TOPIC_27	0.02034	TOPIC_28	0.01982
the	0.40531	自己	0.1937	好似	0.21415	本人	0.18464
of	0.14535	第一	0.12636	did	0.15421	mickeyblueee	0.05282
o 係	0.07653	ge	0.08427	做完	0.03434	預科生	0.05282
3882	0.03066	但係	0.05901	牙醫	0.03434	psp	0.04403
db;	0.02301	畢業	0.03376	rd	0.03434	已經	0.03524
ga~	0.02301	y504 係	0.03376	中醫	0.03434	自備	0.03524
make	0.02301	group	0.03376	問題	0.02577	me	0.03524
be	0.01537	ga.	0.02534	soc	0.02577	:028;;	0.03524
hkcc	0.01537	想像	0.02534	晒呀~	0.02577	:005;;	0.03524
pm	0.01537	女仔	0.02534	嘩~	0.01721	lec	0.03524
TOPIC_29	0.01686	TOPIC_30	0.02087	TOPIC_31	0.02191	TOPIC_32	0.01825
以下	0.14475	星期四	0.13365	今日	0.17497	ive	0.1337
or	0.08276	d 咩	0.08356	各位	0.15907	幾時	0.09552
上門	0.06209	一定	0.08356	英文	0.11138	所以	0.0669
上訴	0.05176	hall	0.05852	hey//	0.04778	面試	0.05735
冇人	0.04143	sorry,	0.05017	push,	0.03983	買咗	0.03827
有~	0.04143	be	0.03348	報得	0.03188	:014:	0.03827
wanna;	0.04143	第二	0.03348	合格	0.02393	geo	0.02872
出售	0.04143	let's	0.02513	廣東話	0.02393	notification	0.02872
part	0.0311	hall 友	0.02513	syb.	0.02393	auditing	0.02872
intro	0.0311	谢	0.02513	大哥	0.02393	同班	0.02872
TOPIC_33	0.02069	TOPIC_34	0.01756	TOPIC_35	0.01965	TOPIC_36	0.0193
兼職	0.09268	不過	0.29775	hd	0.10648	大學	0.17159
asso	0.07585	core	0.10924	考試	0.09762	設計	0.14451
clc	0.05059	my	0.10924	student	0.06215	報名	0.06327
create	0.05059	bio	0.03979	tour	0.04442	tahoma],	0.04522
自己	0.05059	:044:	0.02986	engineering	0.04442	master	0.04522
下~;	0.05059	just	0.02986	19"	0.04442	作品集	0.03619
jupas	0.04218	唔知	0.01994	什麼	0.03555	0	0.03619
gpa	0.04218	g020	0.01994	出讓	0.03555	理工	0.03619
wave	0.03376	gh 既	0.01994	man.	0.03555	深 水	0.02717

三.	0.03376	business	0.01994	only	0.02669	入&t;	0.02717
TOPIC_37	0.01895	TOPIC_38	0.01756	TOPIC_39	0.02087	TOPIC_40	0.01669
覺得	0.11959	有無	0.18861	唔會	0.10861	is	0.27153
design	0.1012	no	0.07947	band	0.06687	星期五	0.06274
nursing	0.07363	學校	0.07947	之前	0.06687	year1	0.0523
project	0.06444	陳果	0.05963	男仔	0.05852	yr	0.0523
me	0.05524	thank	0.03979	hkcc	0.05017	:046:	0.0523
時間表	0.05524	steve	0.03979	m.	0.05017	12 月頭;	0.04186
有去 orientation	0.05524	冇人入	0.02986	associate	0.04182	4311	0.04186
xxdd~	0.05524	haha!	0.02986	應該	0.03348	who	0.03142
hd 土木 ga~	0.03686	bod	0.02986	but	0.03348	平均	0.02098
application	0.02767	個個	0.01994	補領	0.02513	一萬	0.02098
TOPIC_41	0.02592	TOPIC_42	0.02	TOPIC_43	0.02191	TOPIC_44	0.0193
請問	0.63183	in	0.19174	功課	0.12728	al	0.13548
文康大樓;	0.02023	bba	0.09591	lo	0.07163	學生會	0.06327
有關	0.02023	edition	0.09591	msn:	0.06368	:131:	0.05425
鼎鼎大名	0.01351	th	0.06107	靚女	0.05573	大學生	0.05425
本部	0.01351	幫幫手	0.05236	負責	0.05573	下 year1	0.05425
感動	0.01351	china	0.04364	msn	0.04778	that	0.04522
philosophy	0.01351	highlight	0.04364	high	0.03983	nice	0.04522
港大	0.01351	咁多	0.03493	survey	0.03188	:029:	0.03619
hp	0.01351	醫院	0.03493	主要	0.03188	晒啦~	0.03619
左啦	0.01351	支持	0.02622	hk	0.03188	誠徵	0.02717
TOPIC_45	0.01947	TOPIC_46	0.01721	TOPIC_47	0.02662	TOPIC_48	0.02069
因為	0.09849	double	0.10133	and	0.32731	financial	0.17687
讀 d	0.08954	現在	0.10133	唔該	0.15714	add	0.16003
for	0.0806	degree	0.0912	polyu	0.12442	&	0.09268
radio	0.06271	丫;	0.08108	check	0.03934	eng	0.08427
hkal	0.05376	stud	0.05071	讀 nursing	0.03934	/54130;	0.05059
放射	0.05376	屋企	0.05071	trans	0.02625	in,	0.05059
治療	0.03587	工程	0.04059	optometry	0.0197	中文	0.04218
老師	0.03587	fitness	0.04059	門口	0.0197	open	0.03376
課程	0.02693	fd~	0.03047	lecturer	0.0197	當然	0.03376
如上~	0.02693	mr	0.02035	外觀	0.01316	合格	0.01692
TOPIC_49	0.01843	TOPIC_50	0.01808				
興趣	0.13243	it	0.15425				
成績	0.10407	開始	0.08681				
點解	0.07572	very	0.06754				
f.	0.07572	ic	0.06754				
time	0.04736	know	0.06754				
ar;	0.02845	wanna	0.03864				

通常	0.02845	wed	0.03864
有冇興趣	0.02845	mtr	0.03864
confirmed	0.019	利息	0.03864
subject	0.019	for	0.01937



Bibliography

1. Stumme Gerd, Hotho Andreas, and Berendt Bettina, *Semantic Web Mining: State of the art and future directions*. Web Semantics: Science, Services and Agents on the World Wide Web, 2006. 4(2): p. 124-143.
2. G.Salton and M.McGill, *Introduction of Modern Information Retrieval*. 1983: McGraw-Hill.
3. Thomas Hofmann, *Probabilistic latent semantic indexing*, in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 1999, ACM Press: Berkeley, California, United States.
4. Bishop Christopher M., *Pattern Recognition and Machine Learning*. 2006: Springer.
5. Attias H., *A variational Bayesian Framework for Graphical Models*. Adv. Neur. Info. Proc. Sys. Vol. 12. 2000, Cambridge: MIT Press.
6. G Christian P Robert & Casella, *Monte Carlo statistical methods*. 2004, New York: Springer.
7. David M. Blei, Andrew Y. Ng, and Michael I. Jordan, *Latent dirichlet allocation*. 2003, MIT Press. p. 993-1022.

8. David Newman, Chaitanya Chemudugunta, and Padhraic Smyth, *Statistical entity-topic models*, in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006, ACM Press: Philadelphia, PA, USA.
9. Xuerui Wang and Andrew Mccallum, *Topics over time: a non-Markov continuous-time model of topical trends*, in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2006, ACM Press: Philadelphia, PA, USA.
10. Ron Bekkerman, Ran El-Yaniv, and Andrew Mccallum, *Multi-way distributional clustering via pairwise interactions*, in *Proceedings of the 22nd international conference on Machine learning*. 2005, ACM Press: Bonn, Germany. p. 41-48.
11. *Text Mining*. [cited 2008 Mar 29]; Available from: http://en.wikipedia.org/wiki/Text_mining.
12. Lance Parsons, Ehtesham Haque, and Huan Liu, *Subspace clustering for high dimensional data: a review*. ACM SIGKDD Explorations, 2004. **6**(1): p. 90-105.
13. Jing L., et al., *A Text Clustering System based on k-means Type Subspace Clustering and Ontology*. International Journal of Information Technology, 2006. **to appear**.
14. Deerwester Scott C., et al., *Indexing by latent semantic analysis*. Journal of the American Society of Information Science, 1990. **41**(6): p. 391-407.
15. Sivia Devender, *Data Analysis: A Bayesian Tutorial*. 1996: Oxford: Clarendon Press. 7-8.
16. D'ambrosio Z. Li and B., *Efficient inference in Bayes networks as a combinatorial optimization problem*. International Journal of Approximate Reasoning, 1994. **11**(1): p. 55-81.

17. Bellman Richard, *Dynamic Programming*. 1957: Princeton University Press.
18. Xuerui Wang, Natasha Mohanty, and Andrew Mccallum, *Group and topic discovery from relations and text*, in *Proceedings of the 3rd international workshop on Link discovery*. 2005, ACM Press: Chicago, Illinois.
19. Michal Rosen-Zvi, et al., *The author-topic model for authors and documents*, in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*. 2004, AUAI Press: Banff, Canada.
20. Mark Steyvers, et al., *Probabilistic author-topic models for information discovery*, in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2004, ACM: Seattle, WA, USA.
21. Hutchby Ian and Wooffitt, Robin, *Conversation Analysis*. 1988: Polity Press.
22. Paul Resnick, et al., *Beyond threaded conversation*, in *CHI '05 extended abstracts on Human factors in computing systems*. 2005, ACM: Portland, OR, USA.
23. Marc Smith, Cadiz J. J., and Byron Burkhalter, *Conversation trees and threaded chats*, in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. 2000, ACM: Philadelphia, Pennsylvania, United States.
24. Donghui Feng, et al., *Learning to detect conversation focus of threaded discussions*, in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. 2006, Association for Computational Linguistics: New York, New York.
25. Sabidussi G, *The centrality index of a graph*. Vol. 31. 1966: Psychometrika. 581-603.
26. Page Lawrence Brin Sergey, Motwani Rajeev and Winograd Terry, *The PageRank citation ranking: Bringing order to the Web*. 1999.

27. Jon M. Kleinberg, *Authoritative sources in a hyperlinked environment*. 1999, ACM. p. 604-632.
28. Vitor R. Carvalho and William W. Cohen, *On the collective classification of email "speech acts"*, in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 2005, ACM: Salvador, Brazil.
29. Dan Gusfield, *Algorithms on Strings, Tress and Sequences*. Computer Science and Computational Biology. 1997, USA: Cambridge University Press.
30. Teahan W. J., et al., *A compression-based algorithm for Chinese word segmentation*. Computational Linguistics, 2000. **26**(3): p. 375-393.
31. Jianfeng Gao, et al., *Chinese Word Segmentation and Named Entity Recognition: A Pragmatic Approach*. Computational Linguistics, 2005. **31**(4): p. 531-574.
32. Brandes U., *A Faster Algorithm for Betweenness Centrality*. Journal of Mathematical Sociology, 2001. **25**(2): p. 163-177.
33. Michelle L. Gregory Deborah Payne, David Mccolgin, Nicolas Cramer and Douglas Love, *Visual Analysis of Weblog Content*, in *International Conference on Weblogs and Social Media*. 2007: Colorado, USA.
34. Eguchi R. Bekkerman and H. Raghavan and J. Allan and K., *Interactive Clustering of Text Collections According to a User-Specified Criterion*, in *Proceedings of IJCAI-07, the 20th International Joint Conference on Artificial Intelligence*. 2007: Hyderabad, India.
35. Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha, *Information-theoretic co-clustering*, in *Proceedings of the ninth ACM SIGKDD international*

- conference on Knowledge discovery and data mining*. 2003, ACM Press: Washington, D.C. p. 89-98.
36. Lillian Lee and Fernando Pereira, *Distributional similarity models: clustering vs. nearest neighbors*, in *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. 1999, Association for Computational Linguistics: College Park, Maryland. p. 33-40.
37. Huma Lodhi, et al., *Text classification using string kernels*. *J. Mach. Learn.*, 2002. **2**: p. 419-444.
38. Theresa Wilson, Janyce Wiebe, and Paul Hoffmann, *Recognizing contextual polarity in phrase-level sentiment analysis*, in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. 2005, Association for Computational Linguistics: Vancouver, British Columbia, Canada. p. 347-354.
39. Lun-Wei Ku, et al., *Major topic detection and its application to opinion summarization*, in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 2005, ACM Press: Salvador, Brazil.
40. Bing Liu, Minqing Hu, and Junsheng Cheng, *Opinion observer: analyzing and comparing opinions on the Web*, in *Proceedings of the 14th international conference on World Wide Web*. 2005, ACM Press: Chiba, Japan.
41. Turney Peter, *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*, Philadelphia, Pennsylvania: In Proceedings 40th Annual Meeting of the Association for Computational Linguistics (ACL'02). 417-424.
42. Matthew Richardson Pedro Domingos, *Markov Logic Networks*, in *Machine Learning*. 2005.