12-1-2013

A Pointillism Approach for Natural Language Processing of Social Media

Piyo Song

Piyou Song

Candidate

Computer Science

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Jedidiah Crandall, University of New Mexico , Chairperson

Jared Saia, University of New Mexico

George Luger, University of New Mexico

Dan Wallach, Rice University

A Pointillism Approach for Natural Language Processing of Social Media

by

Song, Piyou

M.S., Pennsylvania State University
B.S., Harbin Institute of Technology

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

December, 2013

Acknowledgments

I am heartily thankful to my advisor, Jed Crandall, whose encouragement, guidance, and support from the initial to the final level enabled me to develop an understanding of the subject. I owe my deepest gratitude to Professors Dan Wallach, Jared Saia, and George Luger, their feedback and comments encouraged me to go forward.

My sincere thanks also goes to Dr. Sittichai Jiampojamarn. Without your support, it wouldn't be possible for me to finish my defence during my internship at Google. I thank my co-workers in Google: Yijian, Deepak, Sid, Zhengfan, Manoja. Your friendship means the world to me. Also I thank my officemates Xu, Jong, Geoff, *etc.* for your support and consideration.

感谢臭宝宝，一直像小猪一样，拱啊拱。送给臭宝一个永远最佳小猪奖。

感谢老爸，老妈一直像老铁一样，鼓励和信任。送给老爸老妈一人一个永远最佳老铁奖。

A Pointillism Approach for Natural Language Processing of Social Media

by

Song, Piyou

ABSTRACT OF DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

December, 2013

A Pointillism Approach for Natural Language Processing of Social Media

by

Song, Piyou

M.S., Pennsylvania State University
B.S., Harbin Institute of Technology

Ph.D, Computer Science, University of New Mexico, 2013

Abstract

Natural language processing tasks typically start with the basic unit of words, and then from words and their meanings a big picture is constructed about what the meanings of documents or other larger constructs are in terms of the topics discussed. Social media is very challenging for natural language processing because it challenges the notion of a word. Social media users regularly use words that are not in even the most comprehensive lexicons. These new words can be unknown named entities that have suddenly risen in prominence because of a current event, or they might be neologisms newly created to emphasize meaning or evade keyword filtering. Chinese social media is particularly challenging.

The Chinese language poses challenges for natural language processing based on the unit of a word even for formal uses of the Chinese language, social media only makes word segmentation in Chinese even more difficult. Thus, even knowing

what the boundaries of words are in a social media corpus is a difficult proposition. For these reasons, in this document I propose the Pointillism approach to natural language processing. In the Pointillism approach, language is viewed as a time series, or sequence of points that represent the grams' usage over time. Time is an important aspect of the Pointillism approach. Detailed timing information, such as timestamps of when posts were posted, contains correlations based on human patterns and current events. This timing information provides the necessary context to build words and phrases out of trigrams and then group those words and phrases into topical clusters.

Rather than words that have individual meanings, the basic unit of the Pointillism approach is trigrams of characters. These grams take on meaning in aggregate when they appear together in a way that is correlated over time. I anticipate that the Pointillism approach can perform well in a variety of natural language processing tasks for many different languages, but in this document my focus is on trend analysis for Chinese microblogging. Microblog posts have a timestamp of when posts were posted, that is accurate to the minute or second (though, in this dissertation, I bin posts by the hour).

To show that trigrams supplemented with frequency information do collect scattered information into meaningful pieces, I first use the Pointillism approach to extract phrases. I conducted experiments on 4-character idioms, a set of 500 phrases that are longer than 3 characters taken from the Chinese-language version of Wiktionary, and also on Weibo's hot keywords. My results show that when words and topics do have a meme-like trend, they can be reconstructed from only trigrams. For example, for 4-character idioms that appear at least 99 times in one day in the data, the unconstrained precision (that is, precision that allows for deviation from a lexicon when the result is just as correct as the lexicon version of the word or phrase) is 0.93. For longer words and phrases collected from Wiktionary, including neologisms,

the unconstrained precision is 0.87. I consider these results to be very promising, because they suggest that it is feasible for a machine to reconstruct complex idioms, phrases, and neologisms with good precision without any notion of words.

Next, I examine the potential of the Pointillism approach for extracting topical trends from microblog posts that are related to environmental issues. Independent Component Analysis (ICA) is utilized to find the trigrams which have the same independent signal source, *i.e.*, topics. Contrast this with probabilistic topic models, which leverage co-occurrence to classify the documents into the topics they have learned, so it is hard for it to extract topics in real-time. However, the Pointillism approach can extract trends in real-time, whether those trends have been discussed before or not. This is more challenging because in phrase extraction order information is used to narrow down the candidates, whereas for trend extraction only the frequency of the trigrams are considered. The proposed approach is compared against a state of the art topic extraction technique, Latent Dirichlet Allocation (LDA), on 9,147 labelled posts with timestamps. The experimental results show that the highest F1 score of the Pointillism approach with ICA is 4% better than that of LDA. Thus, using the Pointillism approach, the colorful and baroque uses of language that typify social media in challenging languages such as Chinese may in fact be accessible to machines.

The thesis that my dissertation tests is this: *For topic extraction for scenarios where no adequate lexicon is available, such as social media, the Pointillism approach uses timing information to out-perform traditional techniques that are based on co-occurrence.*

Contents

List of Figures

List of Tables

Glossary

Connector	The program connects that trigrams based on time correlation information.
Root trigram	The Root trigram is the first trigram of a words or phrase that we feed into Connector as an initial input.
Parent trigram	The Parent trigram is the last trigram that was appended to the current phrase.
Stem trigram	The Stem trigram is the one which has the lowest frequency in the current phrase that Connector is working on.
Unknown words	Unknown words are the already existing words that are not in the dictionary. These include named entities and neologisms.
Named entity	Named entities can be the names of people, companies, movies, and anything that is given a name.
Neologisms	Neologisms are words created recently, they are usually not words before they are created. However, there are also neologisms which give a different meaning to an already existing word. For example, 小姐 (Miss.) can also mean "prostitute" recently.

Phrases — Phrases are not fixed relative to words, since there may be different ways of saying phrases in different places and this may evolve over time. Phrases consist of multiple words.

PICA — Pointillism plus Independent Component Analysis.

LCP — Lexicon-constrained precision (LCP):

$$LCP = \frac{|\{Matching\ Wiktionary\ or\ Weibo\}|}{|\{Valid\ Root\ Trigrams\}|}$$

UP — Unconstrained precision (UP):

$$UP = \frac{|\{Matching\ Wiktionary\ or\ Weibo\} \bigcup \{Correct\ (human)\}|}{|\{Valid\ Root\ Trigrams\}|}$$

Chapter 1

INTRODUCTION

Social media poses many challenges for natural language processing. Many of these challenges center around the concept of a word. For example, social media users might invent new words called neologisms, which can express more meaning than the original word or evade content filtering. An example in English would be "Intarweb" to refer to the Internet, by using the neologism "Intarweb" net users are adding the additional meaning that the Internet and web are not something that is fully understood.

Chinese social media compounds these problems that are caused by the notion of a word, both because the written Chinese language does not delimit words with spaces and because Chinese net users use neologisms very heavily. How can we track trends, discover memes, and perform other basic natural language processing techniques for Chinese social media when it is not even clear that the problem of segmenting Chinese social media into words is a tractable problem?

This document is centered around a thought experiment: how much can machines understand about trends in social media for challenging languages such as Chinese without any lexicon or notion of words? I propose a Pointillism approach to natural

language processing and through experiments show that longer words and phrases can be put back together, and trends can be extracted based only on temporal correlations of trigrams.

The explosive growth rate of generating and sharing online content has attracted many researchers from different fields. However, to analyze this online data effectively presents many research problems.

To motivate the Pointillism approach to natural language processing in this document, I focus on the Chinese language. A unique feature of the Chinese writing system is that it is a linear sequence of non-spaced ideographic characters. The fact that there are no delimiters between words poses the well-known problem of segmentation. A natural language processing system with a lexicon could perform quite well, but the unknown words which are not registered in the lexicon become the bottleneck in terms of precision and recall [55]. For Chinese, Chooi and Ling [26] observed that if one can obtain good recall for unknown words, the overall segmentation is better. However, in Chinese social media unknown words are used regularly.

In this document, I illustrate an approach to recognize trends, not based on statistical bag-of-word models, as are traditional methods, but based on the frequency of trigrams. The goal of the Pointillism approach is to find trends (*i.e.*, a group of trigrams which have similar fluctuations in a certain period), not to generate a language model and then later use this model to classify the documents.

The proposed approach uses only a corpus with a time series, meaning that a system dictionary, grammar knowledge, or training are not required. The underlying concept of the proposed method is as follows. The problem of trend analysis is regarded as finding the trigrams which have the same trend. Observations show that the trigrams that have correlated trends over a long period of time have a high probability that they belong to the same topic or even belong to the same word.

The reason the proposed approach, which does not depend on a probabilistic model, is advantageous is because for probabilistic models, such as LDA, *a* priori knowledge is required and this forehand knowledge limits the capability of finding new topics. However, the Pointillism approach depends only on the frequency of trigrams to extract trends, no matter whether they are old or new trends.

In Chapter 4, the Pointillism approach is used to extract phrases. Our results from three kinds of experiments show that when words and topics do have a meme-like trend, they can be reconstructed from only trigrams. These promising results motivated us to go further to use the same method to extract trends from posts.

During phrase extraction, besides the frequency over time of the trigram, there is the order information of the trigrams. When connecting trigrams, Connector (my program for connecting trigrams, described in Chapter 4) only considers trigrams which have overlap with the current trigram, instead of treating all trigrams in the database as candidates. For example, when connecting "Putting lipstick on", only trigrams starting from "lipstick on ..." are scanned.

Phrase extraction can also become a story, in Section 4.5.3. However, in this document I try to answer a question: can a machine automatically summarize what is happening on social media only through trigrams with their respective frequency information?

To answer this question, in Chapter 5 the Pointillism approach is employed, using only trigrams with frequency information, to extract trends from microblog posts.

Contrast this with statistical language models, which leverage co-occurrence to classify the documents to the topic they have learned. The Pointillism approach extracts trends on the fly, no matter whether they have been discussed before or not. Independent Component Analysis (ICA) is utilized to find the trigrams which have the same independent signal source, *i.e.*, topics. In Chapter 5 proposed approach

is compared against a state of the art topic extraction technique, Latent Dirichlet Allocation (LDA), on 9,147 labeled posts with timestamps. The experimental results show that the highest F1 score of the Pointillism approach with ICA is 4% better than that of LDA.

With the Pointillism approach, thus the colorful and baroque uses of language that typify social media in challenging languages such as Chinese may in fact be accessible to machines.

Chapter 2

BACKGROUND

In this Chapter, I first summarize several salient features of the Chinese language, to help non-Chinese readers of this dissertation understand the relevant concepts and challenges. Then, I give a brief introduction of Sina weibo.

2.1 Chinese

English speakers expect words to be separated by whitespace or punctuation. In Chinese, however, words are simply concatenated together. Consequently, the problem of mechanically segmenting Chinese text into its constituent words is a difficult problem. During the process of segmentation of Chinese, two main problems are encountered: segmentation ambiguities and unknown word occurrences.

There are no indicators such as blank spaces to delimit the word boundaries in Chinese text. Therefore, in order to understand Chinese text, the first thing that we need to do is to divide the sentences into word segments. Certain characteristics of the Chinese language have made the segmentation problem more difficult than other languages. The same phenomenon of text not including spaces is not specific to only

Chinese but is also a feature of many other Asian languages such as Japanese and Korean. In Japanese, for example, characters are divided into three types, which are hiragana, katakana, and kanji. These different types of characters can help in telling where the word boundaries are. However, there are no such clues to indicate where the word boundaries are in Chinese text since there is only one single type of character, and only one single form for each word. In Chinese, there are only some punctuation marks that can help delimit the sentence or phrase boundaries.

Consider two major types of word segmentation ambiguity, both of which can happen for known words. First, 日本来 can be 日本/来 (Japan, come) or 日/本来 (sun, originally). In this case 本, can either belong to 日 to form the word 日本 (Japan), or belong to 来 to form 本来 (originally). Another example is 才能 (a word that means ability), it can also be segmented as two adverbs 才/能 (finally can).

These examples show ambiguity in how words can be segmented even for known words, the word segmentation problem in Chinese is exacerbated by the existence of unknown words such as named entities or neologisms. This is especially true in social media. Named entities can be the names of people, companies, movies, and anything that is given a name. Since social media is heavily centered around current events, it contains many new named entities that will not appear in even the most comprehensive lexicons. Neologisms are created to express a new meaning or the same meaning with different nuance, or sometimes to avoid censorship. Neologisms are also an integral part of social media.

For example, the Martian language is a kind of Internet slang predominantly used by young people for friendliness, cuteness, congeniality, self-expression, amusement, and enhancement of group solidarity between users [42]. Written dialect is used to show friendliness and form an exclusive group where only people who are part of a social circle can understand the posts. For example, in the following two posts from a microblogging site of China, weibo.com, there is at least one unknown word in each

sentence:

苦逼小青年-S：每天被这么多不认识的人@ 真的是有一种受宠若惊的感脚。但作为一个女屌丝我只能辜负大家对俺的厚爱了。对唔住啊！

他的妹汁知道：我在叉的床畔吖。非诚勿扰法国版必将是场男屌丝与女屌丝的巅峰对决因为正妹跟帅哥在法国这片romantic的土壤上早就开始生孩子了！#扯闲篇儿#

2.2 Sina Weibo

The number of net users has increased to 51.3 billion in December 2011 in China [16]. There are many reasons for this fast growth. First, economic development results in massive participation with countless number of updates, opinions, news, comments and product reviews. Secondly, China has many different dialects and two Chinese speakers from different cities may be completely unable to understand one another. Most people who cannot speak to one another can still communicate in writing. The Internet provides a platform to enable people from different cities to communicate readily. Lastly, the Chinese version of twitter brings more mobile netizens and is creating a new culture in China. In 2011, the number of Sina Weibo users has increased to 1.95 billion. Among these, 34% access Sina Weibo by mobile phone [81].

Twitter, the original microblog service, has been blocked in China, but major websites have launched their own Twitter clones, and these have become an important alternative channel for information. Microblogs are called 微博 (Weibo) in Chinese; 微 (Wei) means micro, and 博 (bo), which comes from the first sound of blog, means large.

I use microblogs as my experimental corpus since microblogs have several good features which fit my requirements. Microblogging entails real-time sharing of con-

tent that is specific to a time and audience. This is in contrast to traditional media that has a longer news cycle and a prolonged process that makes the content and timing of the content more uniform. Compared to other online corpora, microblogs are distinguished by short sentences and casual language. Most microblog sites limit the maximum length of a post to 140 UTF-8 characters[1], demanding precise and clear execution. Microblogs are important birthplaces of new words. Moreover, microblog posts have timestamps. This information is very useful in terms of the emergence of new words. From our experiments, the average length of posts is 24 characters. This short length may reflect the large percentage of mobile users.

SINA Weibo has the most active user community among the other top 3 portal sites: Tencent, Sohu and NetEase. Moreover, SINA has the best relationship with the Chinese government. Recently, even government employees and government media use SINA weibo to broadcast news and other events [87].

There are many differences between Twitter and Weibo. The most important one for this research is that Weibo does not publish submitted posts right away, instead it takes two minutes for an administrator to read the post. This has changed recently, however. Now the server checks submitted posts right away, if the post contains a keyword which is on a sensitive keyword list, the post will be suspended until a human can read it. If the contents of the post are not sensitive, then it will be published, if the contents are sensitive then this post will be suspended indefinitely.

[1]Compared to English, 140 Chinese characters can present 3-5 times more information.

Chapter 3

RELATED WORK

With regard to natural language processing and social media, my work is distinguished from related work mainly in that related work has not addressed two challenges specific to Internet online corpora: the frequent use of neologisms and the turbulent changes in topics. I address both of these by using time sequences in the corpus for extra information.

Natural language processing can be a very powerful tool for understanding social aspects of content through data mining. Chao and Xu [13] show some interesting demographics of hate groups on a blog platform. Kwak *et al.* [49] study the dynamics of Twitter and find that re-tweeting causes very fast diffusion of information. Asur *et al.* [3] model the formation, persistence, and decay of trends, and show evidence that, surprisingly, factors such as user activity and number of followers are not strong determinants for the creation of trends. Specific to China, Wei [78] discusses new media research papers published in Chinese-language scholarly journals. Wang *et al.* [77] studies Chinese-language scientist bloggers.

There has been some work to characterize deletions on Weibo, which are related to topic extraction because they summarize a set of Weibo posts. Bamman *et al.* [4]

use a log-likelihood comparison to word prevelance on Twitter and deletion rate to summarize deleted posts. Fu *et al.* [24] use a ratio that is based on uncensored *vs.* censored posts. Bamman *et al.* use Wikipedia as a lexicon, constructing their list of potential keywords by finding Chinese-langauge Wikipedia articles with matching articles in the English-language version of Wikipedia. Fu *et al.* use a standard Chinese text segmenter. Both of these approaches are based on a lexicon and severely limit the set of keywords that can be extracted from social media text. King *et al.* use a technique that it is claimed "does not require (inevitably error prone) machine translation, individual classification algorithms, or identification of a list of keywords associated with each category. [45]" This technique had previously been valiated on the English langauge [34]. Note that these works are all focused on characterizing a certain kind of post related to censorship, while my work's aim is to extract topics from Chinese-langauge microblog text on a much larger scale.

Yu *et al.* [89] exams trends in Chinese social media based on user and retweet properties, and concludes, "People tend to use Sina Weibo to share jokes, images and videos and a significantly large percentage of posts are retweets. The trends that are formed are almost entirely due to the repeated retweets of such media content."

Leskovec *et al.* [52] present a framework for tracking trends on a short time-scale, *i.e.*, tracking "memes." They analyzed information flow between traditional media and blogs during the 2008 U.S. Presidential election. They also found that memes, such as "lipstick on a pig," started in the traditional media but moved to blogs within hours. In their mathematical model of memes, only imitation and recency are needed to qualitatively reproduce the observed dynamics of the news cycle.

Meme tracking is different from trend extraction in that memes are associated with a single phrase. In Leskovec *et al.*'s study "memes end up corresponding to clusters containing all the mutational variants of a single phrase.[52]" One of the important aspects of my approach presented in Chapter 5 is that trend extraction

is more than just clustering. My approach uses temporal information as a proxy for context, which can extract topics using the full corpus rather than cluster particular memes.

Chen *et al.* [14] describe methods for detecting posts from hidden paid posters. These posters are paid to post content that influences opinion related to, *e.g.*, a social event or business market. In China these posters are referred to as the "Internet water army." Specifically, paid posters who attempt to sway political opinion in concert with censorship efforts are also referred to as the "50-cent party."

Wang *et al.* [76] conduct semantic analysis on Twitter posts to predict hit-and-run crimes. Latent Dirichlet allocation (LDA) is used to reduce the feature dimension. Linear modeling is applied for the prediction method.

In the following parts of this chapter, surveys of previous work in phrase extraction and topic extraction are presented separately.

3.1 Phrase Extraction

Unknown word extraction in Chinese is a very challenging problem. There are two main approaches for phrase extraction: the syntactic/semantic approach and the statistical approach.

The syntactic/semantic approach depends on a predefined grammar and lexicon to parse the text into a hierarchical structure. It is almost impossible to list all the words in a dictionary, including named entities, compound words, and new words. So the statistical approach is in general more efficient than the syntactic/semantic approach [60].

Goh [26] presents an approach based on tagging characters and then applying

support vector machines and maximum entropy models to the appropriate features. Lu *et al.* [67] propose the use of ant colony optimization for unknown word extraction. Cortez and da Silva [18] have developed an unsupervised approach to information extraction by text segmentation. Shuai [92] proposed a Weibo-oriented method for unknown word extraction which first groups the corpus into multiple categories by K-means, then derives a morpheme set from each category based on term frequency. Adjacency degree is introduced to judge the correctness of the extracted unknown word. UNExtract [88] targets low frequency meaningful words by using an overlap variety method to measure the accessor varieties of the overlapping strings. Xin [40] focus on a short texts corpus. News titles are collected and the potential unknown words are classified into either a single-character model or an affix model. Yu-Chin [56] aims at long unknown Chinese keywords by utilizing an existing parser and filtering the keywords by TF-IDF values. If a newly composed word has a higher unknown word coefficient threshold than a pre-defined value, then it will be treated as a newly discovered unknown word. Later, long keywords are concatenated from parsed words and newly discovered words. Chia-Ming [50] uses a Support Vector Machine classifier to learn the extraction rules and test the proposed approach on Chinese Buddhist texts, which are considered to be difficult to perform natural language processing on. Sun [73] presents an adaptive online gradient descent training method based on feature frequency information, which can achieve an order of magnitude faster performance in terms of training time, with equal or higher accuracies.

Tibetan new word extraction is investigated in Jiang *et al.* [39]. Vector space module similarity and information entropy are applied on more than 18 Tibetan network media corpora of Tibet, Qinghai, Sichuan, Gansu and Yunnan. Jakkrit improves boosting-based ensemble learning to extract unknown words in the Thai language [74]. Note that my proposed work does not need to completely solve the problem of unknown word extraction, since it does not matter in the Pointillism

approach which words are unknown *vs.* known. The Pointillism approach is based on trigrams and their frequency information, and can extract topics without any lexicon.

There are also several notable entropy-based approaches to word segmentation. Kempe [44] describes a technique for segmenting a corpus into words without information about the language or corpus, and no lexicon or grammar information. This technique works with corpora containing "clearly perceptible" separators such as new lines and spaces. Jin and Tanaka-Ishii [41] propose an unsupervised word segmentation algorithm that is based on the entropy of successive characters at word boundaries. Zhang *et al.* [59] propose a method that is based on a maximum entropy model. Zhikov *et al.* [95] utilize the local predictability of adjacent character sequences, which is analogous to entropy. My work also exploits entropy, but the main insight that will allow the Pointillism approach to handle neologisms and other unknown words is that social media has distinctive temporal patterns in word usage. Entropy will be a second step in our framework that comes after frequency of occurrence analysis.

3.2 Topic Extraction

Language modeling is an indispensable ingredient in many natural language processing applications [61].

The goal of statistical language modeling is to estimate the probability distribution of natural language. Assume we have a finite word set $W = \{w_1, w_2 ... w_n\}$, then a sentence in the language is a sequence of words w_i. The set of all sentences with vocabulary W is an infinite set. In information retrieval, language modeling is related to the classification of documents.

N-gram-based Language Models

The n-gram language model estimate probability of each word given prior context [10]. For example, one might estimate the probability of "pong" after given prior context "Do you like to play ping":

$$P(pong|\text{Do you like to play ping})$$

An n-gram language model only uses the $n-1$ words as prior context:

$$p(w_1, w_2, \ldots, w_n) \approx$$
$$P_{Unigram}(w_1)P_{Bigram}(w_2|w_1)\cdots\prod_i^m P_{n-gram}(w_i w|i-n+1,\ldots,w_{i-2},w_{i-1})$$

The assumption underlying n-gram language models is the *Markov assumption*, which assumes that future behavior only depends on recent behavior.

The simplest way to solve an n-gram language model is maximum likelihood estimate (MLE):

$$p(w_n|w_1, w_2, \ldots, w_{n-1}) = \frac{C(w_1, w_2, \ldots, w_n)}{\sum'_w C(w_1, w_2, \ldots, w_{n-1}, w')}$$

$C(w_1, w_2, \ldots, w_n)$ are the frequencies of occurrence of the n-grams $w_1, w_2, \ldots, w_n$ in the corpus.

The n-gram language model is faced with the problem of missing information before the preceding $n-1$ words, called *long-distance dependencies* [84].

Statistical Topic Models

Topic language models are based on the concept that documents consist of topics with a probability distribution. Again, topics also consist of words with a probability distribution. The main job of topic models is to explore the co-occurrence

relationship between words-documents or words-words. Long-span latent topical information, which is hard to extract in an n-gram model, is overcome in topic models.

Document topic models model "word-document" co-occurrence relationships. Here, $P(z|D)$ denotes the distribution of topics z given document D. $P(w|z)$ represents the probability of word w in a particular topic z.

The assumption underlying the topic model is that to make a new document D, topics z are chosen under a probability distribution $P(z|D)$, then words w are drawn according to word distribution $P(w|z)$ in the corresponding topic z. Topic models use statistical techniques to invert this process, inferring the set of topics z that were responsible for generating the document D. This is called a generative model, which contains a set of latent variables. The goal of fitting a generative model is to find the best set of those latent variables that explain the observed data.

The model specifies the following distribution over words within a document:

$$P(w_i) = \sum_{j=1}^{Z} P(w_i|z_i = j)P(z_i = j)$$

where Z is equal to the number of topics z.

Latent Semantic Analysis

Latent semantic analysis (LSA), introduced by Deerwester et al. in [25], uses a term-document matrix to describe documents. LSA, based on the vector space model, assumes that words that have similar meaning appear in similar pieces of text. Singular value decomposition (SVD) is a critical step used to reduce the dimension and extract the relationships between the words and documents [63] .

LSA has been applied in different domains with with remarkable success, such as in word sense discrimination [53]. However, due to its overly simple statistical foundation that assumes all noise is white Gaussian additive noise, it has several deficits. For example, LSA assumes that words and documents follow a joint Gaussian model,

but actually a Poisson distribution has been observed. Probabilistic latent semantic analysis (PLSA) was invented to overcome this problem by using a multinomial model.

The Probabilistic LSA or PLSA model [29] has a more solid statistical foundation, since it is based on the likelihood principle and defines a proper generative model of the data. In PLSA, the LSA and Expectation–Maximization (EM) algorithms are combined to identify the probability distribution of mixtures.

$$P(w_i|D_d) = \sum_{i=1}^{Z} P(w_i|z_i)P(z_i|D_d)$$

One problem with the PLSA model is that it is difficult to test if the model is generalizable to the new documents, because the PLSA model makes no assumptions about the how the mixture weights are generated.

Many extensions on LSA and PLSA are invented to solve specific problems. Zhang *et al.* proposed Cross-Lingual Latent Semantic Analysis (PCLSA) to incorporate a bilingual dictionary into a probabilistic topic model [91]. The PLSA model was extended by regularizing its likelihood function with soft constraints which are defined based on a bilingual dictionary.

Latent Dirichlet allocation

Blei *et al.* introduce a Dirichlet prior for the topic distribution to extend PLSA [9]. LDA is a hierarchical Bayesian model.

In LDA, a document set D, containing M documents, is modeled as probability distributions over K topics, which again are modeled as a probability distributions over a set of words $< w1_{,}w_2, ..., w_N >$.

First, LDA samples a K-dimensional vector θ as topic distribution based on a

Dirichlet prior parameterized by α. Then, a topic z is sampled from θ which is generated in the first step. A word w from $p(w|z.\beta)$ is also sampled, which is a multinomial probability conditioned on the topic z, where β is parameter of the Dirichlet prior on the pre-topic word distribution. This step is repeated until all words w in document d have been generated.

$$p(D|\alpha,\beta) = \prod_{d=1}^{M} \int Dir(\theta_d|\alpha_1,\ldots,\alpha_T) \left(\prod_{w_i \in D_d} \left(\sum_{j=1}^{T} p(w_i|z_j,\beta) p(z_j|\theta_d) \right) \right) d\theta_d$$

The multinomial distribution $p = (p_1,\ldots,p_T)$ of the probability density of a T dimensional Dirichlet distribution is:

$$Dir(\alpha_1,\ldots,\alpha_T) = \frac{\Gamma(\sum_j \alpha_j)}{\prod_j \gamma(\alpha_j)} \prod_{j=1}^{T} p_j^{\alpha_j - 1}$$

LDA generates a more smoothed topic distribution in the document. The parameter α_j determines the smoothness of the document distribution. α_j moves the topics away from the corners of the simplex.

Hisano *et al.* [28] use a large database of business news to extract relevant news that influences trading activity. LDA is used to decompose news information into "thematic" features. Feature Latent Dirichlet Allocation (feaLDA) [15] is a supervised generative topic model which automatically detects Web APIs associated website. To improve the scalability of inference of LDA, LDA in MapReduce (Mr. LDA) uses variational inference which fits into a distributed environment better than the more commonly used Gibbs sampling [90]. Fei *et al.* [57] propose self-adaptive LDA modeling, which uses the highest average sine similarity of all topics as a metric to choose the best topic model. The joint Event and Tweets LDA (ET-LDA) [35] describes a joint statistical model which characterizes topical influences between an

event and its associated tweets. Sungwoo Lee *et al.* [51] adopt LDA on smoothed term frequencies, which estimates the topic trend changes of blogs by weaving time slices.

3.3 Independent Component Analysis

Independent Component Analysis (ICA) aims to recover a set of independent signals from a set of measured signals, called blind source separation (BSS). A number of algorithms for performing ICA are proposed, including InfoMax [7], JADE [96], and FastICA [36].

X, a column vector matrix, is used to denote the observed or measured signals. S, a row vector matrix, denotes the original independent signals as S_i. Each measured signal can be expressed as a linear combination of the original independent signals:

$$X_i = a_1 S_1 + a_2 S_2 + ... + a_n S_n$$

We can express the entire system of n measured signals as:

$$X = AS$$

A is the mixing matrix that generates X from S. The goal of ICA is, given X, find S and A.

Preprocessing Data

X is preprocessed to have a mean of zero, a variance of one and zero correlation. This is called *whitening*, or *sphering*.

Centering is achieved by subtracting the mean of the signal from each reading of that signal. Then we form a covariance matrix, which is the covariance between each pair of signals.

$$C = VV^T$$

$$x' = V^{\frac{-1}{2}}V^T x$$

After whitening, we calculate a new mixing matrix. The inverse of the whitening operation on the new mixing matrix A' can be used to recover the original mixing matrix A.

$$x' = (V^{\frac{-1}{2}}V^T A)s = A'S$$

Now the new mixing matrix A' is orthogonal.

Finding the Mixing Matrix

A method similar to Newton's method is applied to find a mixed signal with minimum nongaussianity and maximum independence.

To approximate a function, a Taylor series can is used as follows:

$$f(x_0 + \varepsilon) = f(x_0) + f'(x_0)\varepsilon$$

To find the zero of $f(x)$, ε is the step needed from x_0 to make $f(x_0 + \varepsilon) = 0$.

So,

$$\varepsilon = \frac{f(x_0)}{f'(x_0)}$$

Applied iteratively, we have:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$J(x)$ is the negentropy approximation. We want to find the minimum value of $J(x)$, which is where $J'(x) = 0$.

ICA has been used widely in graphics processing in the medical field, such as separating electroencephalogram (EEG) signals [62], functional magnetic resonance imaging (fMRI) signals [11], and facial recognition [93].

Next I compare ICA to PCA. Principal component analysis (PCA) is a popular way to find useful data or image representations statistically. For example, EMPATH[19], Eigenfaces [75] and Local Feature Analysis (LFA) [64] are all based on PCA.

In PCA, the given data is decomposed as a linear combination of principal components. After PCA processing, the PCA coefficients become uncorrelated, which means the coordinates are independent of each other. The problem of PCA is that it cannot separate high-order dependencies in the joint distribution. It does successfully separate pairwise linear dependencies.

For image processing applications, for example, the ICA approach has been proposed to deal with high-order relationships among image pixels [20, 6] because of the shortcomings of PCA. To generate spatially localized features, ICA generates statistically independent basis vectors, while the basis vectors generated by PCA are only linearly decorrelated [5]. In the other words, PCA decomposes the input data depending on second-order statistics, which is generating new data with minimum mean-squared reprojection error. In contrast, ICA handles both second-order and higher-order dependencies. So we can also say that ICA is a generalization of PCA [22].

Algorithm 1 The pseudocode for the implementation of ICA.

Input: Observations X, Each row is one observed signals, *i.e.* the frequency of each trigram.

Output: S : independent signals; Each row is one independent signals; W : decomposition matrix; A : mixing matrix

```
Center X (remove the mean from x)
Whiten X (uncorrelated the components)
for i = 1 to n do
    w = random vector
    orthogonalize initial vector w in terms of the previous components
    normalize w
    while w not converged do
        w = approximation of negentropy of wTx
        orthogonalize w in terms of the previous components;
        normalize w;
    end while
    W (i,:) = w;
end for
S = W*whitenedx
A = invert (W)
return S,A
```

In the topic extraction field, X is the term-document matrix and the latent independent signals are the topics, which can be used for classification. There are several studies on applying ICA to topic extraction.

Same as LSA, ICA is also a dimension reduction technique. However, comparing with traditional methods such as LSA and LDA, ICA-based topic extraction uses higher order statistics. As mentioned in the last section, LSA projects data to a

subspace which is spanned by the most important singular vectors of the term-document matrix, X. LSA neglects higher order correlations, only using second-order statistical information.

ICA's use of higher-order statistics is a major reason why it is used for blind source separation. Topic extraction is also a natural problem for ICA, since each topic can be viewed as an independent signal.

Applying ICA in the context of text data was first presented by Isbell [37], Kolenda [27] and Kaban [43]. They assume that the text consists of independent topics which have a probability distribution over the terms. They discover topics and representivie terms in each topic from the text set.

Pu *et al.* [66] use LSA to preprocess the data and then use ICA to classify the text. Kumaran *et al.* show that the ICA with trigram model performs better than the ICA with a bigram model when using ICA for automatic speech recognition [48]. Srinivasan [72] studys the PCA, ICA and non-negative components in term selection and various features for text classification. Yokoi *et al.* select only the useful independent component by a maximum distance algorithm (MDA) and apply ICA in a recommendation system [85]. Yanagimoto *et al.* use ICA to select index terms for document vectors [86]. Väyrynen *et al.* shows that, comparing with SVD, the explicit semantic features for words produced by ICA align more to cognitive components resulting from human activity [33]. Sevillano *et al.* improve the reliability of ICA-based text classifiers [2].

ICA has been used in text-to-speech synthesis [1], segmentation of chat history [46] and concept location of source code [17]. Rapp presents an ICA-based word sense disambiguation method in [68].

ICA is also used for extracting linguistic features [30, 32]. Especially in Honkela [30], which uses emails as a corpus to present that the features extracted by ICA cor-

respond to well-known semantic and syntactic categories. Hyvärinen *et al.* compare three unsupervised methods: ICA, LSA and self organizing maps (SOM) in terms of linguistic feature extraction [31]. The results show that ICA outperforms LSA and SOM based methods.

Hamamoto *et al.* [38] compare the performance of SVD, clustering, and ICA in extracting features from topic detecting context. The experimental results shows that ICA is the best among the three methods with wide windows and it can extract more topics. SVD is the worst comparing with the other three methods.

Even though those previous works show that ICA has great potential in unsupervised topic extraction, there is only two studies that I know of that applies ICA in a corpus with timestamps. Kolenda and Bingham applied ICA on chat room conversations [47, 8]. Kolenda collected 8.5 hours of a chat session in the CNN chat room, which came out to 4900 lines. The 1114 documents are generated by a window size of 300 characters and 150-character intervals. 4 topics with keywords are extracted successfully. Bingham collected contigues stream of 24 hous from 3200 users with 25,000 chat lines. Again a slide window of size of about 130 words is used. 10 topics with 15 keywords in each are collected successfully.

Chapter 4

PHRASE EXTRACTION

In this chapter, I show how the Pointillism approach can be used to extract longer words and phrases. This chapter is organized as follows. We already gave some background about the Chinese language and Sina Weibo in Chapter Section 2. Section 4.1 discusses some preliminary observations that motivate the approach and then Section 4.2 and Section 4.3 explain the key procedures and algorithms for discovering bigram and trigram trends and then concatenating trigrams to form a word or phrase. Our experimental methodology and results are explained in Section 4.4. Then the discussion in Section 4.5 is followed by the conclusion of the chapter. Related works for this chapter were already discussed in Chapter 3.1.

4.1 Observation and Analysis

To illustrate how a word can create ambiguities for word segmenters, we use 中华人民共和国 (People's Republic of China) as an example. This is neither a neologism nor an unknown word that cannot be found in the dictionary, but it is a good example of segmentation ambiguities and gram trends.

The word 中华人民共和国 (People's Republic of China) is seven characters long and has smaller words within, as shown in Figure 4.1.

Figure 4.1: People's Republic of China.

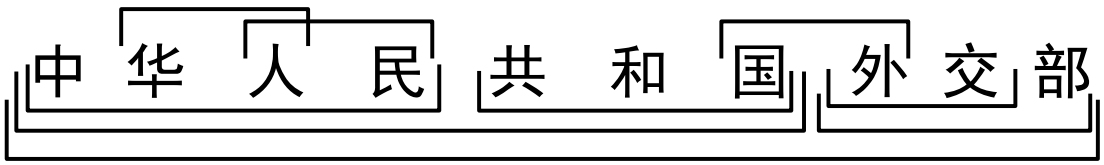

Figure 4.2: Ministry of Foreign Affairs of the PRC.

The first two characters, 中华, are usually not used as a word independently in modern Chinese, though it has a complete meaning (China) and is often used as a part of brand names and company names (this can be called a *semi-word*). Digging further within the word, in 人民 (people) 人 is a word (human) but 民 (civilian or folk) is not a stand-alone word. Complicating things further, there is a hidden word 华人 (Chinese) across 中华 and 人民. As another example, while the proper segmentation of 中华人民共和国外交部 shown in Figure 4.2 (Ministry of Foreign Affairs of the PRC) is 中华人民共和国/外交部, another word, 国外 (overseas), could also be erroneously extracted. Consequently, a good segmenter should not treat 国外 as a word in this case, even though 国外 is in the dictionary and 中华人民共和国外交部 is typically not in the dictionary.

Figure 4.3 shows the plot of trigram frequency of occurrence for a period of 47

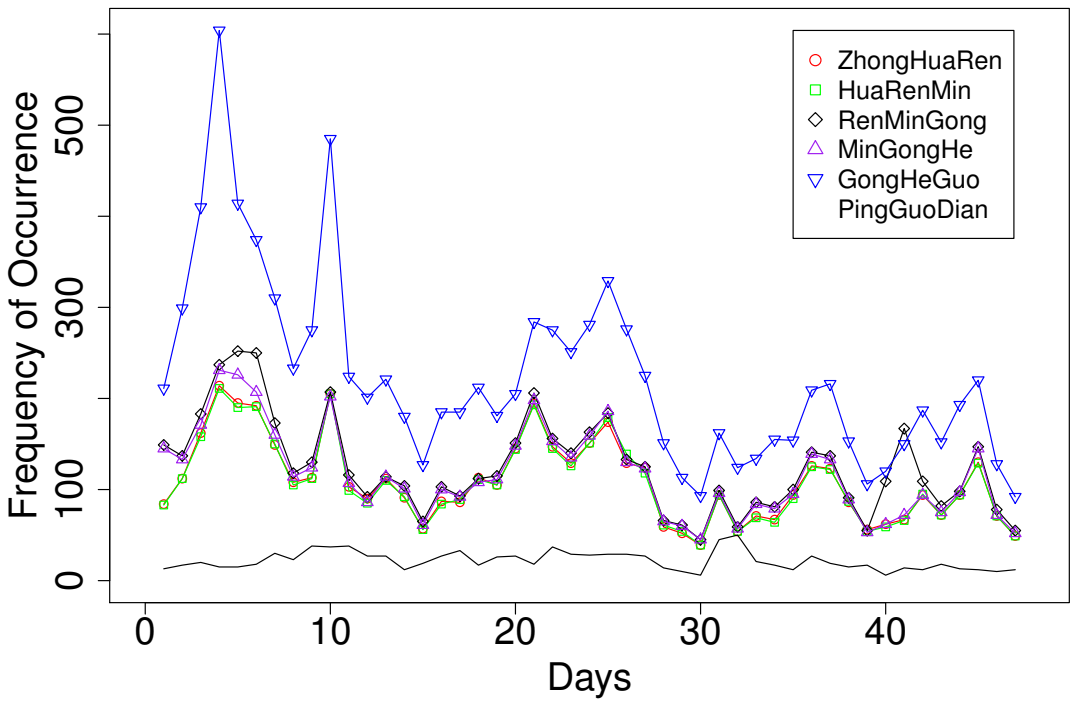

Figure 4.3: Trigram trends for 中华人民共和国 on Weibo.

days[1]. 中华人民共和国 has 5 trigrams: 中华人 (ZhongHuaRen), 华人民 (HuaRenMin), 人民共 (RenMinGong), 民共和 (MinGongHe), and 共和国 (GongHeGuo). What can be seen in Figure 4.3 is that trigrams from 中华人民共和国 have a very distinctive temporal correlation when compared to other trigrams, such as the trigram 苹果电 (PingGuoDian, the first three characters of 苹果电脑, or Apple Computers). The x-axis is time in days. The y-axis is the number of occurrences of the trigram per day for our dataset, which at the time this data was taken was only about 2% of all Weibo posts. What is most interesting are trigrams that are not words themselves, but through time correlations serve as a sort of glue to hold

[1]The data is from 23 July 2011 to 18 August 2011 and 24 August 2011 to 11 September 2011. Due to system failure, we missed the data from 19 August 2011-23 August 2011.

trigrams which belong to the same word together.

共和国 has a higher frequency of occurrence than the other words (Figure 4.3). This is because 共和国 (Republic country) has a complete meaning, and can be used in many other contexts. The other four trigrams, which do not have a complete meaning, fluctuate together demonstrating the fact that they mostly appear in the same context as 中华人民共和国. There is a separation of 人民共 around the 40th day in Figure 4.3. This is because 人民 (people) is such a common word that it can affect the trigram containing it easily. Also, the trigram 人民共 appeared in other contexts which had a trend for three days around the 40th day. 华人 is also a word, as 人民 is, but 华人 is not as commonly used as 人民, thus it does not influence the trigram trend of 中华人 and 华人民 as much.

Another interesting observation can be found in Figure 4.4, which shows some potentially interesting dynamics of a particular censored keyword and neologism. Day 0 for the graph is 15 November 2011, and the y-axis is on a log-scale plot where 1 corresponds to no posts for that particular day. The events of the Wukan village protests are described on Wikipedia [80]. On 9 December 2011 Xue Jinbo was arrested, and he died in police custody on 11 December 2011. The first peak of the original word (乌坎 Wukan, in dashed green) occurs on 12 December 2011, the day that protests began. By 14 December 2011, when posts for Wukan hit zero, potentially due to censorship, thousands of police had laid siege to the village. The neologism (乌坎 Niaokan, in solid red) came into existence on 13 December 2010. The peak for wukan on 21 December 2011 corresponds with an announcement that the government and the villagers had reached a peaceful agreement.

The observations that there are time correlations that appear among trigrams, and that these correlations are related to current events, lead to the question: can we use this feature to find words, phrases, memes or even find topics? In this chapter, we examine the possibility of using only time correlation information of grams to

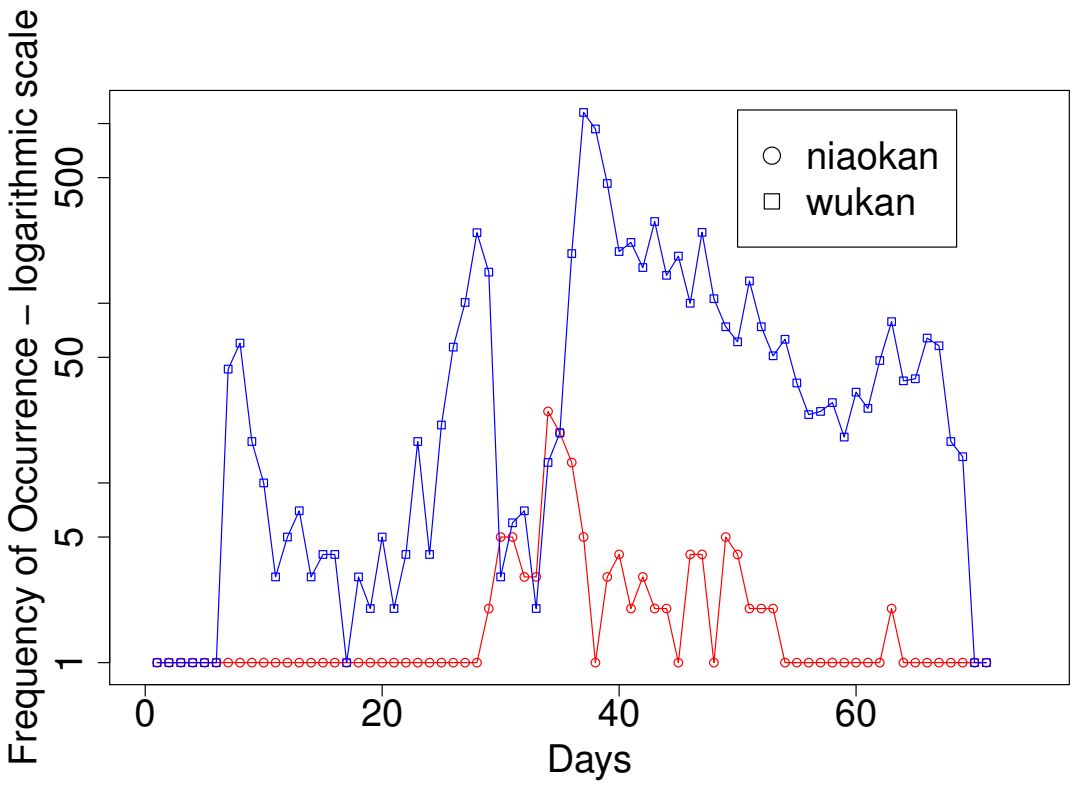

Figure 4.4: Bigram trend for 鸟坎 Niaokan. 鸟坎 is a neologism for 乌坎 (Wukan), a village in southern China that was the site of considerable social unrest in late 2011. Day 0 is 15 November 2011, and the y-axis is on a log scale with 1 corresponding to zero posts for that particular day.

concatenate words and phrases without a lexicon or knowledge of the grammar. Though bigram trends would be more appropriate for the example in Figure 4.4, in this document we defer on shorter words of only two or three characters and focus on longer words and phrases that require putting trigrams together to form them. I defer the discussion about why we use trigrams until Section 4.5.2.

4.2 Procedures

Considering patterns in the frequency of occurrence of trigrams is an important optimization. Ignoring case and punctuation, in English, there are at most $26^2 = 676$ bigrams and $26^3 = 17,576$ trigrams. Chinese has on the order of thousands of characters that are in common usage, and Internet users use many characters that are not in common usage. In our current corpus from about 240 days of Weibo posts, there are approximately 33 thousand characters, making up approximately 14 million observed bigrams and 264 million observed trigrams. Each of these grams is represented by a time series for potentially months or years. Frequency of occurrence correlations can be found relatively efficiently compared to the high computational cost of considering larger grams.

Step 1: Collecting posts with time sequences.

Starting from 23 July, 2011[2], we send a request every second to the public timeline and the Weibo server returns roughly 200 posts responding to each request. These 200 posts are not continuous in terms of post ID[3]. For each request, most of the returned posts are submitted 2 to 5 minutes before the requesting time[4]. However, at around 4 or 5am Beijing time, when the traffic is lowest, the server might even return posts from over 24 hours ago. The parsed data is saved in an HBase database for later analysis.

We found that there are many duplicates in the returned posts. At the beginning, the efficiency was only 9%, which means that among total posts we downloaded per hour of 720,000 (200×3600), only about 65,000 were distinct posts. After 19 August 2011, the efficiency even decreased to 5.6%. We conjecture that this may have been

[2]We have collected data continuously until September 2013, but in this chapter, we only use the data from 23 July to 23 March 2012.

[3]Refer to Discussion session for more information about post ID.

[4]The 2 minutes delay is the time for the administrator to monitor the posts.

related to our frequent requests from the same IP and system cache mechanism.

We made several changes to the collector to increase the efficiency. First, we lowered the request speed and sent requests from two keys, each key sends a request every 8 seconds. Secondly, we added a random parameter at the end of each command. Now the total number of posts we get is $200 \times 3600/4 = 180,000$ per hour, and the number of distinct posts is 80,000 per hour, so the efficiency is increased to 44%. Moreover, we used the Tor network to frequently change our IP address [21] to avoid biased data from the server based on client IP address.

The size of the raw data collected by one IP (in JavaScript Object Notation (JSON) file format) per hour is above 700MB. After removing the repeated posts and parsing the JSON file to extract the contents we need[5], the amount of data is reduced to 20MB per hour. The size of the compressed data is about 5.8M per hour. By our estimation, the coverage of the collected posts by one IP address is 1.7% to 10% of the whole. We calculated this when the post IDs were predictable enough to infer posts that had been missed based on ID. For more details on coverage, please refer to Section 4.5. Note that this number will decrease with the increasing microblog volume for each day of Sina Weibo.

Step 2: Count the frequency of occurrence of grams.

Given a Chinese text with time series information, the system will divide the text into trunks of three consecutive characters that are called trigrams. Our system obtains the frequency of occurrence of each trigram hourly. In this document, we mostly use the frequency of occurrence of each trigram for a daily basis.

Step 3: Check if the trigram is a valid root.

For convenience, we call the program connecting the trigrams based on time

[5]postID, text (contents of the post), time of submission, userID, and a flag which shows if the user is verified and the MID.

correlation information "Connector." We call the first trigram we feed into Connector the *root trigram*. Not all trigrams can be used as the root for concatenating trigrams. There are two kinds of trigrams that cannot be root trigrams.

First: trigrams that are rarely used where most of their daily frequencies are zeros. This is important because if a vector has too many zeros this would bias the cosine similarity value.

Second: trigrams which have no obvious fluctuation in our examination period. We measure this by obtaining the cosine similarity value between the trigram and the constant vector $< 1, 1, ..., 1 >$. If the value is higher than 0.98, then it is considered too normal and will be treated as an invalid root for concatenation. We found that the trigrams which have too high of a cosine similarity with $< 1, 1, ..., 1 >$ tend to mistakenly result in extremely long phrases. Commonly used trigrams may even concatenate indefinitely.

Step 4: Find time correlation of grams.

The trigrams that have temporal correlations and overlap will be concatenated together. The detailed algorithms for this step are described in the next section.

4.3 Algorithms

Cosine similarity is used to judge whether two trigrams have correlated trends.

$$cos.Sim = \frac{< A_i, B_i >}{\sqrt{\sum_{i=1}^{n} {A_i}^2} \times \sqrt{\sum_{i=1}^{n} {B_i}^2}}$$

where $<, >$ denotes an inner product between two vectors. There are three possible values for the vectors A_i and B_i. They are the daily frequency of trigrams (FT),

the difference in frequency of trigrams (DFT) and the rate of change in frequency (CFT).

$$CFT_{day}(trigram) = \log_{10} \frac{FT_{day}(trigram)}{FT_{day-1}(trigram)}$$

$$DFT_{day}(trigram) = FT_{day}(trigram) - FT_{day-1}(trigram)$$

During our initial experiments we found that, for some trigrams, CFT cosine similarity works better, and for other trigrams DFT works better. In light of this, we used an SVM (Support Vector Machine) to learn which parameters we should use for connecting the words. The features we used for the SVM were cosine similarity between the *root trigram* and the constant vector $< 1, 1, ..., 1 >$, maximum change rate of the root, minimum FT, maximum FT, mean of FT, and the length of the time period of vectors. The corresponding variable y is a vector of FT, DFT and CFT. We used the libsvm package [12] to train our model. We labeled 100 data points for training by hand, and used this model to decide which vector should be used in finding the relationship between trigrams. Once we decide which vector we are going to use to find similarity, we concatenate the trigrams.

Let us use an English meme, "putting lipstick on a pig", as an example to explain how we concatenate trigrams.

Suppose Connector has thus far produced "putting lipstick on a". The system now considers which trigram from the set {"on a boat", "on a pig", "on a clear" and "on a plain"} it should concatenate next. The *root trigram* is the first trigram, "putting lipstick on", with which Connector starts. The *parent trigram* is the last trigram appended to the phrase (at this point "lipstick on a" is the parent trigram). The *stem trigram* is the one which has the lowest frequency in "putting lipstick on a". We use the median value of the frequency vector to decide which trigram's frequency

is the lowest, and the *stem trigram* should be closest to the real frequency of the phrase. For example, in Figure 4.3, 华人民 is the stem trigram of 中华人民共和国. Note that the *root, parent, and stem trigrams* may sometimes be the same.

To decide whether we should connect a trigram to the phrase we are working on, we compare the cosine similarity value between each candidate trigram and the *root* (*simRoot*), *parent* (*simParent*), and *stem trigram* (*simStem*). We use simScore to measure how close a gram is to the phrase.

$$simScore = max\{simRoot, simParent, simStem\}$$

The candidate trigrams are sorted by simScore. We recursively search for the next trigram to concatenate by a depth-first traversal into highest-score nodes first. Only the top 5 candidate trigrams are considered. The traversal stops when all candidate trigrams have a simScore lower than a threshold, which is set manually to 0.97. We set a threshold of 60 seconds for Connector to concatenate trigrams, to avoid unbounded loops.

$$simPath = \prod_{i=1}^{n-2}\{cos.Sim(simStem, trigram_i)\}$$

where, n is the number of characters in the phrase.

The phrase which has the highest simPath is considered to be the final result. If there are other phrases which also have a *simPath* higher than the threshold, we output them as possible results, as shown in Section 4.5.3.

So far, the algorithm works well when connecting long words, such as 4-character idioms (Section 4.4.3). However, when we try to connect phrases, especially phrases from weibo.com, the unconstrained precision of Connector was initially only around

0.50. We found that Connector fails when the phrases contain commonly used words, which are 3 characters long, or a bigram word followed by a stop word. For example, in a three word phrase: 纸质书会不会消失 (Will paper books disappear?), 会不会 (will) is a three-character-long, high-frequency word. Even if this longer phrase has a trend in a certain period, it is not as pronounced as the trend shape of 会不会 (will). The cosine similarity of 会不会 with the vector $< 1, 1, ..., 1 >$ is always close to 1. This means the cosine similarity of 会不会 with the root trigram 纸质书 would be lower than the threshold. As a result the phrase would be incomplete (纸质书会不).

To solve this problem, we "suspend" common trigrams by putting them into our selecting scope. In the other words, we connect this trigram to the phrase temporarily. We commit it if one of the children of this suspended common trigram has a high simScore (> 0.98), and remove it if none of its children has a simScore greater than 0.98. If there is still no good candidate, we use these suspended common trigrams as candidates. We identify common trigrams by their frequency and cosine similarity with the constant vector $< 1, 1, ..., 1 >$. In our current implementation, we set the minimum frequency to be 100, and the minimum cosine similarity with $< 1, 1, ..., 1 >$ to be 0.97. By adding this, the unconstrained precision increased to above 0.80.

Now, what if a common trigram is the last trigram of the phrase? That means there are no better candidate trigrams except for the common one, and this common trigram has no children that have a similar trend with the phrase. In this case, we bracket the last character of the common trigram and stop. For example, in phrase 普通青年VS文艺青年 (Common youth *vs.* educated youth), the result returned by Connector is 普通青年VS文艺青年(们), Where 们 is the pluralization of a pronoun.

There is one more method that we employed, which we call "binding." When two continuous trigrams have a similarity greater than 0.995, Connector will bind

them together, which means they will be selected, or not selected, as a unit. After implementing these techniques, the unconstrained precision increases to about 0.90 for long phrases.

If we know the seed trigrams belong to some spurred topics, such as the words from hot keywords of weibo.com, we avoid the grams with very low frequencies. If we are trying to find a well-used phrase which persists for a long period of time, any trigram in that phrase should show up similarly. So if a gram appears every day in 100 days of data, and another gram only appears in 10 days, these two grams probably do not belong to the same phrase.

There are some exceptions:

1. When the trend of the last trigram of a phrase is similar with that of the root, *i.e.*, simRoot is higher than a certain threshold (eg, $> 97\%$), we loosen the condition.

If none of the candidates is good enough, we lower the threshold for simFinal, but we mark those candidates which were added by lowering the threshold as "suspended", which means we just temporarily connect this gram to the phrase. If some of the children of the candidate trigram have a high simRoot value, we keep both the suspended trigram and its best child. Otherwise, we remove this "suspend" gram and its children.

2. When the last trigram of the current phrase is in suspend status (added by rule 1.a), we have to make the rule more strict than normal: the simRoot has to be high enough (the normal case was that any of the three values being higher than the threshold was enough).

4.4 Evaluation

To gain a better understanding of the capabilities and limitations of connecting trigrams into longer words and phrases, we performed three experiments. For these experiments, we were interested in both known words (in the dictionary) and unknown words or phrases (not in the dictionary). Unknown words are the already existing words that are not in the dictionary. They include named entities and neologisms. Named entities can be the names of people, companies, movies, and anything that is given a name. Neologisms are words created recently, they are usually not words before they are created. However, there are also neologisms which give a different meaning to an already existing word. For example, 小姐 (Miss.) can also mean "prostitute" recently. Phrases consist of multiple words. Phrases are not fixed relative to words, since there may be different ways of saying phrases in different places and this may evolve over time.

We collect three different types of word and phrase sets: long words or phrases from Wiktionary, hot keywords or phrases from weibo.com, and 4-character idioms from Wiktionary. Words or phrases in Wiktionary include both known and unknown words and they usually do not have an obvious trend[6]. 4-character idioms are known words and do not usually have obvious trends. Hot keyword lists from weibo.com are newly created words or phrases, and have obvious trend(s) in a certain period.

The procedures of these experiments start from Step 3 in Section 4.2 (Step 1 and 2 have been done before the experiments). The output of Connector is compared with the original dataset, and judged by a human to determine if it is a correct result.

For this chapter, because our emphasis is on connecting trigrams into longer words and phrases rather than on how to detect trends, we assume that the word

[6]A trigram having an obvious trend means it has a deviation from its standard tendency. For example a trigram involved in a news event or a hot topic.

or phrase exists in our database and that the root trigram is given. For this reason we have no negative observations, only positive observations. This is why we focus on precision and do not report recall in this chapter. To compare two scenarios, one where Connector must only match words or phrases in a lexicon and one where a human can judge if Connector constructed a valid word or phrase, we report two different precision scores: lexicon-constrained and unconstrained.

Lexicon-constrained precision (LCP) is calculated as:

$$LCP = \frac{|\{Matching\ Wiktionary\ or\ Weibo\}|}{|\{Valid\ Root\ Trigrams\}|}$$

Unconstrained precision (UP) is calculated as:

$$UP = \frac{|\{Matching\ Wiktionary\ or\ Weibo\} \cap \{Correct\ (human)\}|}{|\{Valid\ Root\ Trigrams\}|}$$

Results matching with Wiktionary or weibo.com are those that are an exact match with the words or phrases listed by them. Correct (human) results are those that are different from the phrases listed by the source, but judged to be correct by a human. This is necessary because there are variations in words, four-character idioms, and phrases where the variants are also correct.

4.4.1 Long words and phrases from Wiktionary

We collected 500 words or phrases which were longer than 3 characters from wiktionary.com [83]. There are 78 invalid *root trigrams*, including the uncommon words that do not appear in our corpus and the ones that share the same *root trigrams* as others. The 422 valid *root trigrams* were fed to Connector, and then we compared the results from Connector to the original phrase set. The time period of vectors

Table 4.1: Precision values for long words in Wiktionary.

	LCP	UP
Named entities (133)	0.5	0.79
New words (16)	0.53	0.93
Phrases (244)	0.44	0.75
4-character idioms (29)	0.90	0.90
Total (422)	0.50	0.79

we use in this experiment is from 23 July 2011 to 18 March 2011. In the 422 valid words, there are 133 named entities, 244 phrases, 29 4-character idioms, and 16 new words. The results for each type are listed in Table 4.1.

Except for 4-character idioms, the LCP scores are less than 0.55. LCP needs to be put in the proper context for our system. For example, net users are more likely to use 中国移动, the shortened form of a name, instead of the full name, 中国移动通信 (China Mobile Limited). 一房一厅 (One bedroom and one living room) becomes 一房一妻制 (One house one wife policy). The latter is a newly created phrase describing a new social phenomenon.

We also found that the results tended to be affected by new events. For example, for one named entity, 中国银行业监督管理委员会 (China Banking Regulatory Commission) in wiktionary, the result of Connector is 中国银行百年行庆 (Bank of China One Hundred Year Anniversary). 中国政法大学 (China University of Political Science and Law) becomes 中国政府对温州 (Chinese government to Wenzhou), which is influenced by the Wenzhou train collision that happened in July 2011.

The fact that UP scores tend to be much higher than LCP scores in our results implies that even with a relatively comprehensive lexicon, such as Wiktionary, the lexicon is incomplete and not up to date. This motivates the Pointillism approach: *my aim in this dissertation is to move away from lexicons.* Using an online corpus and analyzing time enables us to extract newly created words/phrases, and even

topics, with unknown words in real time.

Moreover, we found that when there are several candidate phrases which start from the same root trigrams, Connector would return the one which is used most frequently in a certain period of time, and this one may not be the one listed on Wiktionary. If we shorten the experiment duration and just include the period which had a trend, Connector would just return the exact phrases which were involved in the trend. In light of this, in Section 4.4.2, we use the hot keywords (or phrases) listed by weibo.com and set the experiment duration for from one week before to one week after the peak.

Among the four types of words, 4-character idioms showed the best LCP score. This is because 4-character idioms are fixed and most of them are from ancient Chinese, there are rarely other phrases which share the same root trigrams with 4-character idioms. Both new words and 4-character idioms have high UP scores, which means that if the words or phrases are stable or fixed then they are not influenced by current events heavily. To test this hypothesis, in Section 4.4.3, we collect 4-character idioms only from Wiktionary and test them in a 3-month and an 8-month period separately.

4.4.2 Weibo hot keywords

Weibo.com lists 50 hot keywords, or phrases, hourly, daily and weekly. We used the weekly hot keywords of 2 November 2011 to find out how Connector works on unknown words or phrases with pronounced trends. We picked 7 days before and 7 days after 2 November 2011 as the frequency vector, so the vector length is 15. As in the previous section, UP is the reasonable results judged by a human (39) divided by the valid root trigrams (43). LCP is the results which match the keywords provided by weibo.com (24) perfectly divided by the valid trigrams seeds (43). The UP we

measured in this experiment is 0.907 (39/43), and the LCP is 0.558 (24/43). These are promising results since some of the phrases are longer than 7 characters, contain several words, and especially have stop words in them.

The 7 invalid trigrams are due to the fact that some keywords listed by Weibo are not really hot in terms of what users are actually talking about. For example for 萌物鉴定, the frequency of 萌物鉴 has 13 frequency values equal to 0 and 2 equal to 1. We treated these as invalid root trigrams as described in Section 4.3.

The relatively low LCP value is because we sometimes observe alternate results from the hot keywords provided by weibo.com. For example, 失恋33天 (Love is not Blind) is the title of a movie. The result the connector produced is 失恋33天经典版 (Classical version of Love is not Blind). Similarly for the hot word 乔布斯情书 (A love letter from Steve Jobs), the result of the connector is 乔布斯传 (Steve Jobs: A Biography). The reason for this is because 乔布斯传 (Steve Jobs: A Biography) has higher frequency and fluctuation than 乔布斯情书 (A love letter from Steve Jobs) during the vector period. We only count them as UP but not LCP. For further error analysis, please refer to Section 4.5.

We also used the 8-month data to concatenate those keywords, and the LCP score was 0 with the UP score less than 0.40. After shortening the period to the trend of the data and just including the period of time related to the trend, the accuracy of Connector when connecting unknown words or phrases related to a hot topic increases.

4.4.3 LCP and UP for known words

To test our hypothesis that fixed words or phrases are not influenced by current events as heavily, we collected all 853 Chinese 4-character idioms listed on Wiktionary [82]. The Connector was executed for two periods, one is about a 3-month period (23 July

2011 to 12 September 2011), the other is an 8-month period (23 July 2011 to 22 March 2011). The results are listed in Table 4.2.

The LCP value needs to be put into context because there are two reasons why it is not as high as the precision of common natural language processing tasks using conventional approaches and the UP score in the same experiment. First, some 4-character idioms have common first trigram seeds. For example, 一面之交, 一面之识, 一面之词, 一面之雅, and 一面之缘, have the same seed 一面之, so Connector only returns the one with the most frequent result: 一面之缘. For this case, we only count 一面之缘 as "matches with Wiktionary," the other four are not counted in LCP, but are counted in UP. Secondly, the words listed in Wiktionary may not necessarily be the common usage of the word. For example, people frequently use 一见如故, as Connector gives, instead of 一见如旧, which is listed in Wiktionary[7].

[7]The whole data set and results can be found on the authors' website.

Table 4.2: LCP and UP based on 4-character idioms from Wiktionary.

	Valid Root Trigrams		Match with wiktionary		Correct (human)[1]		LCP		UP	
Data period (month)	8	3	8	3	8	3	8	3	8	3
$Freq > 99$	160	122	123	97	6	3	0.77	0.80	0.93	0.95
$Freq > 29$	344	279	257	214	12	6	0.75	0.77	0.94	0.94
$Freq > 4$	515	484	342	333	23	19	0.66	0.69	0.94	0.95
Total	832	832	364	361	58	56	0.43	0.43	0.92	0.90

[1] In addition to matching with Wiktionary.

Table 4.3: Trigram frequency of 谷歌开发者大会.

	26	**27**	28	29	30	31	01	02	**03**	04	05	06	07	08
谷歌开	1	**68**	5	0	1	1	4	0	2	4	0	0	3	1
歌开发	1	**68**	4	0	1	1	0	0	2	4	0	0	0	0
开发者	49	**127**	43	46	44	65	50	49	**227**	129	38	39	65	63
发者大	15	**56**	5	4	13	10	6	11	**166**	84	14	11	21	14
者大会	15	**56**	5	4	14	9	6	11	**168**	85	14	10	21	13

Though Table 4.2 shows that the 8-month period experiment (columns 2 and 4) gives more words than the 3-month period (column 3 and 5) does, the LCP and UP values are relatively stable. For 4-character idioms, they are usually used when people want to express a strong feeling and are seldom involved in pronounced trends. There are more invalid root trigrams in a 3-month period than in an 8-month period experiment, because some trigrams do not appear often enough.

4.4.4 Error analysis

According to our manual error analysis, the top three errors are the following:

- Stop words (*e.g.* 的(of), 是(is)), numbers (*e.g.* 1998, written in Chinese characters), and high frequency three character words or phrases (*e.g.* 情人节 (Valentine's Day), 会不会 (will)). Even though we "suspend" these kinds of words, they always make the results one character longer than it should be.

- If there are multiple phrases to describe the same thing, or at the time of the experiment period there are multiple memes which contain the same trigrams, then the whole phrase tends to be lengthened, and the results can be a combination of multiple phrases. (*e.g.*, Both 友谊地久天长 (Our friendship will forever last) and 友谊天长地久 (Our friendship will last forever) are correct, the result becomes 友谊地久天长地久 (Our friendship will last forever last).)

- If the target phrases are composed of multiple words and they have different patterns during the experiment period then these independent signals can confuse Connector. For example, the trigram trend of 谷歌开发者大会 (Google's developer conference), one of the hot keywords in our experiment in Section 4.4.3, are shown in Table 4.3. Trigrams 谷歌开 and 歌开发 only have one peak on 27 October 2011. 开发者, 发者大, and 者大会 have two peaks on 27 October and 3 November 2011 respectively. Connector returns 谷歌开发 (Google develop) incorrectly. We hypothesize that there was another 开发者大会 (Developer conference) held on 3 November 2011.

The first two challenges will be the subject of future work. The third challenge is ameliorated by my use of ICA in Chapter 5.

4.5 Discussion

One of the questions this document focuses on is: does the frequent creation and use of new, unknown words exacerbate the existing problems that natural language processing tasks have with word segmentation, or does it provide an opportunity where the temporal variance in these new words can be leveraged to do a better job of the natural language processing task? In this section, I first use an example to show how a trigram which is not a word can actually help with an information retrieval task. Then, I discuss some observations about Weibo's post IDs.

4.5.1 Post density

Before 3 August 2011, Weibo's post IDs were compact. The post IDs would be all even numbers for some amount of time and then change to odd numbers for some

time. So it was easy for us to estimate the posts rate of increase at Weibo on the server side, just from the rate of increase of the post ID. From the distribution of microblog density in Figure 4.5 before 3 August 2011, we can understand that the lowest is at 4am. There are several peaks, *e.g.*, at 1pm and 8pm. This is also why we start a new day at 5am Beijing time for each day, because that is the time when least new posts are submitted.

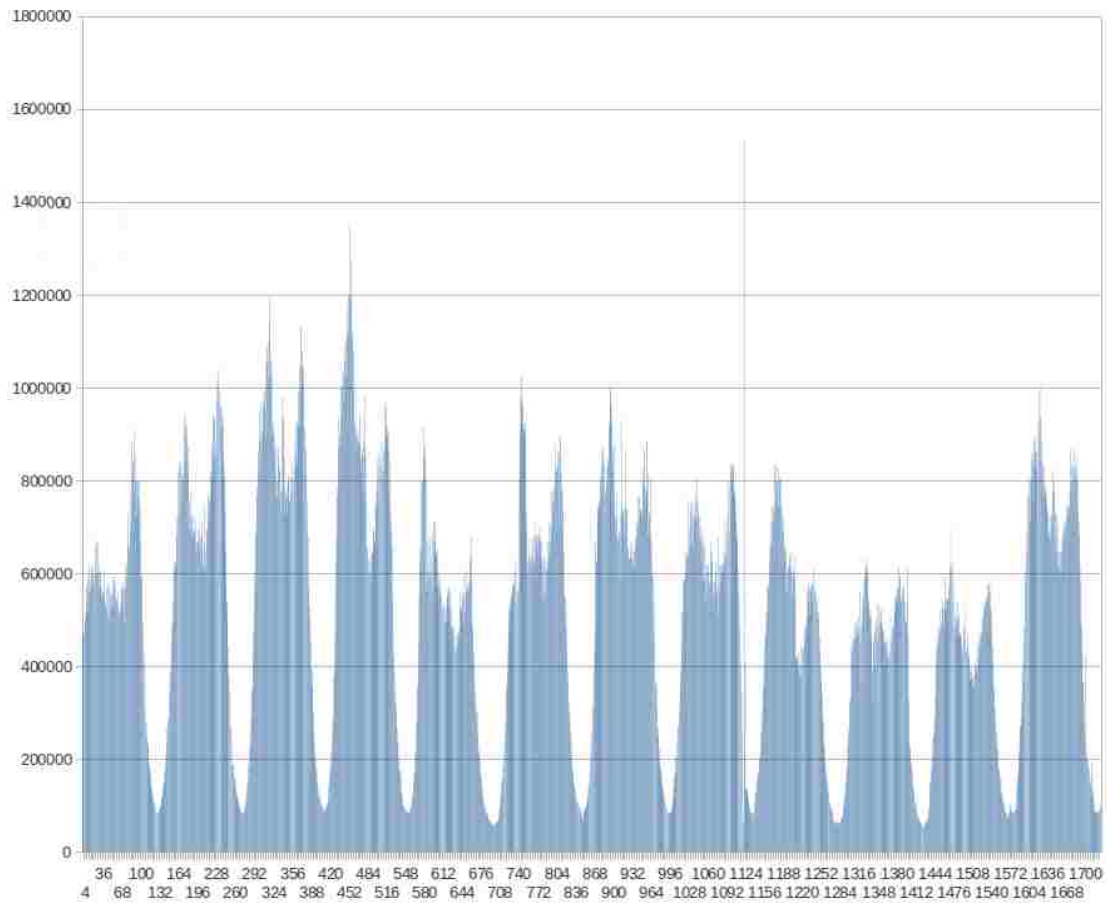

Figure 4.5: ID space before 3 August 2011.

The post ID was extended to 16 bits on 4 August 2011. From Figure 4.6, we can see that the number of posts in the server is now more difficult to estimate. 4 days

after this change, the post ID space was increased by a factor of 2.

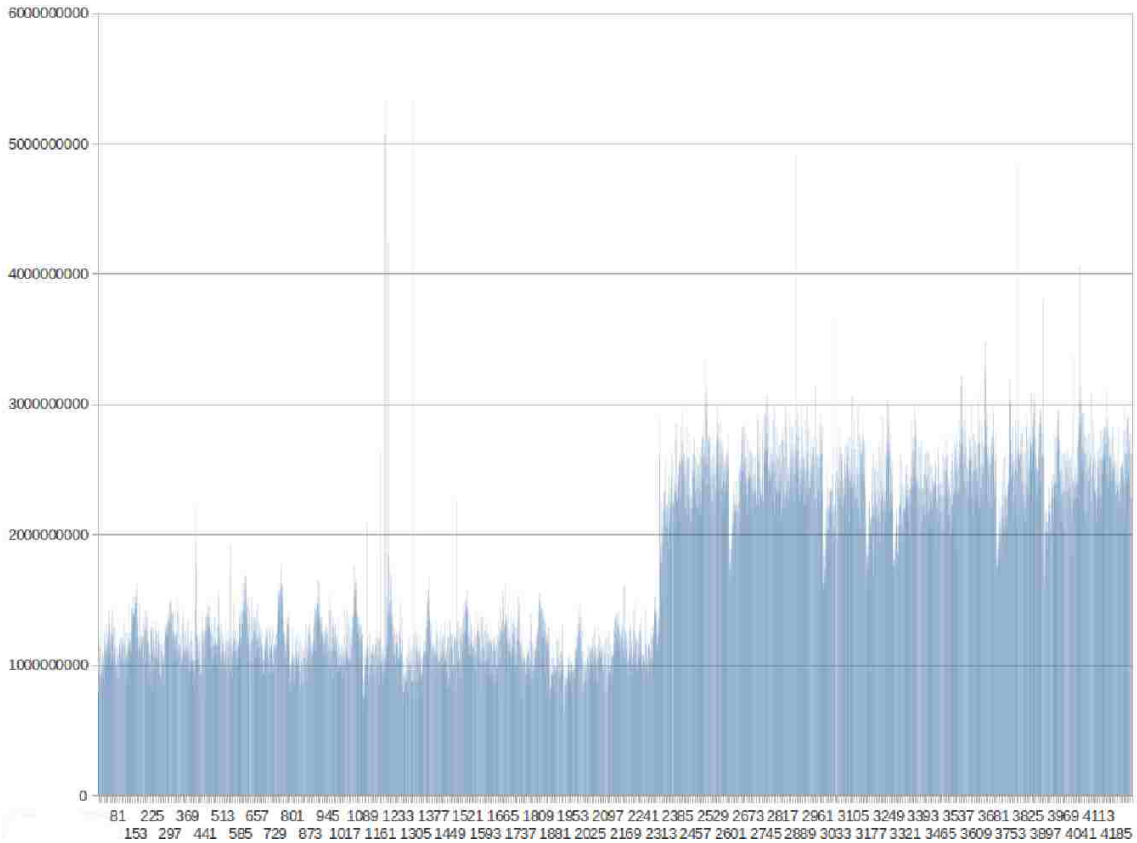

Figure 4.6: ID space after 3 August 2011.

We found an interesting phenomena during this period, that removed posts would appear again, after the ID space was increased. We also found several strange post IDs.

4.5.2 Why trigrams?

As we mentioned in Section 4.1, there can be trends for bigrams, trigrams, quadgrams, and so forth. Here I discuss why we focus on trigrams. By combining individual bigrams, trigrams, and the subject of this document which is longer words and

phrases connected via trigrams, then we can analyze the trends for words or phrases of any length.

Referring back to the example in Section 4.1, 中华人民共和国 can also be successfully concatenated using bigrams. However, we only investigated the feasibility of using trigrams in this document. We chose trigrams instead of bigrams or quadgrams as a tradeoff of efficiency for accuracy.

Monograms have temporal correlation information, but cannot give us order information of the phrase. For bigrams, the bigram trends of 中华人民共和国 are plotted in Figure 4.7. Together with Figure 4.3, we can see that this trend information reflects what we analyzed in Figure 4.1. We can also see that the trigrams information is more efficient than bigrams in terms of serving as "glue" to hold the larger word together. The reason for this is because 70% of Chinese words are bigrams [69]. Trigrams are a good way to remove the noise of the sub-bi-words in the long word, for example, remove the noise of 人民 from 中华人民共和国.

Next, why not longer grams? If we want to find the trend of a phrase that is n characters long, then using n-gram is most accurate for two reasons: First, n-1 overlapping characters decreases the candidates to a larger degree than n-2 overlapping characters. Second, it can eliminate the noise generated by words shorter than n-1.

However, larger grams are computationally expensive. As of 22 March 2012, our dataset has a total of 36,674 monograms, 16,353,985 bigrams, and 323,862,767 trigrams in our database. Figure 4.8 is the daily increment of distinct monograms, bigrams and trigrams[8]. We can see that the distinct grams are constantly growing, though the growth rate is decreasing slightly. For trigrams, it is notable that there are more than 800 thousand new trigrams added to our database every day.

[8]The sudden drop around 26 August 2011 (before day 50) is caused by a system failure for three days around that time. The increase around 12 November 2011 is due to several changes which increased the performance of Collector.

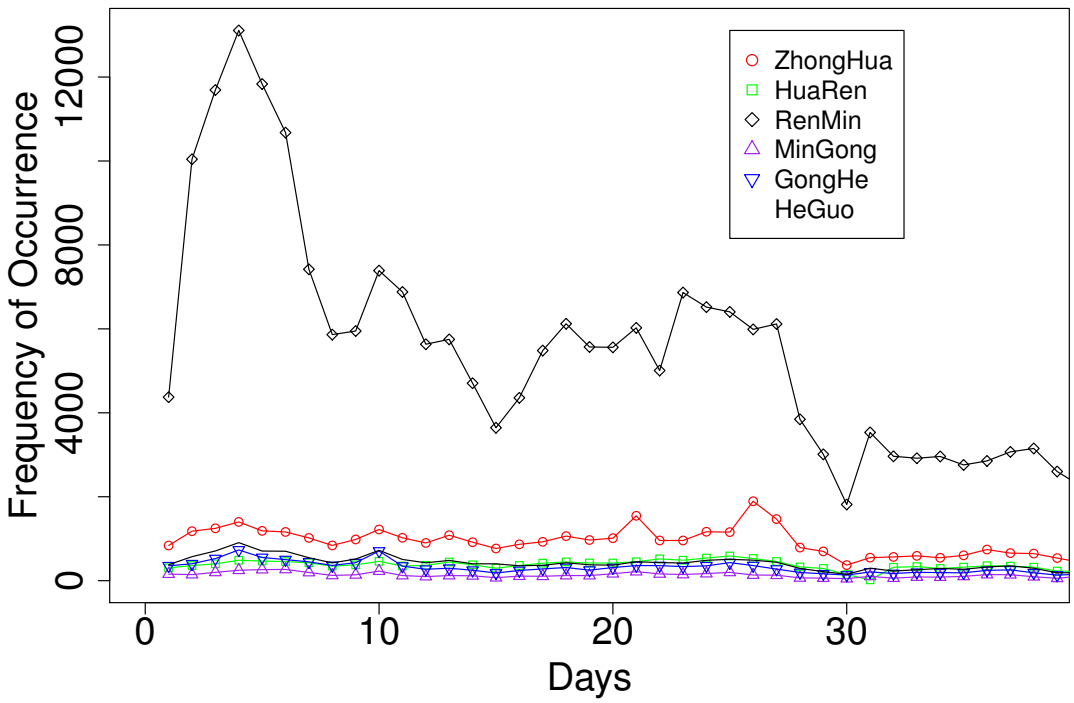

Figure 4.7: Bigram trends for 中华人民共和国 on Weibo.

If we do not consider other characters from other languages which Chinese net users often use, such as the English alphabet, there are 9×10^4 Chinese characters in total. In theory there could then be 8×10^9 bigrams, 7×10^{14} trigrams, and 6×10^{18} quadgrams. It is unrealistic for a computer to track the trend for longer grams than trigrams. On the other hand, the longer the grams we use, the more redundant the information will be. Using n-gram trends to concatenate the phrase equal in length to or longer than n is more accurate than using shorter grams. However, to concatenate the phrases with length m ($m < n$), m-gram works the same as n-gram and n-gram trend only give us redundant information.

Though most Chinese words are bigrams, since the aim of this research is to find trends of memes and new words in their nascent stage, we are interested in

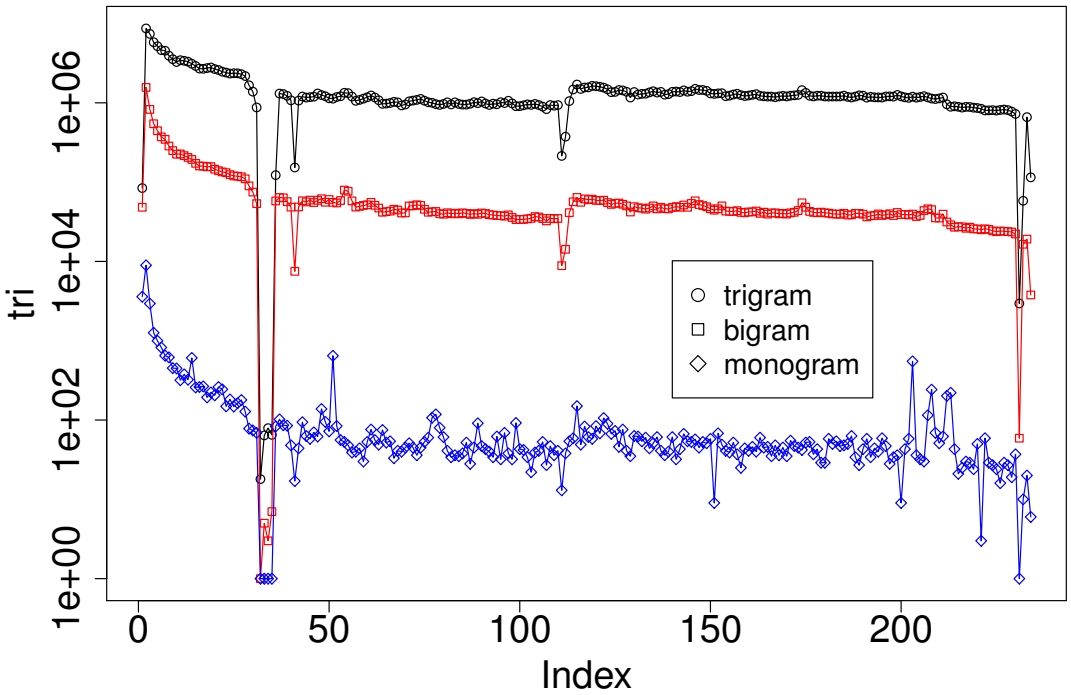

Figure 4.8: Distinct n-grams rate of increase.

constructing words which are longer than two characters, since most of the unknown words and newly created words are words longer than two. In 2010, 52.80% of new words are trigram words [65], followed by quadgram and bigram. Two character words can also be extracted from our data using branch entropy [41].

The histogram in Figure 4.9 shows that the rates of change for bigrams in Weibo is consistent with the sawtooth pattern that Leskovec *et al.* [52] describe for memes, with bigrams peaking at various rates but consistently falling off very slowly. The y-axis is the average number of histograms in each bin over time and the x-axis is binned by the logarithm of the change rate in posts per day. The skew to the right means that bigrams peak at various rates, but fall off very slowly and at a more constant rate.

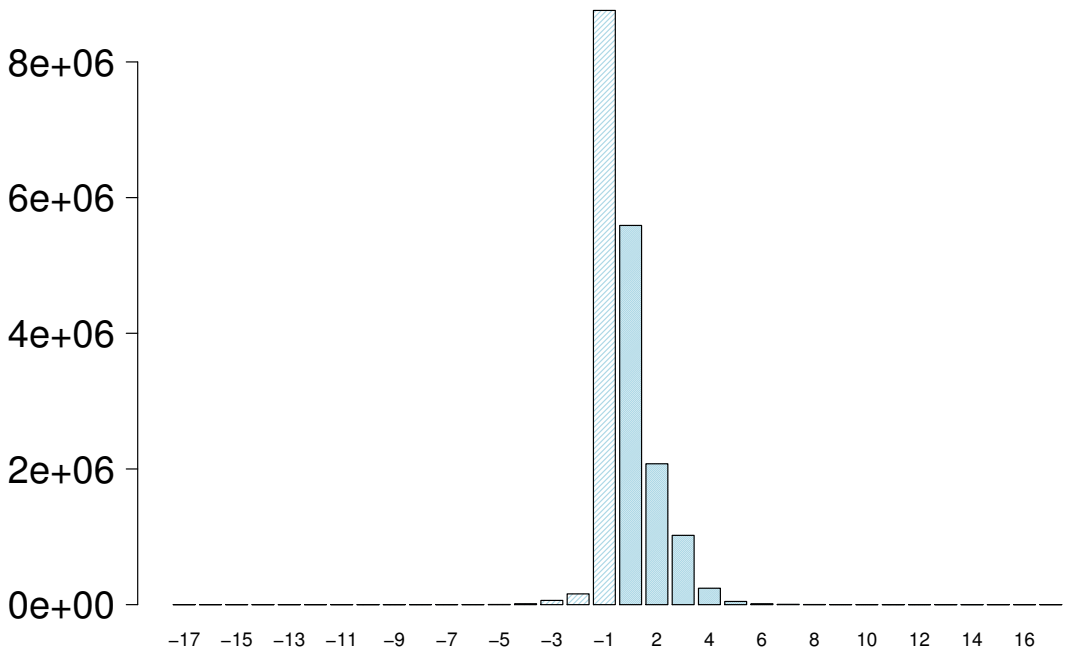

Figure 4.9: Bigram rate change histogram.

4.5.3 From phrases to stories

The volume of text we are capturing is still far too much to process manually. We need automatic methods to classify the posts that we see, particularly the ones which are deleted. For this, we will borrow and extend techniques from the *topic extraction* literature.

Automatic topic extraction is the process of identifying important terms in the text that are representative of the corpus as a whole. Topic extraction was originally proposed by Luhn [58] in 1958. The basic idea is to assign weights to terms and sentences based on their frequency and some other statistical information.

However, when it comes to microblog text, standard language processing tools become inapplicable [94, 54]. Microblogs typically contain short sentences and casual language [23]. Unknown words, such as named entities and neologisms often cause problems with these term-based models. It can be especially challenging to extract topics from Asian languages such as Chinese, Korean, and Japanese, which have no spaces between words. As such, traditional topic extraction techniques based on lexical overlap (use of the same words) become difficult to apply.

We applied the Pointillism approach [71] and TF*IDF to extract hot topics. In the Pointillism model, a corpus is divided into fixed-length grams; words and phrases are reconstructed from grams using external information (*e.g.*, temporal correlations in the appearance of grams), giving the context necessary to manage informal uses of the language such as neologisms. Salton's TF*IDF [70], which stands for "term frequency, inverse document frequency," assigns weights to the terms of a document based on the terms' relative importance to that document compared to the entire corpus. For example, very common words such as the word "the" in English are given a very low weight because the appearance of "the" carries very little information. We next explain how these techniques work together.

For example, the Connector output of the third most popular topic on 4 August 2012 is:

1.头骨进京鸣冤。河北广平县上坡村76岁的农民冯虎，其子在19

skull go Beijing to redress an injustice. The son of a 76 year old farmer Fenghu, from Shangpo village, Guangping city, Hebei province, was ... at 19

2.头骨进京鸣冤。冯出示的头骨赴京鸣...

skull go Beijing to redress an injustice. The skull shown by Feng go Beijing to redress an injustice...

3.头骨进京鸣冤。冯出示的头骨前额有一大窟窿，他...

skull go Beijing to redress an injustice. There is a big hole on the skull shown by Feng,

he...

4.头骨进京鸣冤。冯出示的头骨前额有一个无罪的公民...

skull go Beijing to redress an injustice. There is a innocent citizen on the skull shown by Feng, he...

5.头骨进京鸣冤。冯出示的头骨进...

skull go Beijing to redress an injustice. The skull shown by Feng enter...

6.头骨进京鸣冤。冯出示的头等舱

skull go Beijing to redress an injustice. The first class seat shown by Feng...

7.【華聯社電】上访15年 老父携儿头骨...

Chinese Community report: petition 15 years, old father bring the skull of his son...

Outputs 4 and 6 are incorrectly connected. This is because the same trigrams are shared by different stories that have high TF*IDF scores on the same day. This problem can be solved by examining the cosine similarity of the frequency of occurrence of the first and the last trigram for each result.

Cosine similarity is used to judge whether two trigrams have correlated trends.

$$cos.Sim = \frac{< A_i, B_i >}{\sqrt{\sum_{i=1}^{n} {A_i}^2} \times \sqrt{\sum_{i=1}^{n} {B_i}^2}}$$

where $<,>$ denotes an inner product between two vectors. For details, please refer to Song *et al.* [71].

TF*IDF is a common method to discover the importance of certain words of a document in a corpus. The TF*IDF value in our case is calculated as:

$$f(t, d_{day}) \times log \frac{\text{Total number of posts for the month}}{f(t, d_{month})}$$

Here, $f(t, d)$ means the frequency of the term t in document d. We use trigrams

as t, and documents d are sets of tweets over a certain period of time. d_{day} is the deleted tweets we caught on day day. Similarly, d_{month} is the total deleted tweets we caught in month $month$. IDF is the inverse of the document frequency for deleted posts per month.

First we calculate TF*IDF scores for all trigrams that have more than 20 occurrences in a day. The top 1000 trigrams with the highest TF*IDF score will be fed to our trigram connection algorithm, called "Connector." To connect trigrams back into longer phrases, Connector finds two trigrams which have two overlapping characters. For instance, if there are ABC and BCD, Connector will connect them to become ABCD. Sometimes there is more than one choice for connecting trigrams, *e.g.*, there could also be BCE and BCF. Sometimes trigrams can even form a loop. To solve these problems, we first build directed graphs for the trigrams with a high TF*IDF score. Each node is a trigram, and edges indicate the overlap information between two trigrams. For example, if ABC and BCD can be connected to make ABCD, then there is an edge from 'ABC' to 'BCD'. After all trigrams are selected, we use DFT (Depth First Traversal) to output the nodes. During the DFT we check to see if a node has been traversed already. If so we do not traverse it again. We use the notation '...' to express previously traversed nodes. After the graphs have been traversed, we obtain a set of phrases.

To understand what Chinese net users talk about on social media, we selected 12,891 trigrams which had a rate greater than 100 per day from 23 July 2011 to 22 March 2012 and tried to concatenate them using an 11 day period (5 days before and 5 days after the increasing date). On average, about 56 trigrams[9] had rates higher than 100 everyday.

Interestingly, we found that the Connector sometimes can tell us the story of the event and what caused a large increment in frequency on that day. Here is an

[9] $12,891/229 = 56.3$, where 229 is the number of days of data

example from Connector's output:

100100_20110804_万为开:d: gram=万为开, up1conn=万为开拓团拍电视,
(Wan made a TV program about the first immigrants)
万为开拓团纪念碑被警,
(The statue was ... by the police)
万为开拓团纪念碑被泼上了,
(The statue was splashed ...)
万为开拓团纪念碑被泼红漆,
(The statue was splashed with red paint)
万为开拓团纪念碑被泼红油漆,
(The statue was splashed with red oil paint)
万为开拓团纪念碑被5名男子,
(The statue was ... by 5 men.)
万为开拓团纪念碑被5人砸,
(The statue was defaced by 5 men.)
万为开拓团纪念碑被5人已离,
(The statue was ... by 5 men who have left.)
万为开拓团纪念碑被砸,
(The statue was smashed)
万为开拓团民,
(The first immigrant people)

万为开 is a trigram which has no meaning in Chinese. We caught this particular trigram out of the 323 million trigrams in our database because it appeared 100 times more frequently than average on 4 August 2011. After we fed this trigram into Connector and set the connection time from 5 days before to 5 days afterward, we found the phrase: 万为开拓团拍电视 (Wan made a TV program about the first immigrants). It is still not clear enough to tell us why making a TV program created a trend. However, if you read the candidate results, the whole story becomes clear.

It tells us that 5 men smashed and splashed red oil paint on the statue of "The First Immigrants."

In this event, there are many trigram words, such as 纪念碑 (statues), 开拓团 (immigrants), 红油漆 (red oil paint) and so on. However, the only trigram we caught was, 万为开, a meaningless trigram. In general, these three characters did not appear together before this event. The sudden frequency increase of this trigram from 0 helps our system notice this trigram, which lead us to this event. Other trigrams did not increase in rate as much as 万为开 because of this event. This may be because they already exist and thus it is difficult for them to have a precipitous increase in one day. What my preliminary results regarding trend analysis suggest is that the new emergence of the trend of new words can actually be more conspicuous than known words.

From this example, we can see that by re-thinking the order of operations, a natural language processing task can leverage the frequent creation and use of new words, as shown in Figure 4.10.

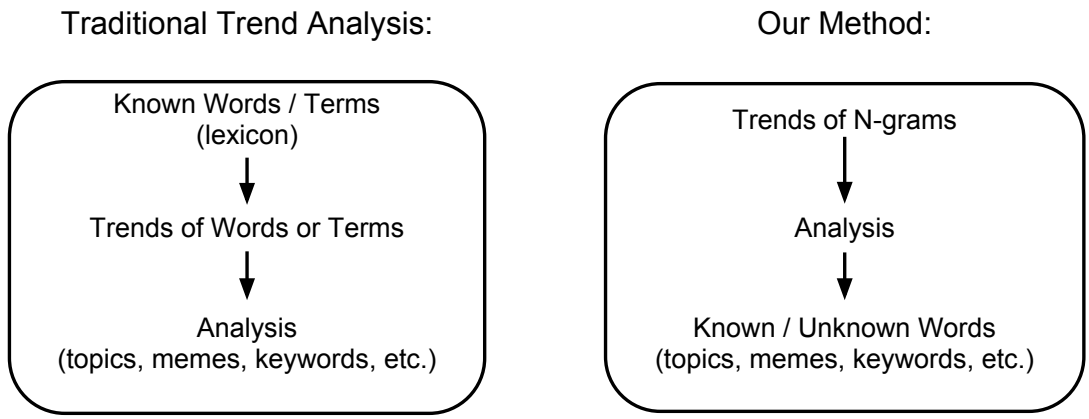

Figure 4.10: Change of order.

Traditional methods need to have a lexicon containing known words or terms to do

analysis. After analyzing the trend, they can infer the topics, memes, hot keywords, *etc.*, for only already-known words and terms. However, in Chinese social media, the unknown words that net users use are an important part of understanding what is being discussed. Our approach is to consider the trends of trigrams first, where we do not need to know the words and the meaning of the words. After analyzing trends, we can get the topics, memes, and keywords for both known words and unknown words and build larger words and phrases from the temporal correlation of trigrams.

4.6 Conclusions

The results in this chapter show that when words and topics do have a meme-like trend they can be reconstructed from only trigrams. For example, for 4-character idioms that appear at least 99 times in one day in our data, unconstrained precision is 0.93. For longer words and phrases collected from Wiktionary (including neologisms), unconstrained precision is 0.87. We consider these results to be very promising, because they suggest that it is feasible for a machine to reconstruct complex idioms, phrases, and neologisms with good precision without any notion of words. Experiments also tell us that a longer period of frequency of trigrams can be used to connect fixed words, while shorter periods which include the trend can be used to connect phrases, memes and, even find the overall topic. Thus the colorful and baroque uses of language that typify social media in challenging languages such as Chinese may in fact be accessible to machines.

The way we can extract hot words without any aid of dictionary and grammar, is not only useful in languages which do not delimit the words by spaces, but also help search query segmentations, dynamic ranking pages, search engine index, *etc.*. for Indo-European languages.

Chapter 5

TREND EXTRACTION

In the last Chapter, it was shown that trigrams with frequency information can be connected into phrases or memes using the Pointillism approach. In this chapter, I examine the potential of the Pointillism approach to extract environment-related trends from microblogging posts. Independent Component Analysis (ICA) is utilized to find the trigrams which have the same independent signal source, *i.e.*, topics. Contrast this with statistical language models, which leverage co-occurrence to classify documents into the topics they have learned. The Pointillism approach extracts trends on the fly, no matter whether those trends have been discussed before or not. This is more challenging because in phrase extraction order information is used to narrow down the candidates. For trend extraction, only the frequencies of the trigrams are considered.

The proposed approach is first compared against a state of the art topic extraction technique, Latent Dirichlet Allocation (LDA), on 9,147 labeled posts with timestamps. The experimental results show that the highest F1 score of the Pointillism approach with Pointillism plus ICA (PICA) is 4% better than that of LDA. Considering that the Pointillism approach with PICA also has a qualitative advan-

tage over LDA, in that topics can be extracted in real-time, the fact that Pointillism with ICA can out-perform LDA shows that topic extraction based on Pointillism and ICA is a promising technique.

Then, to evaluate the relative importance of co-occurrence for PICA vs. LDA, I measure the pointwise mutual information. Finally to show that only a fraction of PICA's performance is due only to clustering, which underlines the importance of the time-varying nature of relationships between trigrams, I do a fuzzy Dunn index score analysis.

5.1 Observation and Analysis

In Chapter 4 I have shown that phrases can be connected back from trigrams which have correlations in their frequency of occurrence. How can we collect trigrams which have similar frequency information into topics?

Microblog post are typified by short sentences and shifting interests that rapidly follow trends. These features create a phenomena that is analogous to bubbles in boiling water. One topic comes up and is soon replaced by another topic. The sentences involved in each topic sometimes have similar meanings, or even almost repeated sentences.

In Figure 5.1, I illustrate how trigrams can be grouped into topics or trends based only on frequency similarity. The nodes represent the 144 top trigrams based on TF*IDF. The edges represent the cosine similarity (> 0.6) of each two nodes.

From Figure 5.1, we can see that some trigrams are tightly clustered together due to a high cosine similarity score between other trigrams in the same cluster. The nodes in between the clusters, from observation, are those trigrams that belong to multiple clusters. Therefore, the frequency of occurrence of trigrams does tell us

trend information.

However, the example in Table 5.1 tells us that if the target phrases are composed of multiple topics during the experiment period, it is not easy to connect them to one single phrase just by the cosine similarity value of the trigrams alone. Recall Table 4.3 from Chapter 4. The existence of two separate trends that share so many trigrams is a major motivation for our use of ICA in this chapter.

To overcome this limitation, we apply independent component analysis (ICA) to separate multiple topics. Originally, ICA was invented to handle the Blind Source Separation (BSS) problem, also known as the cocktail party problem, where the sounds from people talking in a cocktail party are mixed. If we think of the topics as the sources of the signal, the trigrams are the microphones recording mixed signals from the topics, then we can apply ICA to the frequency of the trigrams. The detailed description of ICA can be found in Section 3.3.

In Table 5.1 we expand Table 4.3 and include the other developer conference, 中国移动开发者大会 (China Mobile Developer Conference, held on 3 and 4 November 2011 at Beijing, China). This table shows why ICA is an important part of my trend extraction framework: because these two developer conferences are separate signals that we can extract via ICA because it uses higher-order statistics. Contrast this with LSA, PCA, *etc.*, which use only first- and second-order statistics.

The values in the above table are treated as the observation matrix X and fed into an ICA program. After applying ICA, the two independent signals are extracted in Figure 5.2:

The mixing matrix A ($A \times IC = X$) is:

Table 5.1: Trigram frequency of 谷歌开发者大会, revisited.

	26	**27**	28	29	30	31	01	02	**03**	04	05	06	07	08
谷歌开	1	**68**	5	0	1	1	4	0	2	4	0	0	3	1
歌开发	1	**68**	4	0	1	1	0	0	2	4	0	0	0	0
开发者	49	**127**	43	46	44	65	50	49	**227**	129	38	39	65	63
发者大	15	**56**	5	4	13	10	6	11	**166**	84	14	11	21	14
者大会	15	**56**	5	4	14	9	6	11	**168**	85	14	10	21	13
中国移	35	36	42	39	38	34	45	39	**175**	24	32	34	45	24
国移动	35	35	41	40	36	31	42	39	**181**	23	30	36	40	27
移动开	12	23	29	14	23	13	31	20	**173**	45	32	28	35	40
动开发	13	20	24	10	21	31	20	09	**172**	24	30	24	40	20

$$A_{14,2} = \begin{pmatrix} -1.22 & 17.06 \\ -1.04 & 17.25 \\ 42.52 & 18.15 \\ 37.55 & 10.00 \\ 38.09 & 10.02 \\ 35.84 & 0.05 \\ 37.63 & 0.03 \\ 37.75 & -0.95 \\ 38.66 & -0.51 \end{pmatrix}$$

The element of matrix A is the coefficient of the linear combination of IC to compose X. Thus, from the first column of the matrix A, we know that the last 7 trigrams or microphones are much more closer to the 1st signal source (topic) than the 2nd signal source (topic). On the contrast, the first five trigrams record the 2nd topic more than the 1st topic. This example shows that ICA can group trigrams according to the coefficients of the matrix A.

If source signals (IC) are statistically independent, then ICA can decompose the observed signals (X) into independent sources (IC).

5.2 Procedures

In this section, I explain the steps and procedures for topic extraction.

Step 1: Collecting posts with time sequences.

To examine my hypothesis, I picked a subset of posts from the Weibo public timeline. Pollution related topics are chosen for the following reasons. First of all, it is a hot topic that most users living in China are concerned about, because of the bad condition of the environment in China. Secondly, there are many categories in this broad pollution topic, such as air pollution, nuclear pollution, water pollution, and so on. Those two factors ensure my experimental documents are vivid, with new discussions coming in continuously.

I queried the keyword "污染 (pollution)" in the public timeline database for the period 14 February 2013 to 28 February 2013. The US Air Quality Index (AQI) reading reached 755 on 12 January 2013 and after one month, on 16 February 2013, the neologism "空气末日 (airpocalypse)" was created. Since then, Chinese people have started to change the way they think about their country's toxic air.

To be able to evaluate my method, I use human labelling of the topic for each post. To make this task doable, I chose a 15 day query period, which gives 9,147 posts. The average length of those posts is 81. This is almost twice longer than the average length (42) of the general posts in our public timeline.

Step 2: Count the frequency of occurrence of grams.

The 9,147 posts are grouped into 15 documents by day. The frequency of trigrams appearing in those posts is obtained using the same method described in Section 4.2.

Step 3: Collect top TF*IDF grams.

TF*IDF is a common method to determine the importance of words to a document in a corpus.

In general

$$TF * IDF(t, d, D) = TF(t, d) \times IDF(t, D)$$

$TF(t, d)$ is the term t's frequency in document d, and IDF is:

$$IDF(t, D) = \log \frac{|D|}{|d \in D : t \in d|}$$

.

In which, $|D|$ is the total number of the documents (15, in our case). $|d \in D : t \in d|$ is the number of documents containing the term t.

The documents used in my experiments consist of hundreds of posts with one common keyword: "污染 (pollution)". Compared with regular articles, the ratio of documents to the frequency of the trigrams is too small to have a fair TF*IDF value, because the curve of the logarithm dramatically decreases when x is close to 0 from 1. To weaken the weight of terms frequency, I apply a logarithmically scaled TF' (which is a standard technique for this situation):

$$TF'(t, d) = \log(TF(t, d) + 1)$$

The top 10 trigrams with the highest TF*IDF score are collected for each of the 15 days. In total, there are 144 unique trigrams. We call those trigrams as Top TFIDF Trigrams (TTT).

Step 4: Find trend correlation of grams.

The TTT list is fed in to ICA program with each trigrams' respective frequencies. We use 4 hours as the bin size to aggregate the frequency in 15 days. Thus, the length

of the frequency vector of each trigram is 60 (15 x 4). The detailed algorithms for this step are described in Section 3.3.

The column in A is the coefficient of the IC to compose the observed data. Thus, the mixing matrix A is used to classify the trigrams to the class (IC). The bigger the coefficient in the class (IC), the higher the probability that the trigram belongs to this class.

5.3 Evaluation and Discussion

In this section, the performance of the Pointillism with ICA (PICA) topic extraction approach is compared with LDA.

To measure the precision and recall, the 15 days of microblogging posts were labeled manually by native Chinese speakers. The hierarchical topic diagram in Figure 5.3 was drawn after having read all posts carefully, then posts and trigrams were labeled according to this topic diagram. Each post only belongs to one topic, but trigrams can belong to multiple topics.

There are also 7 spam topics not shown in the above diagram, because they are not closely related to pollution and not widely discussed by Weibo users.

5.3.1 F score

I use precision and recall values to evaluate my methods. Before introducing the precision and recall calculations, we need to define *correctness* with respect to the documents which consist of hundreds of posts no longer than 140 characters.

For a number of reasons, when defining correctness I allow multiple trends to be classified into one class. First, at any given time there are always multiple trends

discussed by Weibo users, some trends may show up and fade at a similar time or in multiple documents. Secondly, I only use 15 days of data to make it possible to have humans label the posts. For LDA, if only using training phrase, it is hard to extract more than 15 topics for only 15 documents. The same is true of PICA.

Both the LDA and PICA algorithms group the top tf-idf trigrams (TTT) into 15 classes. The trigrams classified to the same class may come from different topics, and the trigrams belonging to the same topic may be classified into different groups. Figure 5.4 shows one example of a possible relationship between the classes and the trigrams.

Jaccard index is used to judge the relationships between the labeled trigrams and the classes. A decision can be made about what label a class belongs to by the following calculation:

$$J(C,t) = \frac{|C \cap T|}{|C \cup T|}$$

C is the current trigrams in the class c. T is the set of trigrams that have the same label with as trigram t in class c. In Figure 5.4, class 1 has 11 trigrams, in which trigram 1 through trigram 9 have the same label 'yellow', trigram 8 through trigram 10 have the same label 'blue' and only trigram 11 has the label 'red'. If each kind of label has 8 trigrams in it, and each kind of class has 8 trigrams in it, then the precision of the class 1 is calculated as:

$$J(C_1, T_{1-7}) = \frac{7}{11+1} = 0.58$$

$$J(C_1, T_{8-10}) = \frac{3}{11+5} = 0.19$$

$$J(C_1, T_{11}) = \frac{1}{11+7} = 0.06$$

The label, *e.g.* label 2, with the largest Jaccard index is picked for class 1. $T_{k,c}$ denotes the trigrams which have label k, and class c has label k.

On the other hand, it can be decided to which class a trigram belongs using the same method. For example, to decide which class trigram 11 belongs to:

$$J(C_1, T_{11}) = \frac{1}{11} = 0.09$$
$$J(C_2, T_{11}) = \frac{7}{11} = 0.64$$

The highest score with C_2 means that term 11 belongs to class 2. If one trigram belongs to a class, then all of the trigrams with the same label also belong to that class. c_l denotes the label l of class c. Most of the time, each class only has one label, but one label may belong to several classes.

With those pairs of values, labeled trigrams and the classes are many-to-many mappings.

$$Precision(c) = \frac{\text{The number of trigrams in c, which have the label } c_l}{\text{The number of trigrams in c}}$$

$$Recall(c) = \frac{\text{The number of trigrams in c, which have the label } c_l}{\sum_{c_l} \text{the number of trigrams have label } c_l}$$

5.3.2 Results

The results of my first attempt to apply LDA based on the raw trigrams directly from the corpus were not good. The top 20 trigrams that emerged in each topic were those trigrams with the highest frequency of occurrence in the 15 days, even when I used the "scoring" option.

Those trigrams appear in most every topic, therefore, from the top results, all topics extracted from those 15 documents were the same. To improve the performance of LDA, in the second round of the experiments, I filtered the results by the TTT list I obtained from PICA analysis. After this procedure, the top trigrams in different topics were more diverse. The precision, recall and F1 values are plotted in Figure 5.5.

To further get the best results out of LDA, I treated each post as one document (rather than each day as a document) and fed this to LDA. After filtering with the TTT list, the F score is better than aggregating one day of posts as one document. The precision, recall and F scores are shown in Figure 5.6.

The precision, recall and F values for Pointillism with ICA are shown in Figure 5.7.

The "minimum Valid" parameter on the x-axis in the above-mentioned graphs is a parameter used for internal adjustment. Both the LDA and PICA programs return a matrix as result. Columns are the class, rows are the trigrams. In LDA, the element of the matrix is the frequency of the trigrams appearing in that column class. In ICA, the element of the matrix is the coefficient of the independent component. In this experiment, to match the dimension of the matrix from LDA, all 15 eigenvectors are used as independent signals. For both LDA and PICA, the higher the value of the element, the higher the probability that the trigram belongs to that class.

As training, for each class, we adjust the number of trigrams that can count into that column class, which affects the precision and recall values. Figure 5.5, Figure 5.6, and Figure 5.7 show how precision and recall are affected by the above adjustment.

Both in the aggregated posts as documents version of LDA and the one post per document version of LDA, the number of distinct trigrams having frequency higher

than 40 is 17. In PICA, the number of distinct trigrams having a coefficient larger than 11 is also 17. I set this as the start point. For LDA, I gradually reduce the frequency from 40 to 1 in 14 steps, *i.e.* (40, 37, ... 4, 1). For PICA, the coefficient is reduced from 11 to 2 in 10 steps. *i.e.* (11, 10, ... 3, 2). In each adjustment, the trigrams with values above those numbers are considered as the trigrams belonging to that class. The trigrams lower than that number are not considered.

The highest F score of PICA (0.7802859) is 4% better than that of the aggregated version of LDA (0.7545132), and 2% better than that of the one-post-per-document version of LDA (0.7651325).

5.3.3 Discussion

Topic extraction can also be viewed as trigram classification. PICA performing better than LDA, even in a small scale, suggests that trigrams with similar frequency of occurrence fluctuation can implicitly tell us that they belong to the same class, *i.e.*, topic.

Hypothesis 1: PICA captured the trigrams with similar frequency of occurrence fluctuation better than LDA.

The following explanation of the Dunn index is reproduced from Wikipedia [79]:

> The Dunn index is a metric for evaluating clustering algorithms. The aim is to identify sets of clusters that are compact, with a small variance between members of the cluster, and well separated, where the means of different clusters are sufficiently far apart, as compared to the within cluster variance. For a given assignment of clusters, a higher Dunn index indicates better clustering. One of the drawbacks of using this is the computational cost as the number of clusters and dimensionality of the

data increase.

The formula for the Dunn index is:

$$DI_m = min_{1 \leq j \leq m} \left\{ min_{1 \leq j \leq m, j \neq i} \left\{ \frac{\delta(C_i, C_j)}{max_{1 \leq j \leq m} \Delta_k} \right\} \right\}$$

To test this hypothesis, each trigram is treated as a node. The distance metric used to measure the distance between two trigrams (or nodes) is 1 minus the cosine similarity of these two trigrams. Each topic is treated as a cluster, since the trigrams belonging to the same topic form a cluster.

The distance between the farthest two points (trigrams) inside a cluster is the used to measure the size or diameter of a cluster. The formulation for this is:

Let C_i be a cluster of vectors. Let x and y be any two feature vectors assigned to the same cluster C_i.

$$\Delta_i = max_{x,y \in C_i} d(x, y)$$

For intercluster distance between two clusters I use the minimal distance of all pairs of nodes, one from a cluster and the other from another cluster. The formulation for this is:

$$\delta(C_i, C_j) = min_{x \in C_i; y \in C_j} d(x, y)$$

where $\delta(C_i, C_j)$ is this intercluster distance metric, between clusters C_i and C_j.

Fuzzy Dunn Index

Since a trigram could belong to multiple topics, it is not meaningful to apply traditional Dunn Index to evaluate how well the classification is performed. The

traditional Dunn index uses the minimal intercluster distance of all pairs of clusters as numerator and the largest cluster diameter as the denominator. For a soft clustering (a node could belong to multiply cluster), this will lead to 0 Dunn Index all the time. I modified the Dunn Index to a Fuzzy Dunn Index (FDI) that uses the average intercluster distance and average cluster diameter. The formulation for this is:

$$FDI = \frac{Avg_{i \neq j}(\delta(C_i, C_j))}{Avg(\Delta_i)}$$

Using PICA with coefficient of 7.0 the FDI is 0.5888. For LDA with frequency limit more than 16, the FDI is 0.1549. The higher FDI of PICA shows that my application of PICA obtains better results than LDA in terms of how well the terms are clustered based on their frequency vector similarity.

Hypothesis 2: LDA relies more on trigrams in the same class being captured by co-occurrence with each other than PICA. From Figure 5.3, we know there are usually sub-topics, called *trends*, in larger topics. Trends are the small topics discussed by users, which come and go frequently. Some trends may last longer such as months, but many of them may only last one or two days. Users frequently switch their interests to newly born topics.

The topics extracted by PICA and LDA are marked in Figure 5.8. The number of topics extracted by PICA is smaller than that of LDA. Blue circles marks the topics extracted by LDA, and the topics with the red star on the right are the ones extracted by the PICA approach. This may be because LDA captures the co-occurrence information more than PICA. That is, if trigrams A and B appear together in a subtopic, and trigrams B and C appear together in a different subtopic, then LDA is more likely to put trigrams A, B, and C into one broader topic.

To test this hypothesis, I plot pointwise mutual information (PMI) values for these two methods. PMI is used to measure the degree co-concurrency of two term

x and y.

$$pmi(x;y) = \log \frac{p(x,y)}{p(x)p(y)}$$

In this document, I use normalized pointwise mutual information (npmi) for each pair of the trigram in every class.

$$npmi(x;y) = \frac{pmi(x;y)}{-\log[p(x,y)]}$$

The PMI for each class is defined as:

$$npmi(c_1) = \frac{\sum npmi(\text{Every pair of trigrams in } c_1)}{\frac{n^2-n}{2}}$$

The plot of PMI for LDA and PICA is shown in Figure 5.9.

LDA is better than PICA in terms of the PMI value. This proves our hypothesis that LDA allocates topics depending on the co-occurrence of the keywords, more-so than PICA. The reason the PMI value of PICA is lower is because the classes extracted by PICA do not depend on co-occurrence.

In a nutshell, Table 5.2 shows the results of comparing of PICA (Pointillism with ICA) with LDA. PICA is more suited for real-time trend extraction because it does not rely on co-occurrence, so there is no need for the topic to have appeared in the corpus in the past for PICA to detect an emerging trend. This is a qualitative property that I have not evaluated in this dissertation, but plan to evaluate in future work. My results from this chapter show that PICA can out-perform LDA, the current state of the art. I hypothesized that PICA is less dependent on co-occurrence than LDA, and then used pointwise mutual information (PMI) to show that this is indeed the case. I also hypothesized that ICA is doing more than just clustering, and

Table 5.2: Qualitative comparison of PICA and LDA.

	Real-time	Depends on co-occurrence	Depends on similarity of frequency vector
PICA	Yes	No	Yes
LDA	Maybe	Yes	No

is actually using higher-order statistics to extract meaningful topics. As evidence to support this hypothesis, I used the fuzzy Dunn index to show that clustering alone cannot explain PICA's good results.

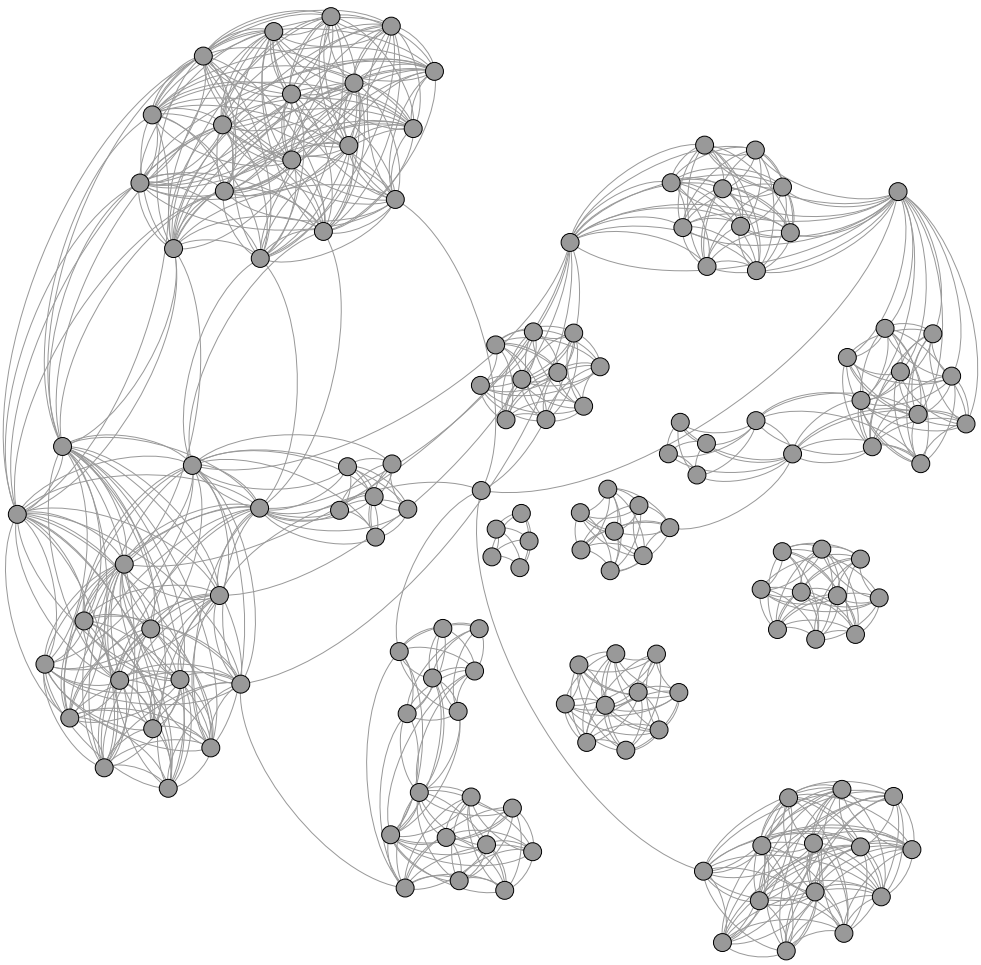

Figure 5.1: Frequency similarity between trigrams.

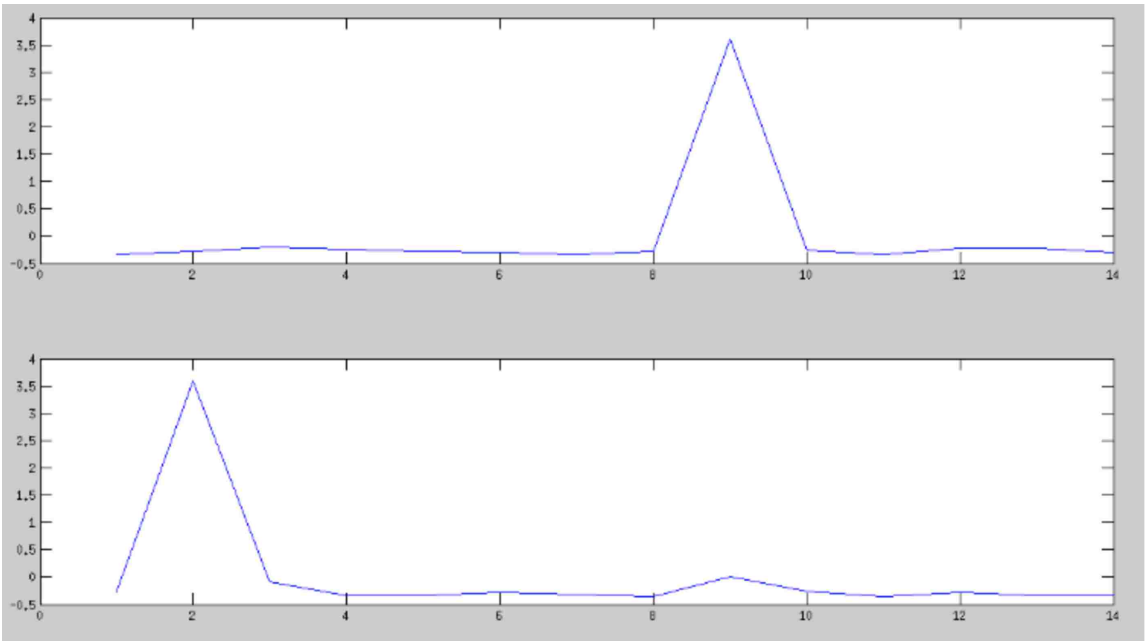

Figure 5.2: Two independent components.

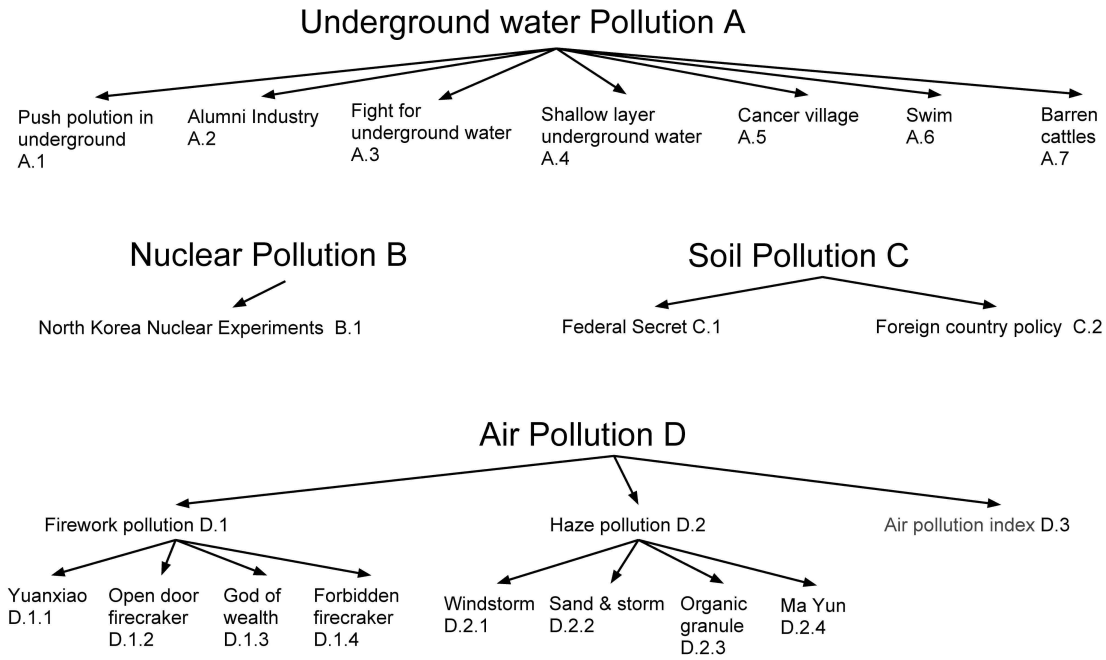

Figure 5.3: Hierarchical topics in 15 days of posts containing the keyword "污染" (pollution).

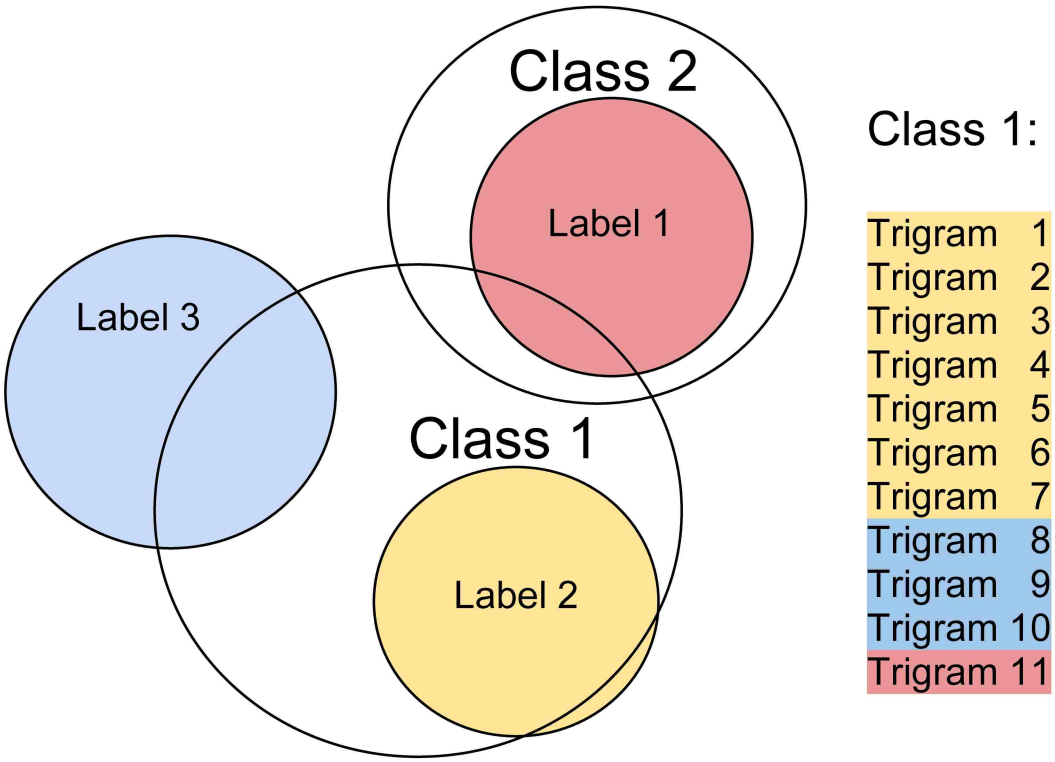

Figure 5.4: The relationship between labeled trigrams and the classes.

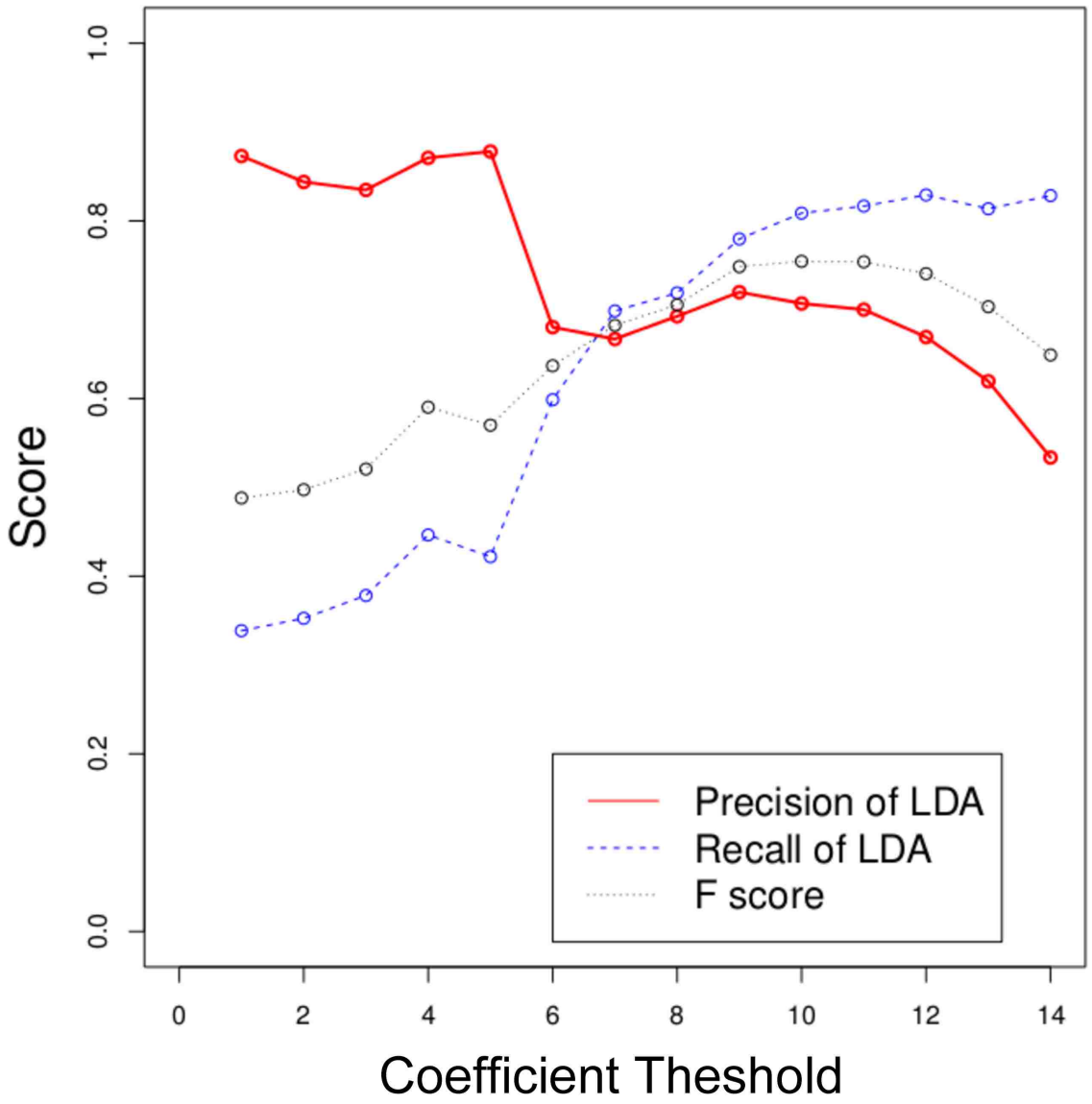

Figure 5.5: The Precision and Recall for LDA by treating aggregated posts in one day as one document.

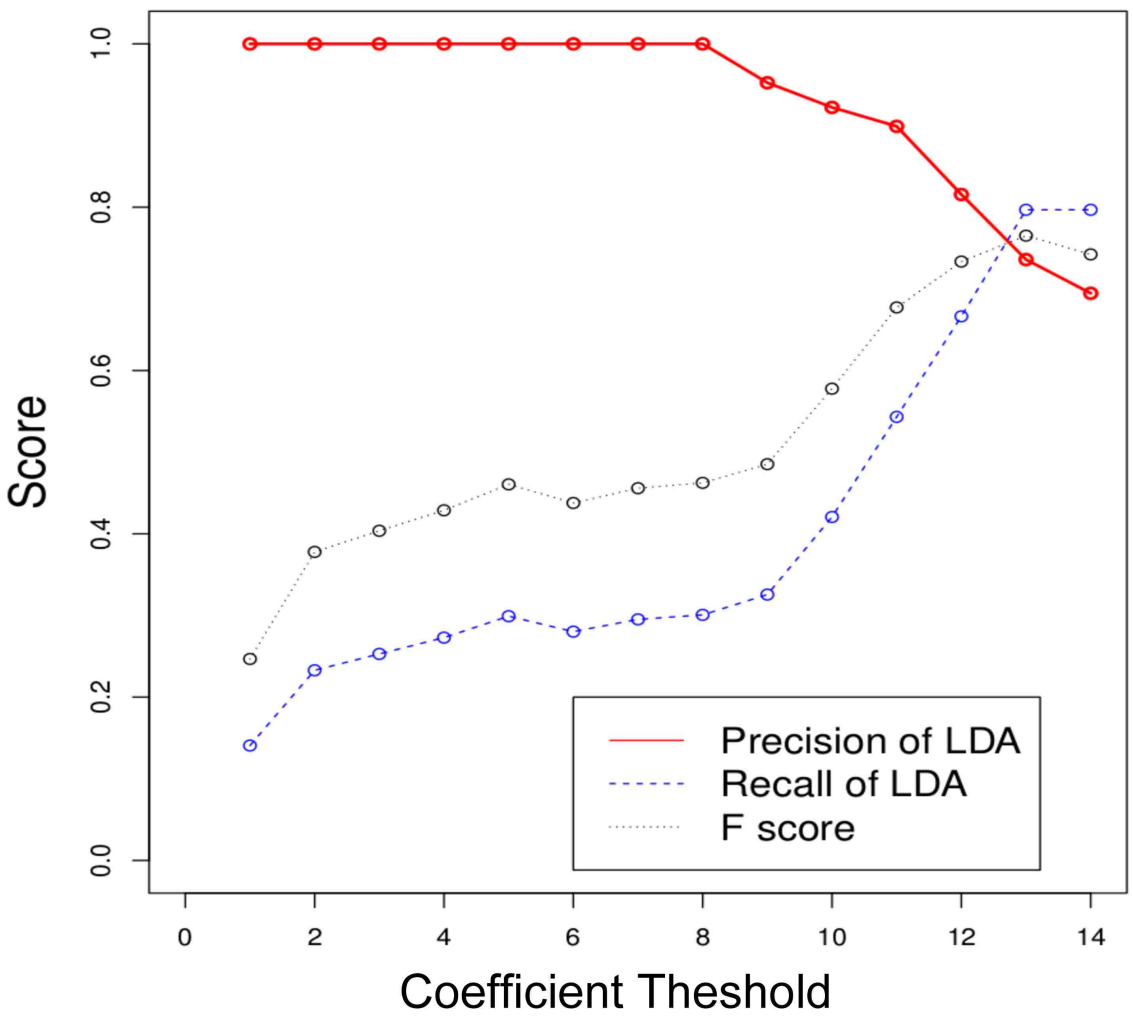

Figure 5.6: The Precision and Recall for LDA by treating each post as one document.

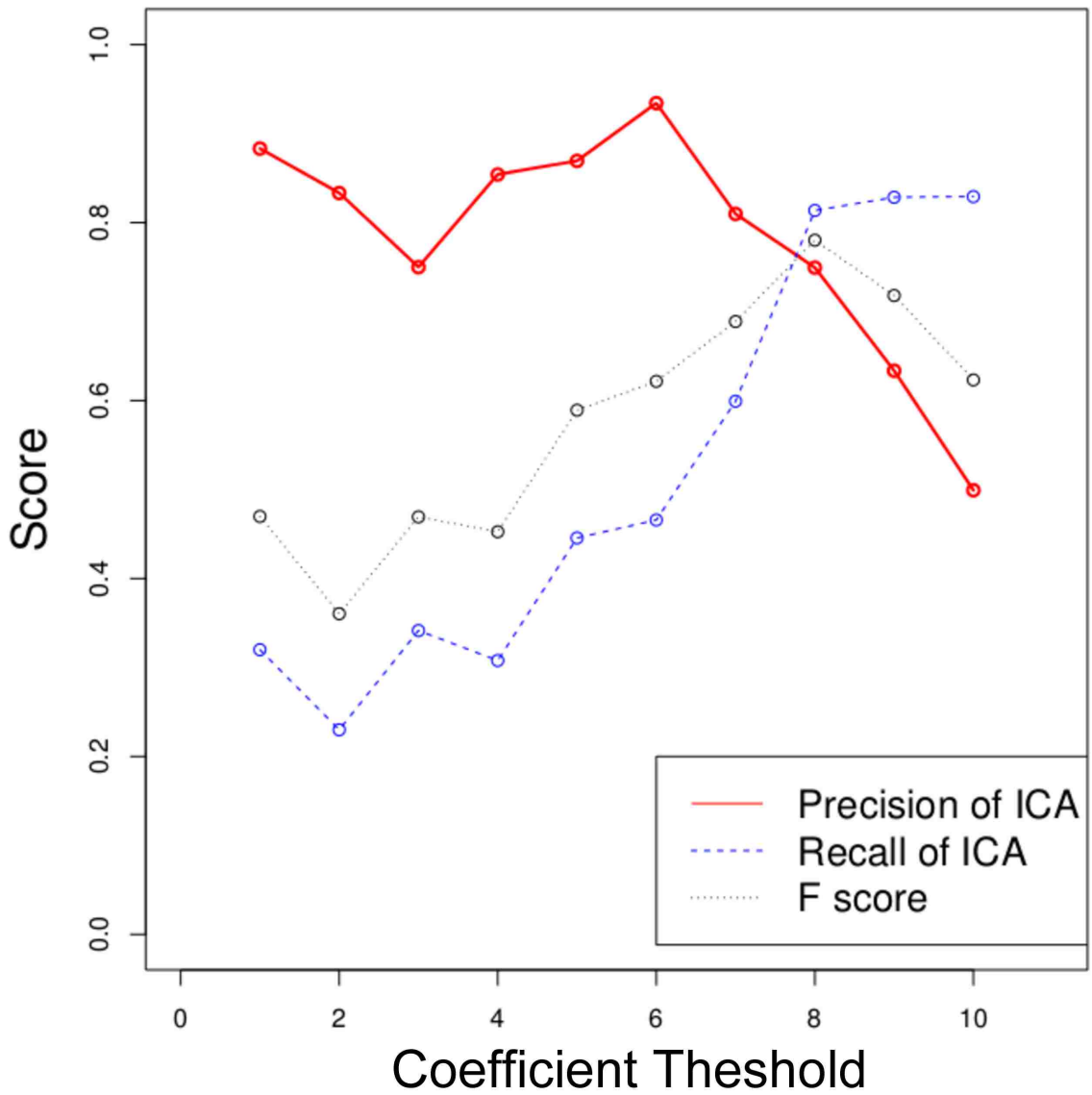

Figure 5.7: The Precision and Recall for PICA.

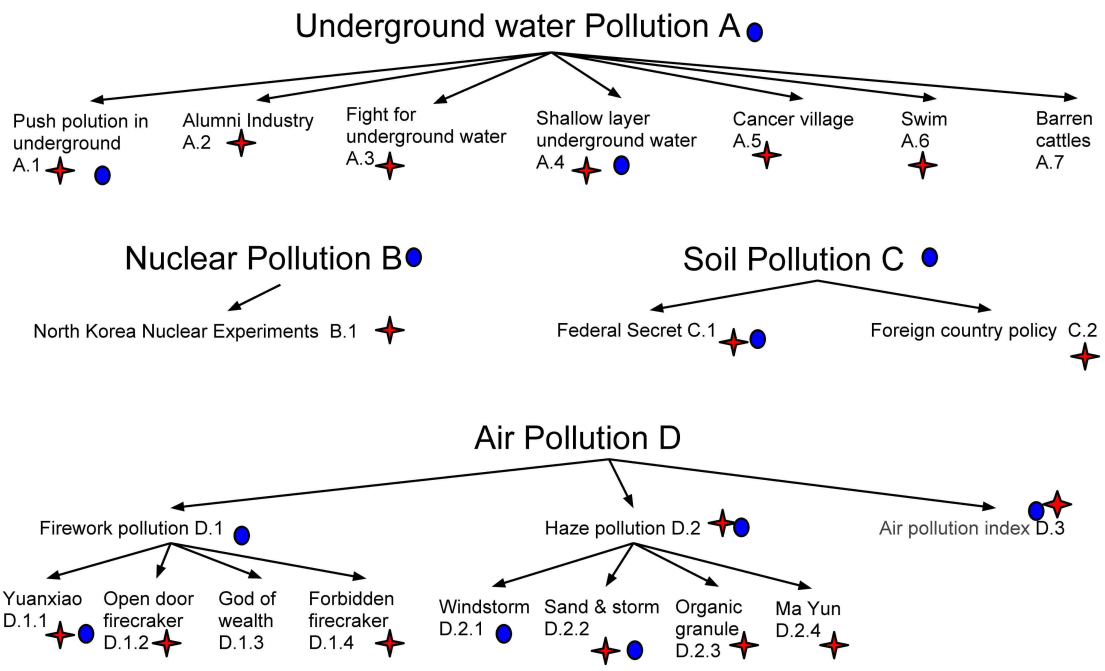

Figure 5.8: The cluster results analysis between LDA and PICA.

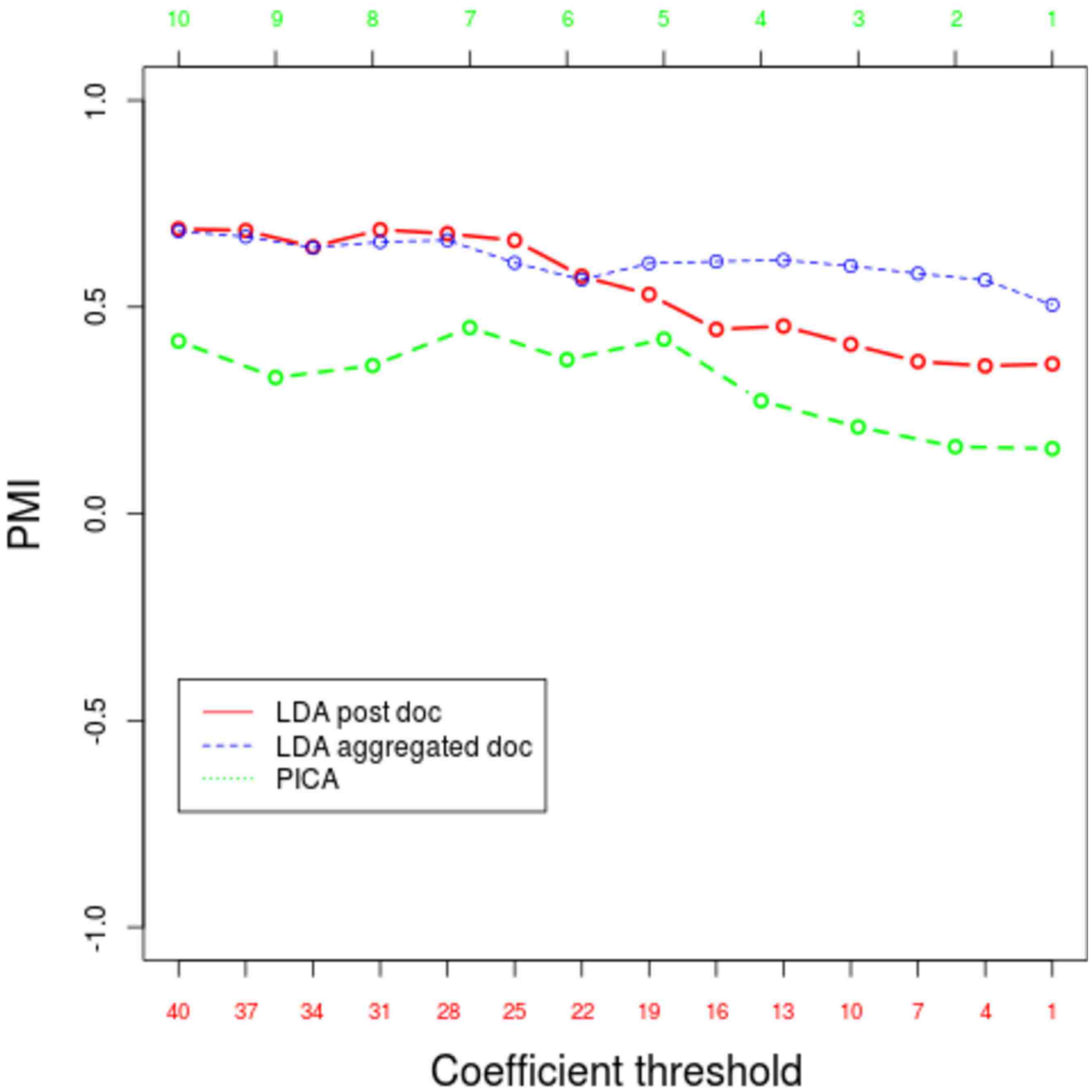

Figure 5.9: The PMI for LDA and PICA.

Chapter 6

CONCLUSIONS

My development and analysis of the Pointillism approach for natural language processing of social media has led to a number of issues and results which will be discussed in this chapter. A brief overview of the results of the proposed approach is given in Section 6.1. Section 6.2 presents my suggestions for potential future work based on what I have learned in this thesis.

6.1 Summary

Social media is very challenging for natural language processing because it challenges the notion of a word. Social media users regularly use words that are not in even the most comprehensive lexicons. These new words can be unknown named entities that have suddenly risen in prominence because of a current event, or they might be neologisms newly created to emphasize meaning or evade keyword filtering. Chinese social media is particularly challenging.

Natural language processing tasks typically start with the basic unit of words, and then from words and their meanings a big picture is constructed about what

the meanings of documents or other larger constructs are in terms of the topics discussed. However, when there are unknown words or phrases, for computers the image which is composed by a corpus has many missing pieces. As for social media, such as microblogging, the unknown words used by users are often more important than the known words. It is difficult for traditional natural language processing to find keywords and extract topics with the presence of unknown words.

For these reasons, we propose a Pointillism approach for Chinese social media natural language processing. Time is an important aspect of the Pointillism approach. Language is viewed as a time sequence of points that represent the grams. The Pointillism approach allows us to look at a corpus in a different dimension and from a different perspective. In the Pointillism approach, trigrams serve as an intermedium to extract trend information, with no need for a lexicon of words, either known or unknown.

My results show that when words and topics do have a meme-like trend, they can be reconstructed from only trigrams. For example, for 4-character idioms that appear at least 99 times in one day in my data, the unconstrained precision was 0.93. For longer words and phrases collected from Wiktionary (including neologisms), the unconstrained precision was 0.87.

I further applied the Pointillism approach to trend extraction with the aid of ICA. The results were compared with state-of-the-art technique LDA. The highest F score of ICA (0.78) was 4% better than that of aggregated version of LDA (0.75), and 2% better than that of one-post-per-document version of LDA (0.76).

The way that we can extract hot words and topics without any aid of dictionary and grammar is not only useful in languages which do not delimit the words by space, but may also help search query segmentations, dynamic page ranking, search engine indexing, *etc.*, for Indo-European languages.

6.2 Future Work

One interesting challenge I would like to adress as an extension of the Pointillism approach is to detect trends using "junk" trigrams.

Other researchers view trigrams as parts of words and phrases and treat all trigrams the same or prefer trigrams with inherent meaning, the Pointillism approach views trigrams as connections between words and phrases. This insight allows the Pointillism approach to catch trending topics with high fidelity since topics are often connections of ideas rather than just prevalence of ideas on a specific day. Since the connections between ideas often appear as meaningless trigrams, we call them "junk" trigrams.

One of the important features of "junk" trigrams is that most of the time the frequency in a time period is zero. Only the specific events can trigger the appearance of "junk" trigrams.

Comparing with meaningful trigrams, which appear constantly in our database, "junk" trigrams have less noise. As an analogy, people used to think that 99% of DNA was junk, because it never appeared directly in the sequences that actually mattered. Then they realized that those 99% were actually quite important, and play a functional role in regulating gene expression. My insight is that for natural language processing of Chinese social media the seemingly "meaningless" trigrams are a critical part of the analysis.

References

[1] F. Alias, X. Sevillano, and J. Socoro. ICA-based hierarchical text classification for multi-domain text-to-speech synthesis. In *Proceedings of the 29th Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, pages 697–700, 2004.

[2] F. Alias, X. Sevillano, and J. Socoro. Reliability in ICA-based text classification. In *Proceedings of the 5th Independent Component Analysis and Blind Signal Separation*, pages 1213–1220. Springer-Verlag, 2004.

[3] S. Asur, B. A. Huberman, G. Szabo, and C. Wang. Trends in social media : Persistence and decay.

[4] D. Bamman, B. O'Connor, and N. Smith. Censorship and deletion practices in Chinese social media. *First Monday*, 17(3-5), March 2012.

[5] M. S. Bartlett. *Face Image Analysis by Unsupervised Learning*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[6] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski. Face recognition by independent component analysis. *Trans. Neur. Netw.*, 13(6):1450–1464, Nov. 2002.

[7] A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.*, 7(6):1129–1159, Nov. 1995.

[8] E. Bingham. Topic identification in dynamical text by extracting minimum complexity time components. In *Proceedings of ICA2001*, pages 546–551, 2001.

[9] N. A. Blei, D.M. and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[10] P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479, Dec. 1992.

[11] V. Calhoun, G. Pearlson, and T. Adali. Independent component analysis applied to fMRI data: A generative model for validating results. *J. VLSI Signal Process. Syst.*, 37(2/3):281–291, June 2004.

[12] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

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

[14] C. Chen, K. Wu, V. Srinivasan, and X. Zhang. Battling the Internet water army: Detection of hidden paid posters.

[15] L. Chenghua, H. Yulan, P. Carlos, and J. Domingue. Feature LDA: A supervised topic model for automatic detection of Web API documentations from the web.

[16] China Internet Network Information Center. *29th Statistical Report on Internet Development in China*, Dec. 2012.

[17] J. R. Cordy, S. Grant, and D. Skillicorn. Automated concept location using independent component analysis. In *Proceedings of the 15th Working Conference on Reverse Engineering*, 2008.

[18] E. Cortez and A. S. da Silva. Unsupervised strategies for information extraction by text segmentation. In *Proceedings of the Fourth SIGMOD PhD Workshop on Innovative Database Research*, IDAR '10, pages 49–54, New York, NY, USA, 2010. ACM.

[19] G. W. Cottrell and J. Metcalfe. Empath: face, emotion, and gender recognition using holons. In *Proceedings of the 1990 conference on Advances in neural information processing systems 3*, NIPS-3, pages 564–571, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[20] O. Déniz, M. Castrillón, and M. Hernández. Face recognition using independent component analysis and support vector machines. *Pattern Recogn. Lett.*, 24(13):2153–2157, Sept. 2003.

[21] R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *Proceedings of the 13th conference on USENIX Security Symposium - Volume 13*, SSYM'04, pages 21–21, Berkeley, CA, USA, 2004. USENIX Association.

[22] B. A. Draper, K. Baek, M. S. Bartlett, and J. R. Beveridge. Recognizing faces with PCA and ICA. *Comput. Vis. Image Underst.*, 91(1-2):115–137, July 2003.

[23] J. Ellen. All about microtext: A working definition and a survey of current microtext research within artificial intelligence and natural language processing. In *3rd International Conference on Agents and Artificial Intelligence (ICAART '11)*, Jan. 2011.

[24] K. W. Fu, C. H. Chan, and M. Chau. Assessing censorship on microblogs in China: Discriminatory keyword analysis and the real-name registration policy. *IEEE Internet Computing*, 17(3):42–50, 2013.

[25] G. W. Furnas, T. K. Landauer, S. Deerwester, S. T. Dumais, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[26] C.-L. Goh. *Unknown Word Identification for Chinese. Morphological Analysis.* PhD thesis, Nara Institute of Science and Technology, 2006.

[27] L. K. Hansen, T. Kolenda, and S. Sigurdsson. *Advances in Independent Component Analysis*, chapter Independent components in text. Springer-Verlag, 2000.

[28] R. Hisano, D. Sornette, T. Mizuno, T. Ohnishi, and T. Watanabe. High quality topic extraction from business news explains abnormal financial market volatility. In *PLoS ONE*, 8(6), 2013.

[29] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence (UAI'99)*, 1999.

[30] T. Honkela and A. Hyvarinen. Linguistic feature extraction using independent component analysis. In *International Joint Conference on Neural Networks*, 2004.

[31] T. Honkela, A. Hyvarinen, and J. Vayrynen. Emergence of linguistic representations by independent component analysis. Technical report, Helsinki University of Technology, 2003.

[32] T. Honkela, J. Vayrynen, and A. Hyvarinen. Independent component analysis of word contexts and comparison with traditional categories. In *the 6th Nordic Signal Processing Symposium (NORSIG)*, 2004.

[33] T. Honkela, J. Vayrynen, and L. Lindqvist. Towards explicit semantic features using independent component analysis. In *Proceedings of the Workshop Semantic Content Acquisition and Representation (SCAR)*, 2007.

[34] D. J. Hopkins and G. King. A method of automated nonparametric content analysis for social science. American Journal of Political Science 54, no. 1: 229–247, 2010.

[35] Y. Hu, A. John, F. Wang, and S. Kambhampati. ET-LDA: Joint topic modeling for aligning events and their Twitter feedback. *CoRR*, abs/1211.3089, 2012.

[36] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Comput.*, 9(7):1483–1492, Oct. 1997.

[37] C. Isbell and P. Viola. Restructuring sparse high dimensional data for effective retrieval. Technical report, Cambridge, MA, USA, 1998.

[38] H. K. J.-Y. Pan M. Hamamoto and C. Faloutsos. A comparative study of feature vector-based topic detection schemes for text streams. In *International Workshop on Challenges in Web Information Retrieval and Integration*, 2005.

[39] X. Jiang, L. Wang, Y. Cao, and Z. Lu. Automatic extraction method of Tibetan new valid words. *Proceedings of the Sixth International Conference on Intelligent Systems and Knowledge Engineering, ISKE2011*, pages 435–444, 2011.

[40] X. Jiang, L. Wang, Y. Cao, and Z. Lu. Automatic recognition of Chinese unknown word for single-character and affix models. *Knowledge Engineering and Management, AISC 123*, page 435–444, 2011.

[41] Z. Jin and K. Tanaka-Ishii. Unsupervised segmentation of Chinese text by use of branching entropy. In *Proceedings of the COLING/ACL on Main conference poster sessions*, COLING-ACL '06, pages 428–435, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[42] X. Jingyue. Low variety in diglossia: the research of users' attitudes toward Zhuyin Wen. Master's thesis, National Chung Cheng University, 2005.

[43] A. Kaban and M. Girolami. Unsupervised topic separation and keyword identification in document collections: a projection approach. Technical report, Dept. of Computing and Information Systems, University of Paisley., 2000.

[44] A. Kempe. Experiments in unsupervised entropy-based corpus segmentation. *Conference on Computational Natural Language Learning (CoNLL-99)*, June 1999.

[45] G. King, J. Pan, and M. E. Roberts. How censorhsip in China allows government criticism but silences collective expression. *American Political Science Review*, 107:1–18, 2013.

[46] T. Kolenda and L. K. Hansen. Dynamical components of chat. Technical report, Department of Mathematical Modeling, Technical University of Denmark, 2000.

[47] T. Kolenda, L. K. Hansen, and J. Larsen. Signal detection using ICA: Application to chat room topic spotting, 2001.

[48] R. S. Kumaran, K. Narayanan, and J. N. Gowdy. Language modeling using independent component analysis for automatic speech recognition. In *Eurpoean Signal Processing Conference*, 2005.

[49] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[50] C.-M. Lee, C.-K. Huang, K.-M. Tang, and K.-H. Chen. Iterative machine-learning Chinese term extraction. In *14th International Conference on Asia-Pacific Digital Libraries*, ICADL 2012, pages 309–312, 2012.

[51] S. Lee, J. Lee, C.-Y. Park, and J.-H. Lee. Blog topic analysis using TF smoothing and LDA. In *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, ICUIMC '13, pages 75:1–75:6, New York, NY, USA, 2013. ACM.

[52] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 497–506, New York, NY, USA, 2009. ACM.

[53] E. Levin and M. Sharifi. Evaluation of utility of LSA for word sense discrimination. In *Proceedings of HLT/NAACL*, 2006.

[54] J. Li, Z. Liu, Y. Fu, and L. She. Chinese hot topic extraction based on Web log. In *Proceedings of the 2009 International Conference on Web Information Systems and Mining (WISM '09)*, pages 103–107, Nov. 2009.

[55] M. Y. Lin, T. H. Chiang, and K. Y. Su. A preliminary study on unknown word problem in Chinese word segmentation. In *ROCLING 6*, pages 119–141, 1993.

[56] Y.-C. Liu and C.-W. Lin. A new method to compose long unknown Chinese keywords. *J. Inf. Sci.*, 38(4):366–382, Aug. 2012.

[57] F. Lu, B. Shen, J. Lin, and H. Zhang. A method of SNS topic models extraction based on self-adaptively LDA modeling. In *Proceedings of the 2013 Third International Conference on Intelligent System Design and Engineering Applications*, ISDEA '13, pages 112–115, Washington, DC, USA, 2013. IEEE Computer Society.

[58] H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165, Apr. 1958.

[59] L.-Y. Z. M. Q. X.-M. Z. H.-X. Ma. A Chinese word segmentation algorithm based on maximum entropy. *Machine Learning and Cybernetics (ICMLC), 2010 International Conference on*, pages 1264 – 1267, July 2010.

[60] W.-Y. Ma and K.-J. Chen. Introduction to CKIP Chinese word segmentation system for the first international Chinese word segmentation bakeoff. In *Proceedings of the second SIGHAN workshop on Chinese language processing - Volume 17*, SIGHAN '03, pages 168–171, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[61] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[62] M. Milanesi, N. Vanello, V. Positano, M. F. Santarelli, D. De Rossi, and L. Landini. An automatic method for separation and identification of biomedical signals from convolutive mixtures by independent component analysis in the frequency domain. In *Proceedings of the 5th WSEAS international conference on Signal, speech and image processing*, SSIP'05, pages 74–79, Stevens Point, Wisconsin, USA, 2005. World Scientific and Engineering Academy and Society (WSEAS).

[63] G. W. O'Brien, M. W. Berry, and S. T. Dumais. Using linear algebra for intelligent information retrieval. *SIAM Review*, 27:573–595, 1995.

[64] P. S. Penev and J. J. Atick. Local feature analysis: A general statistical theory for object representation, 1996.

[65] People.com.cn. 2010 Chinese language situation report published by ministry of education of China. http://edu.people.com.cn/GB/14620075.html.

[66] Q. Pu and G.-W. Yang. Short-text classification based on ICA and LSA. *Advances in Neural Networks*, 3972:265–270, 2006.

[67] Q. L. H. W. P. Qian. AntSeg: an ant approach to disambiguation of Chinese word segmentation. *Information Reuse and Integration, 2006 IEEE International Conference on*, pages 420–425, Sept. 2006.

[68] R. Rapp. Mining text for word senses using independent component analysis. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, 2004.

[69] C. S. Richard W. Sproat. A statistical method for finding word boundaries in Chinese text. *Computer Processing of Chinese and Oriental Languages*, 4:336–351, 1990.

[70] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, Aug. 1988.

[71] P. Song, A. Shu, A. Zhou, D. S. Wallach, and J. R. Crandall. A pointillism approach for natural language processing of social media. In *IEEE International Conference on Natural Language Processing and Knowledge Engineering*, Hefei, China, Sept. 2012.

[72] S. H. Srinivasan. Features for unsupervised document classification. In *6th Conference on Natural Language Learning*, 2002.

[73] X. Sun, H. Wang, and W. Li. Fast online training with frequency-adaptive learning rates for Chinese word segmentation and new word detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 253–262, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[74] J. TeCho, C. Nattee, and T. Theeramunkong. Boosting-based ensemble learning with penalty setting profiles for automatic Thai unknown word recognition. In *Proceedings of the Second international conference on Computational collective intelligence: technologies and applications - Volume Part II*, ICCCI'10, pages 132–141, Berlin, Heidelberg, 2010. Springer-Verlag.

[75] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, Jan. 1991.

[76] X. Wang, M. S. Gerber, and D. E. Brown. Automatic crime prediction using events extracted from Twitter posts. In *Proceedings of the 5th international conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, SBP'12, pages 231–238, Berlin, Heidelberg, 2012. Springer-Verlag.

[77] X. Wang, T. Jiang, and F. Ma. Blog-supported scientific communication: An exploratory analysis based on social hyperlinks in a Chinese blog community. *J. Inf. Sci.*, 36:690–704, December 2010.

[78] R. Wei. The state of new media technology research in China: a review and critique. *Asian Journal of Communication*, 19(1):116–127, 2009.

[79] Wikipedia. Dunn index. Available at http://en.wikipedia.org/wiki/Dunn_index.

[80] Wikipedia. Protests of Wukan. Available at http://en.wikipedia.org/wiki/Protests_of_Wukan.

[81] Wikipedia. Sina Weibo. http://zh.wikipedia.org/wiki/新浪微博.

[82] Wiktionary. Chinese 4-word idioms. http://zh.wiktionary.org/wiki/附录:成语索引.

[83] Wiktionary. Chinese nouns. 4-character idoms list available at http://zh.wiktionary.org/w/index.php?title=Category:汉语名词.

[84] D. Xiong, M. Zhang, and H. Li. Enhancing language models in statistical machine translation with backward n-grams and mutual information triggers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1288–1297, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[85] H. Yanagimoto, T. Yokoi, and S. Omatu. Information recommendation using ICA. *Artificial Life and Robotics*, pages 9:103–106, 2005.

[86] H. Yanagimoto, T. Yokoi, and S. Omatu. Index words selection with ICA. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 4:3348 –3353, 2006.

[87] S. Ye. *Sina Weibo Controls the "Holy Shit Idea of a Generation", Launches New URL Weibo.com*, Apr. 2011. http://techrice.com/2011/04/07/sina-weibo-controls-the-holy-shit-idea-of-a-generation-launches-new-url-weibo-com/.

[88] Y. Ye, Q. Wu, Y. Li, K. P. Chow, L. Hui, and S. M. Yiu. Unknown Chinese word extraction based on variety of overlapping strings. *Inf. Process. Manage.*, 49(2):497–512, Mar. 2013.

[89] L. Yu, S. Asur, and B. A. Huberman. What trends in Chinese social media.

[90] K. Zhai, J. Boyd-Graber, N. Asadi, and M. L. Alkhouja. Mr. LDA: a flexible large scale topic modeling package using variational inference in mapreduce. In *Proceedings of the 21st international conference on World Wide Web*, WWW '12, pages 879–888, New York, NY, USA, 2012. ACM.

[91] D. Zhang, Q. Mei, and C. Zhai. Cross-lingual latent topic extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1128–1137, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[92] S. Zhang, Q. Liu, and L. Wang. A Weibo-oriented method for unknown word extraction. In *SKG*, pages 209–212, 2012.

[93] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, Dec. 2003.

[94] X. Zhao, P. Jin, and L. Yue. A novel POS-based approach to Chinese news topic extraction from Internet. In *Proceedings of the Second International Conference on Future Generation Communication and Networking Symposia (FGCNS '08)*, pages 39–42, Dec. 2008.

[95] V. Zhikov, H. Takamura, and M. Okumura. An efficient algorithm for unsupervised word segmentation with branching entropy and MDL. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 832–842, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[96] M. Zibulevsky and B. A. Pearlmutter. Blind separation of sources with sparse representations in a signal dictionary. In *International Workshop on Independent Component Analysis and Blind Source Separation*, pages 86388–2, 2001.