**Topic Modeling for Natural Language Understanding**

A Thesis

Submitted to the Faculty

of

Drexel University

by

Xiaoli Song

in partial fulfillment of the

requirements for the degree

of

Doctor of Philosophy

December 2016

# Acknowledgments

Thank God for giving me all the help, strength and determination to complete my thesis. I thank my advisor Dr. Xiaohua Hu. His guidance helped to shape and provided much needed focus to my work. I thank my dissertation committee members: Dr. Weimao Ke, Dr. Yuan An, Dr. Erjia Yan and Dr. Li Sheng for their support and insight throughout my research. I would also like to thank my fellow graduate students Zunyan Xiong, Jia Huang, Wanying Ding, Yue Shang, Mengwen Liu, Yuan Ling, Bo Song, and Yizhou Zang for all the help and support they provided. Special thanks to Chunyu Zhao, Yujia Wang, and Weilun Cheng.

# Table of Contents

# List of Tables

# List of Figures

# Abstract

Topic Modeling for Natural Language Understanding
Xiaoli Song
Dr. Xiaohua Hu

This thesis presents new topic modeling methods to reveal underlying language structures. Topic models have seen many successes in natural language understanding field. Despite these successes, the further and deeper exploration of topic modeling in language processing and understanding requires the study of language itself and remains much to be explored.

This thesis is to combine the study of topic modeling with the exploration of language. Two types of language are explored, the normal document texts, and the spoken language texts. The normal document texts include all the written texts, such as the news articles or the research papers. The spoken language text refers to the human speech directed at machines, such as smart phones to obtain a specific service.

The main contributions of this thesis fall into two parts. The first part is the extraction of word/topic relation structure through the modeling of word pairs. Although the word/topic and relation structure has long been recognized as the key for language representation and understanding, few researchers explore the actual relation between words/topics simultaneously with statistical modeling. This thesis introduces a pairwise topic model to examine the relation structure of texts. The pairwise topic model is implemented on different document texts, such as news articles, research papers and medical records to get the word/topic transition and topic evolution.

Another contribution of this thesis is the topic modeling for spoken language. Spoken language refers to the spoken text directed at machine to obtain a specific service. Spoken language understanding involves processing the spoken language and figure out how it maps to actions the user intents. This thesis explores the semantic and syntactic structure of spoken language in detail and provides the insight into the language structure. Also, a new topic modeling method is proposed to incorporate these linguistic features. The model can also be extended to incorporate prior knowledge, resulting in better interpretation and understanding of spoken language.

# Chapter 1: Introduction

Natural language understanding (NLU) is a subtopic of natural language processing (NLP) in artificial intelligence that deals with machine reading comprehension. Natural language processing is an interdisciplinary field combining computer science, artificial intelligence (AI), and computational linguistics. The study of NLP requires both the knowledge about the language and techniques in AI and computer science.

Different from other types of data, language is quite complicated and the study of itself is a research topic. As a field of scientific study of language, linguistic was present before the study of computer science and AI. Entering into the Information Age, researchers begin to study language from a computational perspective and provide computational models of various kinds for linguistic phenomena.

The efforts towards letting machine to understand natural language can be traced back to 'Turing Test' in 1950s. It is proposed by Alan Turing as the criteria for intelligence. The early years of NLP development focuses on machine translation, based on hand-written rules. In the 1980s, machine learning algorithms revolutionized NLP, and the efforts have then been shifted to statistical models.

From linguistics' perspective, natural language has the features as morphology, lexicon, syntax, semantics and pragmatics. From the computer perspective, the specialists try to use grammar, including the syntactic features to infer the semantic meaning. The efforts include POS tagging, chunking, parsing, name entity recognition, text retrieval and text summarization.

Although topic modeling covers a wide variety of languages, due to the complex nature of language, there are still much to be explored. This thesis explores the language characteristics for both long documents and short texts, and incorporates the language phenomena into the topic modeling to obtain the language structures to facilitate language understanding.

The statistical modeling for long documents, such as Latent Dirichlet Allocation and latent semantic analysis tries to capture semantic properties of documents. They model documents as the mixture of word distributions, know as topics. The early topic models assume a document-specific distribution over topics, and then repeatedly select a topic from this distribution and draw a word from the topic selected. Although lots of

models have been proposed to incorporate the relatedness between words/topics, they focus on modeling the order of concepts and terms, instead of relations between them.

This thesis shifts the focus from the concepts and terms to the relations by modeling the word pairs instead of individual words. A relation cannot exist by itself but has to relate two words or topics. Although there are relation of three or more terms, most of the them are binary relations having two slots. We explore the term association and show the shift to relation achieve greater effectiveness and refinement in topic modeling.

Another part of the thesis focuses on the statistical modeling of spoken language. Spoken language is human's speech text directed at machines to obtain a certain service. Topic modeling is not efficient to model the short texts due to the lack of words in texts. Researchers deal with different type of short texts through different methods. Most of the researchers put the short texts together to form long texts. They may also model the topic distribution over the whole corpus instead of over each document. However, these methods can not directly applied to spoken language due to the fact spoken language differs enormously both semantically and syntactically from normal short texts.

In this thesis, we examine the semantic and syntactic structures of spoken language in detail and introduce a statistical modeling way of spoken language processing and understanding.

This thesis improves topic modeling through the exploration of language features. New models are proposed leveraging the language features to reveal new language structures.

## 1.1 Overview

Next chapter gives a detailed introduction to the classical topic modeling methods.

The main work in this thesis includes two parts. The first part consists of three chapters. Chapter 3 raises the problem of pairwise relation network extraction. Chapter 4 and Chapter 5 present the pairwise topic modeling for natural language understanding. Chapter 4 extracts word pairs with information extraction tool to represent the word/topic relation, and examines all possible directional relation between them. In Chapter 5, the word pairs are generalized to include all word pairs with mutual information exceeding a certain threshold to represent the word/topic relations, the resulting word/topic relation network can help explore the topic transition and evolution.

The second part of the thesis consists of three Chapters. Chapter 6 examines the linguistic characteristics

for spoken language and provides insights into the semantic and syntactic structures for spoken language. Also, intent specific sub-language is defined to represent spoken language as a subset of natural language. Chapter 7 shows the method to segment the spoken language into its syntax structures, while Chapter 8 proposes a statistical way of modeling the spoken language.

Chapter 9 summarizes the thesis and highlights the future research work.

# Chapter 2: Classical Topic Modeling Methods

The coming of information age necessitates new tools to help us organize, search and understand the explosive amount of information. Topic modeling offers a powerful toolkit for automatically organizing, understanding, searching, and summarizing large amount of documents. Its ability to organize, understand, search and summarize documents has attracted the attention from researchers for more than a decade. Topic modeling covers a wide variety of methods including the early efforts of latent semantic analysis and Latent Dirichlet Allocation. It is then extended to include syntax, authorship, dynamics, correlation and hierarchies, and can be used for information retrieval, collaborative filtering, document similarity and visualization.

### 2.0.1 Latent Semantic Analysis (LSA)

The Latent Semantic Analysis (LSA) is to use matrix factorization to obtain hidden topics. They are used to represent the documents and terms. Using topics to represent both documents and terms helps calculate the document-document, document-term and term-term similarity.

Originally, as in the tutorial Thomo (2009), all corpus of documents are represented as a matrix, with each column representing one document and each row representing one word. Through singular value decomposition (SVD), each document and each term can be represented by hidden topics.

Formally let A be the $m \times n$ term-document matrix for a collection of documents. Each column of A corresponds to a document. If term $i$ occurs $a$ times in document $j$ then $A[i, j]$ = a. The dimensionality of $A$ is $m$ and $n$, corresponding to the number of words and documents respectively. Assume $B = A^T A$ is the document-document matrix. If documents $i$ and $j$ have $b$ words in common then $B[i, j] = b$.

On the other hand, $C = AA^T$ is the term-term matrix. If terms $i$ and $j$ occur together in $c$ documents then $C[i, j] = c$. Clearly, both $B$ and $C$ are square and symmetric; $B$ is an $m \times m$ matrix, whereas $C$ is an $n \times n$ matrix.

Now, we perform a $SVD$ on $A$ using matrices $B$ and $C$ as

$$A = S\Sigma U^T$$

where $S$ is the matrix of the eigenvectors of $B$, $U$ is the matrix of the eigenvectors of $C$, and $\Sigma$ is the diagonal matrix of the singular values obtained as square roots of the eigenvalues of $B$.

We keep $k$ singular values in $\Sigma$, but keep its dimensionality. The other values are put to zero. We also keep the dimensionality and reduce $S$ and $U^T$ into $S_k$ and $U_k^T$. Matrix $A$ now becomes

$$A_k = S_k \Sigma_k U_k^T.$$

$A_k$ is again an $m \times n$ matrix. The $k$ remaining ingredients of the eigenvectors in $S$ and $U$ correspond to $k$ hidden topics. The terms and documents have now represented by these topics. Namely, the terms are represented by the row vectors of the $m \times k$ matrix $S_k \Sigma_k$, whereas the documents by the column vectors the $k \times n$ matrix $\Sigma_k U_k^T$.

### 2.0.2 PLSA

The Probabilistic Latent Semantic Analysis(PLSA) provides a solid statistical foundation for automated document indexing based on likelihood principle. The PLSA method comes to improve the method of LSA, and solve some other problems that LSA cannot solve. The main advantages for PLSA over LSA is its ability to distinguish polysemy and to cluster the terms into different groups, each group representing one topic.

In this model, each appearance of word $w \in W = \{w_1, ..., w_m\}$ in document $d \in D = \{d1, \ldots, d_n\}$ is associated with unobserved topic variables $z \in Z = \{z_1, \ldots z_k\}$.

Using these definitions, the documents are generated by the following steps:

1) Select a document $d_i$ with probability $P(d_i)$,

2) Pick a latent class $z_k$ with probability $P(z_k|d_i)$,

3) Generate a word $w_j$ with probability $P(w_j|z_k)$.

The joint probability model can be shown as follows:

$$P(d, w) = P(d) \sum z \in ZP(w|z)P(z|d)$$

This model is depended on two assumptions. One is the bag of words assumption that words in document

is independent of each other. Another assumes that the word is independent of document, given topic variable $z$, which means on latent topic $z$, word $w$ is generated independently of the specific document. PLSA has been successful in many real-world applications, including computer vision, and recommender systems. However, it suffers from the overfitting, since the number of parameters grows linearly with the number of documents.

## 2.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation is one of the most classical approaches used today. The appearance of Latent Dirichlet Allocation (LDA) model is to improve the mixture model by capturing the exchangeability of both words and documents.

Topic models are algorithms that can discover the sematic information from a collection of documents. The original purpose of topic modeling is to analyze the collection of documents through topic extraction. Nowadays the topic model has been applied to model data from varied fields, including text mining, searching technology, software technology, computer vision, bio-informatics, finance and even social sciences.

In LDA, a topic is a distribution over a vocabulary. Then, for each document, first randomly choose a topic distribution of this document. Then, for each word in this document, randomly assign a topic from the distribution of the topic we chose before. Finally, the word is chosen under that topic corresponding to the word distribution over that topic. In this model, the latent variables are the proportion of topics and topic assignment for each word. The only observed data is the set of words in document. In statistics, the Bayesian inference is the process to compute the posterior distribution when the prior distributions, a distribution of parameters before data is observed, are given.

LDA is a generative model to mimic the writing process. It models each of $D$ documents as a mixture over $K$ latent topics, each of which describes a multinomial distribution over a $V$ word vocabulary. The generative process for the basic LDA is as follows.

1) Choose a topic $z_{i,j} \sim \text{Mult}(\theta_j)$

2) Choose a word $x_{i,j} \sim \text{Mult}(\Phi z_{i,j})$

Where the parameters of the multinomials for topics in a document $\theta_j$ and words in a topic $\Phi_k$ have Dirichlet priors.

As we can see from the related work, the assumption of topic modeling is too restricted and the language

features are not fully considered during the modeling process. In this thesis, we will combine the language study and model improvement by incorporating the language features into topic modeling.

# Chapter 3: Pairwise Topic Model

*'Relations between ideas have long been viewed as basic to thought, language, comprehension, and memory'.*

Chaffin (1989)

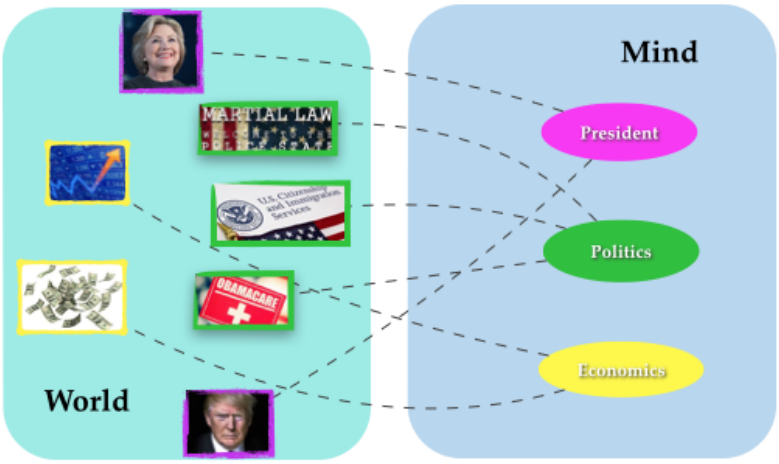



**Figure 3.1:** Concept Network

The world consists of a whole lot of objects with different kinds of features, which are the tangible representation of objects. There are physical world of living things, such as humans and animals. Human may have the features of age, height and nationality. There are intangibly world of knowledge, such as images and languages. The features for images may be the pixels and the features for language can be the words. When we see through the world, we perceive not a mass of features, but objects to which we automatically assign category labels. The categories refer to the sets of objects with similar features. Our perceptual system automatically segments the world into concepts. The concepts are the mental representation of categories. Therefore, when we try to perceive things, we extract from the features of physical representations and translate them into the concepts of mental representations.

While the concepts are the basics of our knowledge, relations between concepts are linking the concepts into the knowledge structures. It has long been recognized that concepts and relations are the foundations of

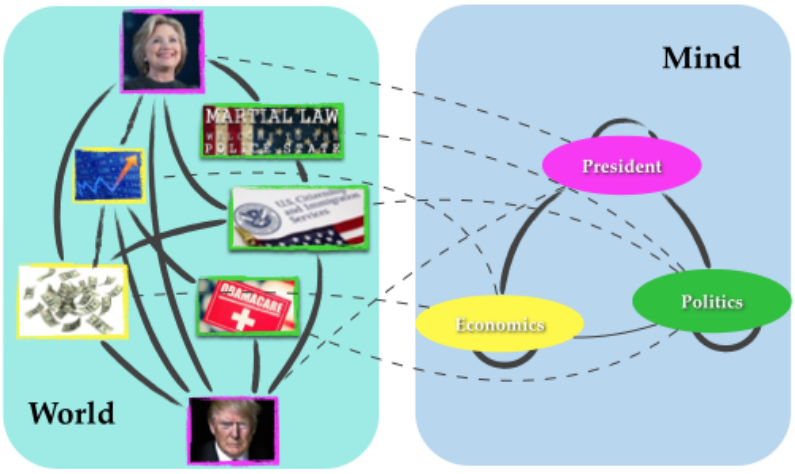



**Figure 3.2:** Concept & Relation Network

our knowledge and thoughts. Our lives and work depend on our understanding of knowledge of concepts and the web of relations.

The concepts and relation structure also presents in language and text. Concepts cannot be defined on their own but only in relation to other concepts, and the semantic relation can reflect the logical structure in the fundamental nature of thought Caplan & Herrmann (1993). Bean & Myaeng Green et al. (2013) noted that semantic relations play a critical role in how we represent knowledge psychologically, linguistically and computationally, and the knowledge representation relies heavily on the examination of internal structure, or in other words, internal relationships between semantic concepts.

Concepts and relations are often expressed in language and text. The words are the features of the language object, while the topics are the human interpretation of the concepts. The generation of language is to present the topic and relation structure in people's mind through the words, and the understanding of language is to restore the structure from the words. Therefore, the understanding of language is a process of translating from the observable words to the topic and relation structure.

Traditional topic modeling focuses on the study of individual terms instead of relations between them. We shift the focus from terms to relations by focusing on the study of word pairs instead of individual words. KhooKhoo & Na (2006) states 'frequently occurring syntagmatic relation between a pair of words can be part of our linguistic knowledge and considered lexical-semantic relations'. As Firth (1957) also puts it, 'you shall

know a word by the company it keeps.'.

Therefore, in this thesis, we examine the pairwise relation through the pairwise relation graph. We formally define the pairwise relation graph as follows.

## 3.1 Pairwise Relation Graph

In this section, we formally define the pairwise relation graph as:

Assume $A$ is a set of $n$ words $\{a_1, a_2, ..., a_n\}$, and $C$ is a set of $m(m < n)$ topics $\{c_1, c_2, ..., c_m\}$. The pairwise relation graph consists of two important components.

a. Nodes. The graph has two types of nodes: words and topics. Each topic is a distribution over words.

b. Node pairs. The graph has two types of node pairs: pairs of words or pairs of topics, with each pair representing the relation between each node pair. Each pair of topic is a distribution over word pairs.

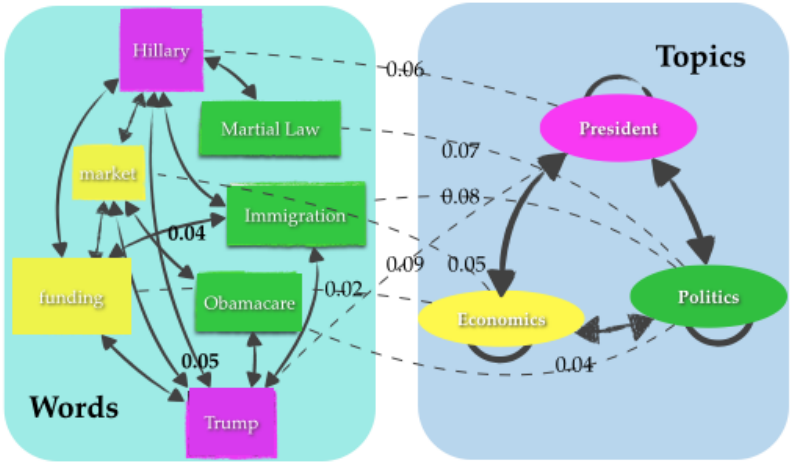



**Figure 3.3:** word/topic network

## 3.2 Problem Formulation

Assume we have a corpus of $n$ documents: $\{d_1, d_2, ...d_n\}$, with vocabulary set $V$. We aim to extract the pairwise relation graph from the collection of documents.

In the following two chapters, pairwise topic modeling is proposed to find the pairwise relation graph. Chapter 4 uses pairwise topic model to model the relation between the word pairs with relations extracted through information extraction tool, while Chapter 5 models the word pairs with the mutual information exceeding a threshold.

Actually, the pairwise topic model can also be extended to other fileds, a extension and modification of the pairwise topic model for image annotation can be found in Appendix.

# Chapter 4: Pairwise Topic Model I

In this chapter, we will model the word pairs with relation extracted using information extraction tool. The pairwise relation graph can be obtained and the perplexity shows the pairwise topic model is more expressive than traditional topic models.

## 4.1 Introduction

Topic modeling is a good way to model language. Two important issues for the topic modeling are to select the text unit to carry one topic and how to model the relationship between the text units and their corresponding topics. For the first issue, different granularities are explored from word, phrase to sentence level. For the second issue, some works assume the semantic dependency between the sequential text units and try to model the dependency between the text unit sequence and their corresponding underlying topics using HMM (Gruber et al., 2007). Others use the syntactic structure information and then model the dependency among the hierarchical syntactic structure accordingly. Obviously, the model of the relationship is effective only when the choice of the text units ensures topical significance and there are explicit topical dependencies between the text units. However, few work view this two related issues as a whole and thus fail to model the relationship between two topics effectively. Therefore, the main problems for the models trying to find the underlying topic structure are twofolds. First, the text units selected may not have topical significance. Second, there is no definite relationship between the text units. In our work, we address the aforementioned two issues by firstly repre'senting the document as the structured data and then modeling the explicit dependency embedded in this specific data structure. Thus, two problems need to be addressed here. The first is what kind of data structure to use. The second is how to model the data structure. For the first part, we will explore the entity pairs within relations as the data structure. The relation is a tuple 'entity1, entity2, relation'. The data structure we explore here is 'entity1, entity2'. Therefore, documents can be represented as a series of entity pairs. Only entities are used here, for they are more likely to convey central ideas and thus have a higher chance of carrying one topic. For example, in the sentence 'As a democrat, Obama does support some type of universal health care',

the relation is support ‘Obama, health care’. The central idea of this sentence is delivered actually by the two entities within it. Thus, the document is viewed as the representation of the closely related key idea pairs. Both the open relation and relation of a specific type are explored here. The relation of a specific type is better structured but can only represent the document from a certain perspective, while the open relation extraction are more flexible to capture more diverse relations, but may not as structured. Thus, the document represented by open relations could capture all aspects of key ideas of a document, while the relation of a specific type may only capture one aspect of all the key ideas.

As for the modeling of the structure, the explicit extraction of the targeted data structure, in this scenario, the entity pair, could to large extent, simplifies the modeling of the relation, since the relation is structured and much easier to model. Using the aforementioned example here, it is much easier to model the relation between ‘Obama’ and ‘health care’ than to model all the words ‘Obama does support some type of universal health care’. Also, the modeling of the relationship between entity pairs makes more sense compared to model the words in the whole sentence, since the entity pair will carry more topical dependency than simply two words appearing together. Here we only focus on the relation between the entities, ignoring the relationship between different relation pairs. This is actually a balance between simplicity and effectiveness. Six models are proposed here. They examine two aspects of the relationship within the entity pair. The first aspect examines whether to treat each entity or each entity pair as a unit. Further, if we treat each entity pair as a unit, whether it is generated from one topic or two. The second aspect examines how the dependency within one relation pair should be modeled. Both the dependency between the entity pair and the dependency between their underlying topics are modeled. As we can see, the modeling of the data structure is dependent on the selection of the structure, and the two are quite correlated.

## 4.2 Related Work

In this part, we would examine how the structure of the document is modeled in previous work. There are mainly three lines of work. In modeling the structure of a document, the first branch is to model the transition of the text units or their underlying topics. Two kinds of transition are modeled. The first is the transition of the observed text units. The second is the transition of the underlying topics of each text unit. The most representative work for modeling the text unit transition is the n-gram topic model. Xuerui Wang treats words

as the topical unit and models the relationship between words. He introduces a binary variable to control whether a consecutive word is dependent on the current topic and previous word, or dependent on the current topic only. Another way is to model the transition of the topic. Hongning Wang treats each sentence as a unit to carry one topic and views the generation of the consecutive underlying topics as a Hidden Markov Chain. Thus, the topic of one sentence depends on the topic of its previous sentence. Gruber Gruber et al. (2007) views each word as the unit to carry one topic and models their dependency of the topics underlying the words by introducing one binary variable to control whether a consecutive word has the same topic as the previous word or is to generate from the document topic distribution as in the LDA model. Another related work is done by Harr Chen et al. (2009). She also views each word as the topic carrier, and introduced a new topic ordering variable into LDA to permute the topic assignment of each word. Thus, the topic assignment for each word in LDA is not only dependent on the document-topic distribution, but also on this topic ordering variable. However, all of these methods have the inherited problem that the text unit to carry one topic may not have topical significance and these topic carriers have no significance in semantic dependency among each other. Therefore, the structure of the document data is not well-defined for the model to perform on, and thus the further modeling of the structure is not appropriate. Another branch to model the document structure is to use the syntactic knowledge. The most relevant model to ours is the syntactic topic model Boyd-Graber & Blei (2009). It is quite similar to our notion of modeling the data structure. In syntactic topic model, Boyd-Graber tried to model the syntactic tree structure resided in the Penne Tree Dataset by adding the dependency between the parent and child of the tree structure. To some extent, this work could be seen as the topic modeling of the structured data, since the document is represented by a series of syntactic trees and the tree structured is modeled by adding the dependencies between the parent and child of the syntactic tree. However, the author fails to fully examine the potential dependencies within one tree structure. Also, this work has two drawbacks from the perspective of modeling of the topical structure. The first is that the syntactic structure may not guarantee the semantic significance, since some of the words of a specific syntactic feature may not carry topical significance. Second, since the syntactic relationship may not have corresponding topical dependency, the topical transition assumption made when modeling the structure may not holds for most of time. Further, from the application perspective, the extraction of the syntactic tree is quite complex and the full exploration

of the tree structure is hard. Further, there are lots of works done on the combination of the LDA model and relation extraction. Yao modeled the relation tuples. In his work, one tuple is represented by a collection of features including the entities themselves. Thus, all these features are generated either by relation type or entity type, which are modeled separately. Although this line of study also focuses on the modeling of the relation, the main purpose of them is to do the relation extraction instead of modeling the document structure, thus may fall short of explicitly modeling entity pair structure.

## 4.3 Document Representation

In this section, we will examine why we choose the entity pairs as the data and how this structure could benefit the document representation and appropriate for topic modeling. We need to know how the topic modeling works before we select the data structure to use. The topic modeling actually takes advantage of the co-occurrence pattern of the text units to find the underlying topics. Intuitively, if two text units co-occur more within one document, they would have a higher chance to be in one topic. Thus, the redundancy of the co-occurred text units across the documents plays an important role to obtain good result. Further, for the structured data of the document, we not only need to model the underlying topics of the text units within the specific structure, but also to model that specific structure of the text units. Thus, we need to take advantage of the redundancy of the co-occurred text units with specific data structure to find the underlying topics.

Therefore, the two standards for us to select the data structure are:

1) the text unit should carry semantic and contextual topical significance and

2) the structure embedded within the text unit should have significance in dependency relation and at the same time simply enough to be captured. The selection of the entities as the text unit satisfies both of the standards.

## 4.4 Pairwise Topic Model

After we select the data structure to represent the document, we need to examine how to model the data structure. To model the entity pair structure, we need to explore the following two questions. The first is whether to treat each entity or each entity pair as a unit. And if each entity pair is treated as a unit, the further question should be whether it comes from one topic or two. The second is to examine how to model the dependency between two entities (if each entity is treated individually) and between the underlying topics

of two entities within an entity pair. Six models are proposed to answer the above two questions. The first three models all view each entity pair as one topic. The first model assumes each entity pair is generated from one topic. The second model assumes that each entity pair comes from two independent topics. The third model also views each entity pair is generated from two topics but the topics have dependency between each other. For the other three models, they all treat each entity as one topic. The fourth model assumes that the two entities are independent, but they come from two dependent underlying topics. The fifth model assumes that the two entities are dependent, but they come from two independent underlying topics. The sixth model assumes that both the entities pairs and topic pairs are dependent. Formally, we assume a corpus consists of D documents and the entity vocabulary size is E. There are K topics embedded in the corpus. For a specific document d, there are N entity pairs. Each model will be described in detail from three perspectives. The generative process of the model (the intuition behind the generative process), the graphical model for the generative process, and the joint probability for the model are introduced step by step in this section. Table 1 lists all the notations used in our models.

**Table 4.1:** Annotations in the generative process for relational topic model

| Notation | Description |
|---|---|
| $D$ | Number of the documents |
| $E$ | Number of the entities |
| $e_p(e_1, e_2)$ | Entity pair |
| $z_p(z_1, z_2)$ | Underlying topic pair for each entity pair |
| $\alpha$ | Dirichlet prior for $\theta_d$ |
| $\alpha_k$ | Dirichlet prior for $\theta_{d,k}$ |
| $\beta_e$ | Dirichlet prior for $\Phi_k$ |
| $\beta_{e_p}$ | Dirichlet prior for $\Phi_{k'}$ |
| $\beta_{e_p{}'}$ | Dirichlet prior for $\Phi_{k_p}$ |
| $\beta_{k,e}$ | Dirichlet prior for $\Phi_{k,e}$ |
| $\theta_d$ | Topic distribution for document $d$ |
| $\theta_{d,k}$ | Topic transition distribution from topic $k$ for document$d$ |
| $\Phi_k$ | Entity distribution for each topic $k$ |
| $\Phi'_k$ | Entity pair distribution for each topic $k$ |
| $\Phi_{k_p}$ | Entity pair distribution for each topic pair$k_p$ |
| $\Phi_{k,e}$ | Word distribution given the topic of the first entity $k$ and the first entity $e$. |

## 4.5 Model Description

After we select the data structure to represent the document, we need to examine how to model the data structure. To model the entity pair structure, we need to explore the following two questions. The first is whether to treat each entity or each entity pair as a unit. And if each entity pair is treated as a unit, the further question should be whether it comes from one topic or two. The second is to examine how to model the dependency between two entities (if each entity is treated individually) and between the underlying topics of two entities within an entity pair. Six models are proposed to answer the above two questions. The first three models all view each entity pair as one topic. The first model assumes each entity pair is generated from one topic. The second model assumes that each entity pair comes from two independent topics. The third model also views each entity pair is generated from two topics but the topics have dependency between each other. For the other three models, they all treat each entity as one topic. The fourth model assumes that the two entities are independent, but they come from two dependent underlying topics. The fifth model assumes that the two entities are dependent, but they come from two independent underlying topics. The sixth model assumes that both the entities pairs and topic pairs are dependent. Formally, we assume a corpus consists of D documents and the entity vocabulary size is E. There are K topics embedded in the corpus. For a specific document d, there are N entity pairs. Each model will be described in detail from three perspectives. The generative process of the model (the intuition behind the generative process), the graphical model for the generative process, and the joint probability for the model are introduced step by step in this section.

### 4.5.1 PTM-1

In PTM-1, each word pair is treated as one unit to be generated from one topic. The generative process is as follow.

1. Draw a topic - entity pair distribution for each topic $k$( $k_1, k_2 = 1, 2, 3...K$):

$$\Phi_k \sim \text{Dirichlet}(\beta_e)$$

2. For each document d (d = 1, 2, . . . D)

(a) Draw a document specific topic distribution:$\theta_d \sim \text{Dirichlet}(\alpha_k)$

(b) For each relation pair, draw the topic of the entity pair from the document-topic distribution $\theta_d$.

$$z_p \sim \text{Dirichlet}(\beta_k)$$

Draw the entity pair from the topic-entity pair distribution

$$e_p \sim \text{Dirichlet}(\beta_k)$$

The joint probability of PTM-1 is as follows.

$$\begin{aligned} & p(E, Z, \theta_d, \theta_{d,k}, \Phi_{z'} | \alpha_k, \beta_{e_p}) \\ = & \prod_{d=1}^{D} \frac{\Gamma(\sum_{k=1}^{K} \alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha-1} \prod_{k=1}^{K} \frac{(\sum_{e=1}^{E})\beta_e)}{\prod_{e=1}^{E}(\beta_e)} {\Phi'_k}^{\beta_{e_p}-1} \prod_{d=1}^{D} \prod_{k=1}^{K} \theta_{d,k}^{n_k} \prod_{k=1}^{K} {\Phi'_k}^{n_{k,e_p}} \\ = & (\frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)})^D \prod_{d=1}^{D} \prod_{k=1}^{K} \Theta_{d,k}^{n_k+\alpha_k-1} (\frac{\Gamma \sum_{e=1}^{E} \beta_e}{\prod_{e=1}^{E} \Gamma(\beta_e)})^K \prod_{k=1}^{K} {\Phi'_k}^{n_k,e_p+\beta_{e_p}-1} \end{aligned} \tag{4.1}$$

### 4.5.2 PTM-2

In PTM-2, each word pair is treated as one unit to be generated from two topics. The generative process is as follow.

1. For each topic pair (k1, k2) (k = 1, 2, 3. . . K),

Draw a topic pair-entity pair distribution for each topic pair

$$\Phi_{k_p} \sim \text{Dirichlet}(\beta_{k_p})$$

2. For each document d (d = 1, 2, . . . D), a. Draw a document specific topic distribution

b. For each relation pair, draw the first and second topics from the document-topic distribution

$$z_1 \sim \text{Categorical}(\theta_d)$$

$$z_2 \sim \text{Categorical}(\theta_d)$$

Draw the entity pair from the topic-entity distribution

$$e_1 \sim \text{Categorical}(\beta_k)$$

$$e_2 \sim \text{Categorical}(\beta_k)$$

The joint probability of PTM-2 is as follows.

$$
\begin{aligned}
& p(E, Z, \theta_d, \Phi_{k_p} | \alpha, \beta'_{e_p}) \\
=& \prod_{d=1}^{D} \frac{\Gamma(\sum_{k=1}^{K} \alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha-1} \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \frac{(\sum_{e_p=1}^{E_p})\beta'_{e_p})}{\prod_{e_p=1}^{E_p}(\beta'_{e_p}-1)} \Phi_{k_p}^{\beta'_{k_p}-1} \prod_{d=1}^{D} \prod_{k_1=k_2=1}^{K} \theta_d^{n_{k_1}+n_{k_2}} \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \Phi_{k_p}^{n_{k_p,e_p}} \\
=& (\frac{\Gamma(\sum_{k=1}^{K} \alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)})^D \prod_{d=1}^{D} \prod_{k_1=k_2=1}^{K} \Theta_d^{n_{k_1}+n_{k_2}+\alpha-1} (\frac{\Gamma \sum_{e_p=1}^{E_p} \beta_{e_p}}{\prod_{e_p=1}^{E_p} \Gamma(\beta_{e_p})})^K \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \Phi_{k_p}^{n_{k_p,e_p}+\beta'_{e_p}-1}
\end{aligned}
\tag{4.2}
$$

### 4.5.3 PTM-3

This model is the similar to R2 but it assumes that there is dependency between the two underlying topics.

Thus, the whole generative process is as follow:

1. For each topic k (k = 1, 2, 3 . . . K),

Draw a topic-entity pair distribution for each topic

$$\Phi_p \sim \text{Dirichlet}(\beta_e)$$

2. For each document d (d = 1, 2, . . . D),

a. Draw a document specific topic distribution

$$\theta_d \sim \text{Dirichlet}(\alpha_k)$$

b. Draw a document specific topic transition distribution for each topic k (k =1,2, . . . ,K).

$$\theta_{k'|k} \sim \text{Dirichlet}(\alpha k'|k)$$

For each entity pair. Draw the first topic from the document-topic distribution

$$z_1 \sim \text{Categorical}(\theta_d)$$

Draw the second topic from the topic transition probability conditioned on the first topic.

$$z_2 \sim \text{Categorical}(\theta_{d,k'|k})$$

Draw the entity pair from the topic-entity pair distribution

$$e_p \sim \text{Categorical}(\Phi_k)$$

The joint probability for PTM-3 is as follow.

$$
\begin{aligned}
& p(E, Z, \theta_d, \theta_{d,k}, \Phi_{k_p} | \alpha, \alpha_k, \beta'_{e_p}) \\
=& \prod_{d=1}^{D} \frac{\Gamma(\sum_{z=1}^{K} \alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)} \prod_{k=1}^{K} \theta_d^{\alpha-1} \prod_{d=1}^{D} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha'_k - 1} \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \frac{(\sum_{e_p=1}^{E_p}) \beta_{e_p})}{\prod_{e_p=1}^{E} (\beta_{e_p} - 1)} \Phi_{k_p}^{\beta'_{e_p} - 1} \\
& \prod_{d=1}^{D} \prod_{k=1}^{K} \theta_{d,k}^{n_{k_1}} \prod_{d=1}^{D} \prod_{k_2=1}^{K} \theta_{d,k}^{n_{k_1,k_2}} \prod_{k_p=1}^{K_p} \Phi_{k_p}^{n_{k_p,e_p}} \\
=& (\frac{\Gamma(\sum_{k=1}^{K} \alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)})^D \prod_{d=1}^{D} \prod_{k_2=1}^{K} \Theta_{d,k}^{n_{k_1} + \alpha - 1} (\frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)})^D \prod_{d=1}^{D} \prod_{k_2=1}^{K} \Theta_{d,k}^{n_{k_1,k_2} + \alpha_k - 1} \\
& (\frac{\Gamma \sum_{e_p=1}^{E_p} \beta_{e_p}}{\prod_{e_p=1}^{E_p} \Gamma(\beta_{e_p})})^K \prod_{k=1}^{K} \prod_{e_p=1}^{E_p} \Phi_{k_p}^{n_{k_p,e_p} + \beta'_{e_p} - 1}
\end{aligned}
\tag{4.3}
$$

### 4.5.4 PTM-4

From this model on, we will view each entity as one unit. Thus, model four will assume that two entities are generated from two dependent topics. The generative process of the first model is as follows:

1. For each topic k (k = 1, 2, 3 . . . K), draw a topic-entity distribution for each topic

$$\Phi_k \sim \text{Dirichlet}(\beta_e).$$

2. For each document d (d = 1, 2, ... D),

a. Draw a document specific topic distribution

$$\theta_d \sim \text{Dirichlet}(\alpha_k)$$

b. Draw a document specific topic transition distribution for each topic k.

$$\theta_{k'|k} \sim \text{Dirichlet}(\alpha_{k^{prime}|k})$$

c. For each relation pair,

Draw the topic of the first entity from the document-topic distribution

$$z_1 \sim \text{Categorical}(\theta_k)$$

Draw the first entity from the topic-entity distribution

$$e_1 \sim \text{Categorical}(\Phi_k)$$

Draw the topic of the second entity from the topic transition probability conditioned on the first topic.

$$z_1 \sim \text{Categorical}(\theta_k)$$

Draw the second entity from the topic-entity distribution

$$e_2 \sim \text{Categorical}(\Phi_k)$$

The joint probability for PTM-4 is as follows.

$$
\begin{aligned}
& p(E, Z, \theta_d, \theta_{d,k}, \Phi_k | \alpha, \alpha_k, \beta_e) \\
=& \prod_{d=1}^{D} \frac{\Gamma(\sum_{z=1}^{K} \alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)} \prod_{k=1}^{K} \theta_d^{\alpha-1} \prod_{d=1}^{D} \frac{\Gamma(\sum_{z=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha_k - 1} \\
& \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \frac{(\sum_{e=1}^{E}) \beta_e)}{\prod_{e=1}^{E} (\beta_e - 1)} \Phi_k^{\beta_e - 1} \prod_{d=1}^{D} \prod_{k=1}^{K} \theta_{d,k}^{n_{k_1}} \prod_{d=1}^{D} \prod_{k_2=1}^{K} \theta_{d,k}^{n_{d,k_1,k_2}} \prod_{k}^{K} \Phi_k^{n_{k,e}} \\
=& (\frac{\Gamma(\sum_{k=1}^{K} \alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)})^D \prod_{d=1}^{D} \prod_{k_1=1}^{K} \Theta_{d,k}^{n_{k_1} + \alpha - 1} (\frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)})^D \prod_{d=1}^{D} \prod_{k_2=1}^{K} \Theta_{d,k}^{n_{k_1,k_2} + \alpha_k - 1} \\
& (\frac{\Gamma \sum_{e=1}^{E} \beta_e}{\prod_{e=1}^{E} \Gamma(\beta_e)})^K \prod_{k=1}^{K} \prod_{e=1}^{E} \Phi_k^{n_{k,e} + \beta_e - 1}
\end{aligned}
\tag{4.4}
$$

### 4.5.5 PTM-5

Different from the fourth model, this model models the dependency between two entities. The formal steps for the second generative model are:

1. For each topic k (k = 1, 2, 3 . . . K), draw a topic-word distribution

$$\Phi_k \sim \text{Dirichlet}(\beta_e)$$

2. For each topic k (k = 1, 2, 3 . . . K) and an entity e (e = 1, 2, 3,. . . ,E), draw an entity distribution

$$\Phi_{k,e} \sim \text{Dirichlet}(\beta_{k,e})$$

3. For each document d (d = 1, 2, . . . D),

a. Draw a document specific topic distribution

$$\theta_d \sim \text{Dirichlet}(\alpha_k)$$

b. For each relation pair,

Draw the topic of the first entity from the document-topic distribution.

$$z_1 \sim \text{Categorical}(\theta_k)$$

Draw the first entity from the topic-entity distribution.

$$e_1 \sim \text{Categorical}(\Phi_k)$$

Draw the topic of the second entity from the document-topic distribution.

$$e_1 \sim \text{Categorical}(\beta_e)$$

Given the first entity and second entity and its topic, draw the second entity from (topic, entity) entity distribu-

tion. $$e_2 \sim \text{Categorical}(\beta_{k,e})$$

The joint probability for the fifth model is:

$$\begin{aligned}
&p(E, Z, \theta_d, \Phi_k, \Phi_{ke}|\alpha, \beta_e, \beta_{ke}) \\
=&\prod_{d=1}^{D} \frac{\Gamma(\sum_{k=1}^{K}\alpha)}{\prod_{k=1}^{K}\Gamma(\alpha)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha-1} \prod_{k=1}^{K} \frac{(\sum_{e=1}^{E})\beta_e)}{\prod_{e=1}^{E}(\beta_e-1)} \Phi_k^{\beta_e-1} \prod_{k=1}^{K}\prod_{e=1}^{E} \frac{(\sum_{e=1}^{E})\beta_{ke})}{\prod_{e=1}^{E}(\beta_{ke}-1)} \Phi_{ke}^{\beta_{ke}-1} \\
&\prod_{d=1}^{D} \prod_{k_1=k_2=1}^{K} \theta_d^{n_{k_1}+n_{k_2}} \prod_{k_1=1}^{K} \Phi_k^{n_{k_1,e_1}} \prod_{k_2=1}^{K}\prod_{e_1=1}^{E} \Phi_{k,e}^{n_{k_2,e_p}} \\
=&(\frac{\Gamma(\sum_{k=1}^{K}\alpha)}{\prod_{k=1}^{K}\Gamma(\alpha)})^D \prod_{d=1}^{D} \prod_{k=k_1=k_2=1}^{K} \Theta_d^{n_{k_1}+n_{k_2}+\alpha-1} (\frac{\Gamma\sum_{e=1}^{E}\beta_e}{\prod_{e=1}^{E}\Gamma(\beta_e)})^K \prod_{k_1=1}^{K} \Phi_k^{n_{k_1,e_1}+\beta_e-1} \\
&(\frac{\Gamma\sum_{e=1}^{E}\beta_{ke}}{\prod_{e=1}^{E}\Gamma(\beta_{ke})})^{KE} \prod_{k_2=1}^{K}\prod_{e_2=1}^{E} \Phi_{ke}^{n_{k_2,e_p}+\beta_{ke}-1}
\end{aligned} \tag{4.5}$$

### 4.5.6 PTM-6

The sixth model inserts both the dependency between the entities and dependency between the underlying topics. The generative process is as follow:

1. For each topic k (k = 1, 2, 3 . . . K), draw a topic-word distribution for each topic

$$\Phi_k \sim \text{Dirichlet}(\beta_e).$$

2. For each topic k (k = 1, 2, 3, ..., K) and an entity e (e = 1, 2, 3, . . . ,E)

Draw an entity distribution

$$\Phi_{k,e} \sim \text{Dirichlet}(\beta_{k,e})$$

3. For each document d (d = 1, 2, . . . D),

a. Draw a document specific topic distribution

$$\theta \sim \text{Dirichlet}(\alpha_k)$$

b. Draw a specific topic transition distribution from topic k.

$$\theta_k \sim \text{Dirichlet}(\alpha_k)$$

c. For each relation pair, draw the topic of the first entity from the document-topic distribution

$$z_1 \sim \text{Categorical}(\theta_k)$$

Draw the first entity from the topic-entity distribution

$$e_1 \sim \text{Categorical}(\beta_k)$$

Draw the topic of the second entity from the topic transition probability conditioned on the first topic.

$$z_2 \sim \text{Categorical}(\alpha_{k'|k})$$

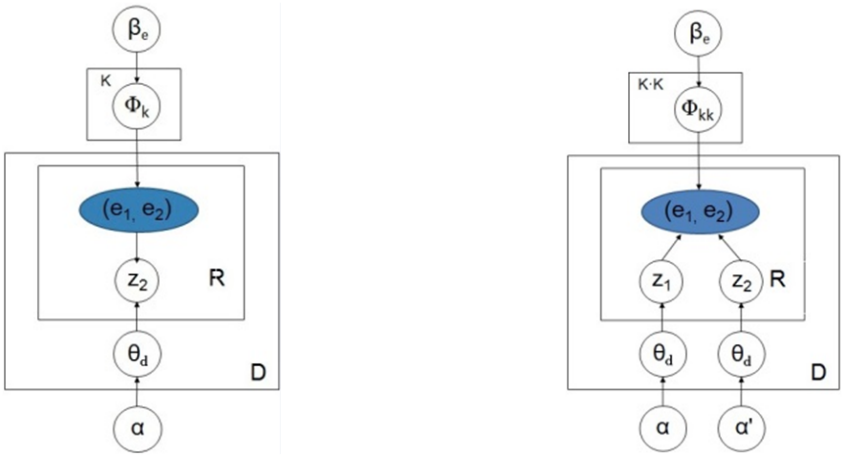
Draw the second entity from (topic, entity)-entity distribution.

$$e_2 \sim \text{Categorical}(\beta_{k,e})$$

The joint probability of the sixth model is:

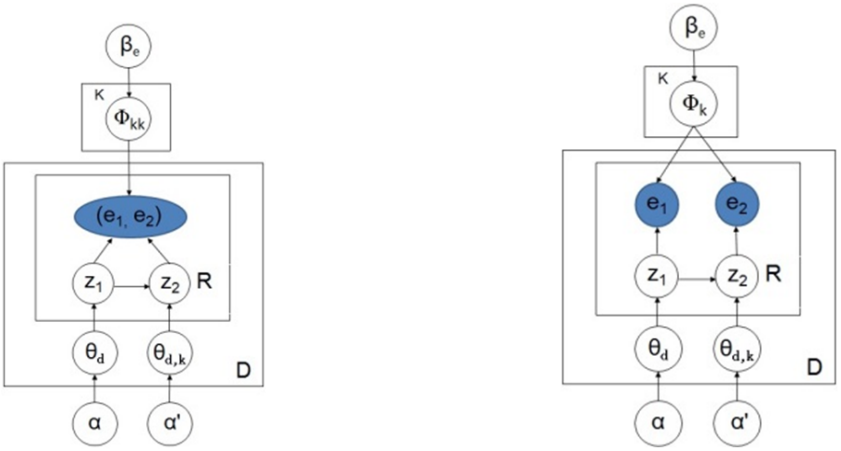
$$\begin{aligned}
&p(E, Z, \theta_d, \theta_{d,k}, \Phi_k, \Phi_{ke} | \alpha, \alpha_k, \beta_e, \beta_{ke}) \\
=&\prod_{d=1}^{D} \frac{\Gamma(\sum_{z=1}^{K} \alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)} \prod_{k=1}^{K} \theta_d^{\alpha-1} \prod_{d=1}^{D} \frac{\Gamma(\sum_{z=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha_k - 1} \prod_{k=1}^{K} \frac{(\sum_{e=1}^{E})\beta_e)}{\prod_{e=1}^{E}(\beta_e - 1)} \Phi_k^{\beta_e - 1} \\
&\prod_{k=1}^{K} \prod_{e=1}^{E} \frac{(\sum_{e=1}^{E})\beta_e)}{\prod_{e=1}^{E}(\beta_e - 1)} \Phi_{ke}^{\beta_{ke} - 1} \prod_{d=1}^{D} \prod_{k_1=1}^{K} \theta_d^{n_{k_1}} \prod_{d=1}^{D} \prod_{k_2=1}^{K} \theta_d^{n_{k_1,k_2}} \prod_{k_1=1}^{K} \Phi_k^{n_{k_1,e_1}} \prod_{k_2=1}^{K} \prod_{e_1=1}^{E} \Phi_{k,e}^{n_{k_2,e_p}} \\
=&(\frac{\Gamma(\sum_{k=1}^{K} \alpha)}{\prod_{k=1}^{K} \Gamma(\alpha)})^D \prod_{d=1}^{D} \prod_{k=k_1=1}^{K} \Theta_{d,k}^{n_{k_1} + \alpha - 1} (\frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)})^D \prod_{d=1}^{D} \prod_{k=k_2=1}^{K} \Theta_{d,k}^{n_{k_1,k_2} + \alpha_k - 1} \\
&(\frac{\Gamma \sum_{e=1}^{E} \beta_e}{\prod_{e=1}^{E} \Gamma(\beta_e)})^K \prod_{k_1=1}^{K} \Phi_k^{n_{k_1,e_1} + \beta_e - 1} (\frac{\Gamma \sum_{e=1}^{E} \beta_{ke}}{\prod_{e=1}^{E} \Gamma(\beta_{ke})})^{KE} \prod_{k_2=1}^{K} \prod_{e_2=1}^{E} \Phi_{ke}^{n_{k_2,e_p} + \beta_{ke} - 1}
\end{aligned} \tag{4.6}$$

The graphical model of the generative process is shown in Figure 4.1-4.3.



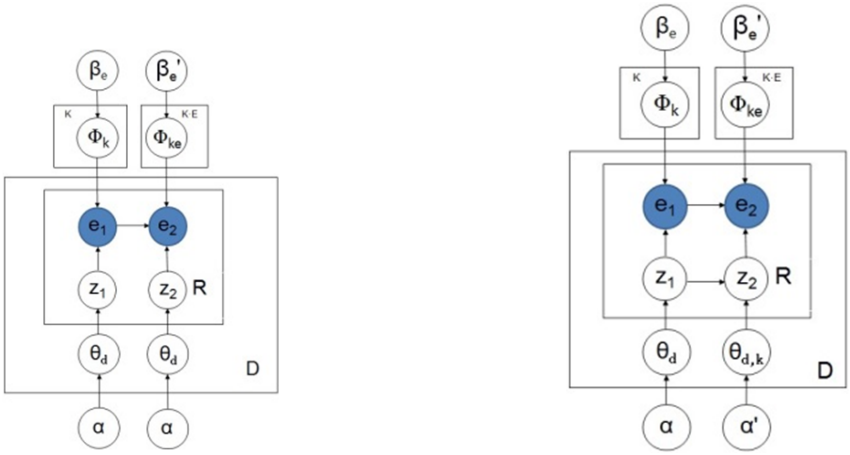

PTM-1 PTM-2

**Figure 4.1:** Graphical Model for PTM-1 and PTM-2




PTM-3 PTM-4

**Figure 4.2:** Graphical Model for PTM-3 and PTM-4




PTM-5 PTM-6

**Figure 4.3:** Graphical Model for PTM-5 and PTM-6

## 4.6 Inference

We use the Gibbs Sampling to perform the parameter estimation and model inference. Because of the independency between the relations, two topics within one relation are sampled simultaneously. Table 4.2 lists all the notations used in Gibbs Sampling. Given the assignment of all the other hidden topic pairs, we use the following formula to sample the topic pairs for the model PTM-1 through model PTM-6.

**Table 4.2:** Annotations for the inference of relational topic model

| Notation | Description |
|---|---|
| $n_{d,z_i}^{\neg i}$ | Number of the entity pairs assigned to topic $z_i$ in document d except for the current entity pair |
| $n_{d,(z_{i_1} z_{i_2})}^{\neg i}$ | Number of the second entity assigned to topic $z_{e_2}$ given the topic of the first entity is $z_{e_1}$ except for the current entity pair |
| $n_{z_i,e_{i_p}}^{\neg i}$ | Number of entity pair $e_{i_p}$ assigned to topic $z$ except for the current entity pair. |
| $n_{z_{i_p},e_{i_p}}^{\neg i}$ | Number of entity pair $e_{i_p}$ assigned to topic pair $z_p$ except for the current entity pair |
| $n_{(z_{i_1},e_{i_1}),e_{i_2}}^{\neg i}$ | Number of the second entity $e_{i_2}$ assigned to $z_{i_2}$, given the first entity is $e_{i_1}$ . |

PTM-1:

$$\begin{aligned} & p(z_i|E, Z_{\neg i}, \alpha, \beta_{e_p}) \\ \propto & \frac{\alpha + n_{d,z_i}}{\sum_{k=1}^{K} \alpha + \sum_{k=1}^{K} n_{d,k}^{\neg i}} \frac{\beta_{e_p} + n_{z_i,e_{i_p}}^{\neg i}}{\sum_{e_p=1}^{E_p} \beta_{e_p} + \sum_{e_p=1}^{E_p} n_{z_i,e_p}^{\neg i}} \end{aligned} \tag{4.7}$$

PTM-2

$$\begin{aligned} & p(z_{i_1}, z_{i_2}|E, Z_{\neg i}, \alpha, \beta'_{e_p}) \\ \propto & \frac{\alpha + n_{d,z_{i_1}=z_{i_2}}^{\neg i}}{\sum_{k=1}^{K} \alpha + \sum_{k=1}^{K} n_{d,k}^{\neg i}} \frac{\beta'_{e_p} + n_{z_{i_p},e_{i_p}}^{\neg i}}{\sum_{e_p=1}^{E_p} \beta'_{e_p} + \sum_{e_p=1}^{E_p} n_{z_{i_p},e_p}^{\neg i}} \end{aligned} \tag{4.8}$$

PTM-3

$$\begin{aligned} & p(z_{i_1}, z_{i_2}|E, Z_{\neg i}, \alpha, \alpha_k, \beta_{e'_p}) \\ \propto & \frac{\alpha + n_{d,z_{i_1}}^{\neg i}}{\sum_{k=1}^{K} \alpha + \sum_{k=1}^{K} n_{d,k}^{\neg i}} \frac{\alpha_k + n_{d,(z_{i_1},z_{i_2})}^{\neg i}}{\sum_{k=1}^{K} \alpha_k + \sum_{k=1}^{K} n_{d,(z_{i_1},k)}^{\neg i}} \frac{\beta'_{e_p} + n_{z_{i_p},e_{i_p}}^{\neg i}}{\sum_{e_p=1}^{E_p} \beta'_{e_p} + \sum_{e_p=1}^{E_p} n_{z_{i_p},e_p}^{\neg i}} \end{aligned} \tag{4.9}$$

PTM-4

$$\begin{aligned} & p(z_{i_1}, z_{i_2}|E, Z_{\neg i}, \alpha, \beta, \beta_{e'_p}) \\ \propto & \frac{\alpha + n_{d,z_{i_1}}^{\neg i}}{\sum_{k=1}^{K} \alpha + \sum_{k=1}^{K} n_{d,k}^{\neg i}} \frac{\alpha_k + n_{d,(z_{i_1},z_{i_2})}^{\neg i}}{\sum_{k=1}^{K} \alpha_k + \sum_{k=1}^{K} n_{d,(z_{i_1},k)}^{\neg i}} \frac{\beta_e + n_{(z_{i_1},e_{i_1})=(z_{i_2},e_{i_2})}^{\neg i}}{\sum_{e=1}^{E} \beta_e + \sum_{e=1}^{E} (n_{z_{i_1}=z_{i_2},e}^{\neg i})} \end{aligned} \tag{4.10}$$

PTM-5

$$
\begin{aligned}
&p(z_{i_1}, z_{i_2}|E, Z_{\neg i}, \alpha, \beta, \beta_{e'_p}) \\
\propto &\frac{\alpha + n^{\neg i}_{d,z_{i_1}=z_{i_2}}}{\sum_{k=1}^{K}\alpha + \sum_{k=1}^{K} n^{\neg i}_{d,k}} \frac{\beta_e + n_{z_{i_1},e_{i_1}}}{\sum_{e=1}^{E}\beta_e + \sum_{e=1}^{E}(n_{z_{i_1},e_{i_1}})} \frac{\beta_{ke} + n_{z_{i_2},e_{i_1}}}{\sum_{e=1}^{E}\beta_{ke} + \sum_{e=1}^{E}(n_{z_{i_2},e})}
\end{aligned}
\tag{4.11}
$$

PTM-6

$$
\begin{aligned}
&p(z_{i_1}, z_{i_2}|E, Z_{\neg i}, \alpha, \beta, \beta_{e'_p}) \\
\propto &\frac{\alpha + n^{\neg i}_{d,z_{i_1}}}{\sum_{k=1}^{K}\alpha + \sum_{k=1}^{K} n^{\neg i}_{d,k}} \frac{\alpha_k + n^{\neg i}_{d,(z_{i_1},z_{i_2})}}{\sum_{k=1}^{K}\alpha_k + \sum_{k=1}^{K} n^{\neg i}_{d,(z_{i_1},k)}} \\
&\frac{\beta_e + n_{z_{i_1},e_{i_1}}}{\sum_{e=1}^{E}\beta_e + \sum_{e=1}^{E}(n_{z_{i_1},e})} \frac{\beta_{ke} + n_{z_{i_2},e_{i_1}}}{\sum_{e=1}^{E}\beta_{ke} + \sum_{e=1}^{E}(n_{z_{i_2},e})}
\end{aligned}
\tag{4.12}
$$

### 4.6.1 Experiment

**Data Sets**

Four data sets are used in the experiment. They are AP news articles, DUC 2004 task2, Medical Records and Elsevier article papers. Data Preprocessing For both the dataset AP news articles and DUC 2004 task 2 data, we run the open relation extraction tool Reverb to first extract all the open relations from the raw data and use the entity pairs only. For the medical records and Elsevier papers data, we use the entity recognition tool: Metamap (UMLS) to first extract both the symptoms and medications. Then we define the entity pair as the 'symptom, medication', in which the symptom and medication co-occurred in one section of one medical record as the one entity pair. We use the same strategy for the Elsevier data. We defined the entity pair as 'gene, brain part' and extract from the raw data the entities of both the genes and brain parts. The entities are only extracted from the articles except abstract, which is used for later evaluation. We then treat the gene and brain parts co-occurred in one sentence as entity pairs. The overall statistics of our dataset are listed in table 4.3.

**Table 4.3:** Dataset Statistics

| Datasets | # Files | # Words before preprocessing | # Words after Preprocessing |
|---|---|---|---|
| AP News | 2250 | 76848 | 20153 |
| DUC 2004 | 500 | 24713 | 6231 |
| Medical Records | 1249 | 67950 | 1148 |
| Elsevier Papers | 2058 | 141188 | 1132 |

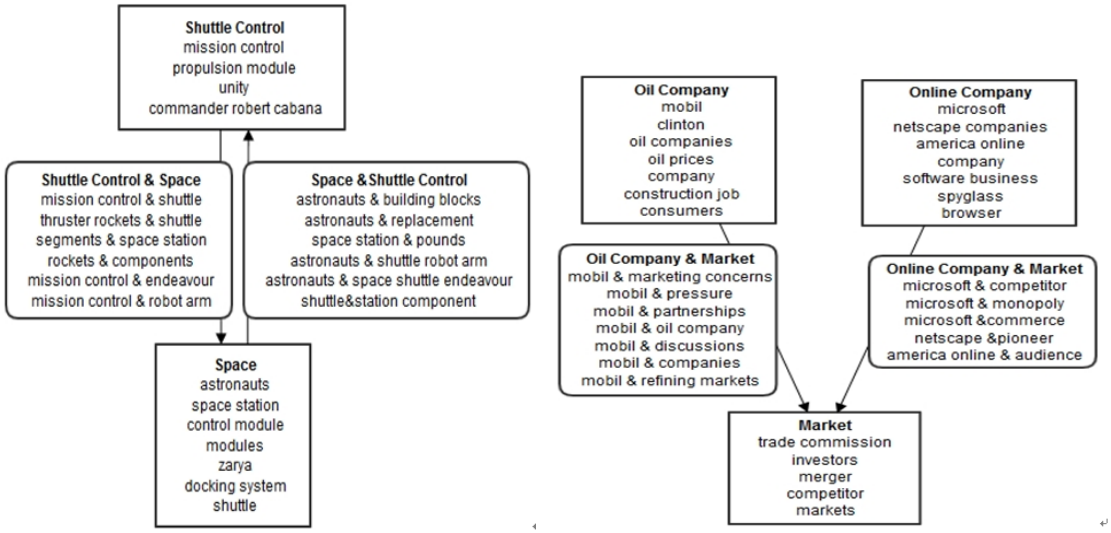
For PTM-1, PTM-2, PTM-3 and PTM-4, we simply set all the super-parameter to 0.1. But for the PTM-5

and PTM-6, the data sparsity problem becomes obvious, as the number of parameters to be calculated becomes much larger. Therefore, we set all the hyper-parameters to be 0.01 to contradict the effect of the priors.

## 4.7 Result and Evaluation

### 4.7.1 Empirical Result

One advantage of the model is that it can capture the pairwise dependency between the topics. Next we will show a subset of the topics and how they are related. The result is obtained from the DUC 2004 dataset. The whole data set covers 50 news event and we only select the news covering 10 events. Therefore, the number of topic is set to 10. We sort the entities in each topic and the entity pairs in each pairwise topic according to their probabilities, and a subset of the topics and the top entity pairs relate them together are shown in Figure 4.4.




(a). Integrate PTM. (b). Separate PTM.

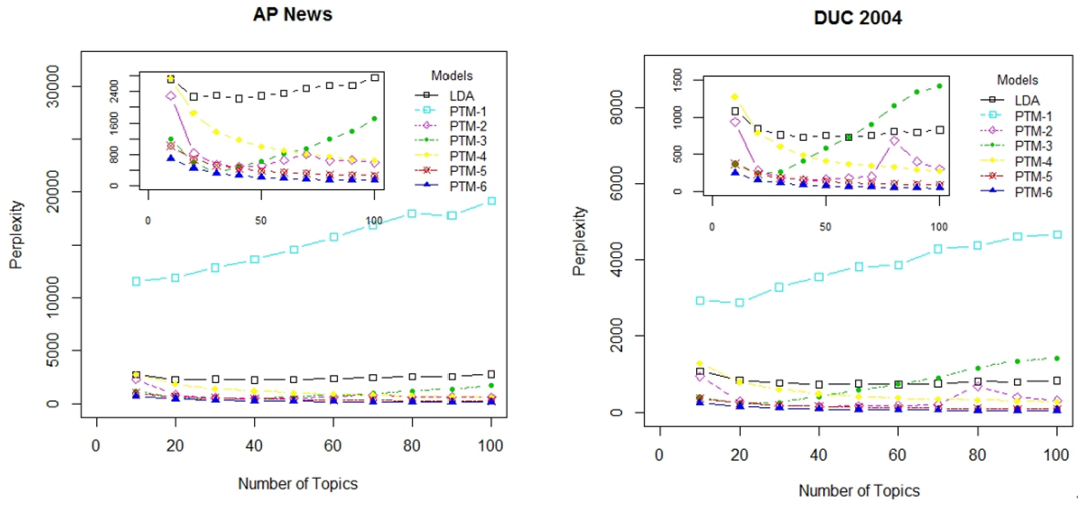
**Figure 4.4:** Topic Relatedness for DUC 2004 dataset

As we can see that the relatedness of the pairwise topics could be effectively explained.

### 4.7.2 Evaluation

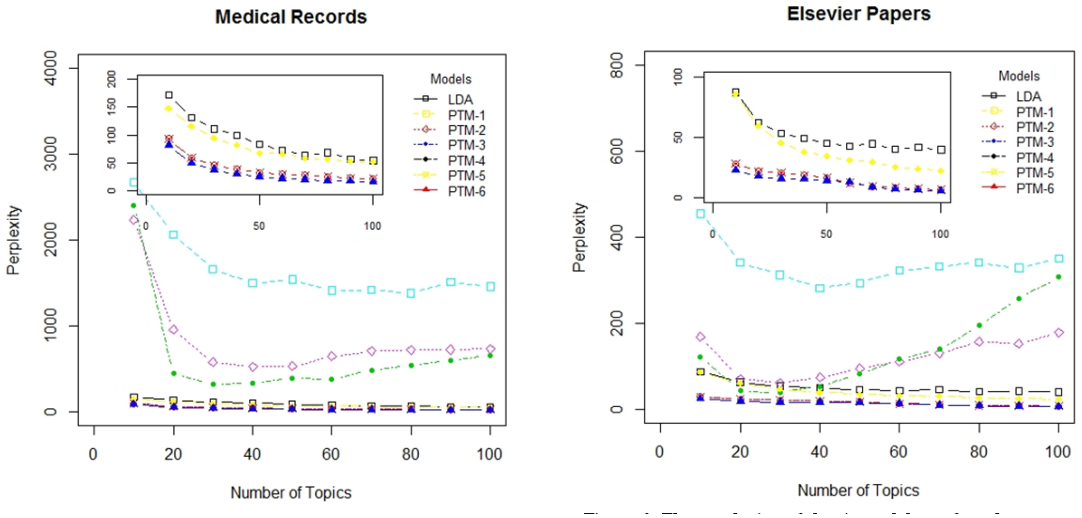
We use the perplexity to evaluate our modeling on the four datasets. The perplexity is widely used as the evaluation for the language model. It is the log likelihood on some unseen held-out, given a language model. For a corpus C of D documents, the perplexity is defined as: Where is the number of unit of text in document d, and w denotes one individual text unit. For traditional LDA, the text unit is word, and for the two models

proposed here, the text unit is entity. We compute and compare the perplexity of PTM-1 to PTM-6 with LDA on all the four datasets. For each dataset, we randomly held-out 80 percent of the document for training and 20 percent of the document for text. The perplexity on held-out data for all the six models is shown in Figure 4.5-4.6.




(a). Integrate PTM. (b). Separate PTM.

**Figure 4.5:** Perplexity comparison for AP news and DUC2004 Datasets




(a). Integrate PTM. (b). Separate PTM.

**Figure 4.6:** Perplexity comparison for Medical Records and Elsver Paper Datasets

Next we will try to answer the two questions asked at the beginning of section 5 according to the perplexity. First, we examine how the choice of one entity or entity pair as one unit could affect the model performance.

We compare the performance of the first three models and other models. LDA generally performs better than; while PTM-4, PTM-5 and PTM-6 perform better than LDA. Thus, the three models PTM-4, PTM-5 and PTM-6 perform better than PTM-1, PTM-2 and PTM-3 on all the four data sets, which is the same as our intuition that we should use one entity as a unit. To examine whether each entity pair carry one topic or two, we make comparison between PTM-1, PTM-2 and PTM-3. We find that the PTM-1 performs worst among all the models, meaning that the entity pair should be generated by two topics, which is also what intuition tells us. Further, comparing PTM-2 and PTM-3, we could find that PTM-3 performs relatively the same as PTM-2 on the dataset presented by the entity pairs of open relation, but better than PTM-2 on the entity pairs of a specific relation. The reason might be that the dependency relation between the entity pair of a specific type is much obvious than the dependency relation between the entity pairs of open relation. For example, the open relation might retrieve two entity pairs from the corpus, such as 'Obama, healthcare' and Healthcare, Obama'. They should be generated from the same topic pairs with opposite direction of dependency. Therefore, the two entity pairs will be assigned different topic pairs, for RTM only model one direction. This also explains why it works better on specific relation dataset, where the entity pairs have definite dependencies. Now we will try to find the best model to capture the dependency relationship for the entity pair structure. We have seen from the above discussion that if we treat entity pair as one unit, whether we should model the topic dependency depends largely on the kind of structure we model. That is, if there is obviously one direction dependency between the entities, the model of dependency between their underlying topics is preferred; while the modeling of dependency makes no difference to the entities that have no definite direction of dependency. Now we will examine how the modeling of dependency affects the performance when each entity is treated as one individual topic carrier. The pattern we get from PTM-4 to PTM-6 on the four different datasets are quiet consistent. PTM-6 performs better than PTM-5 and PTM-5 performs better than PTM-4 on all the four datasets. Therefore, PTM-6 is the best model to capture the entity pair structure.

# Chapter 5: Pairwise Topic Model II

In this chapter, we generalize the word pairs to include all the pairs with mutual information exceeding a certain threshold.

## 5.1 Introduction

The spread of the WWW has led to the boom of explosive Web information. One of the core challenges is to understand the massive document collection with topic transition and evolution.

Among content analysis techniques, topic modeling represents a set of powerful toolkits to describe the process of documents generation. Generally, topic models are based on the unigram and the term based topic models are proved to be a good way to model the languages. While individual words and the hidden topics are the building blocks of language, relations between the terms and underlying topics act as cement that links the words into language structures. This chapter shifts the focus from the terms to the relations by modeling the word pairs instead of individual words. We explore the term association and show the shift to relation achieve greater effectiveness and refinement in topic modeling. To the best of our knowledge, this is the first effort to explicitly model the semantically dependent word pairs.

In this chapter, word pairs refer to two words that are semantically related. They cooccur in the same sentence as in original order, but not necessarily to be consecutive. One advantage of pairwise topic relation is its coverage of the long-range relationship. For example, in the sentence ‘Obama supports healthcare’, the extracted word pairs should be ‘obama, support’, ‘support,healthcare’ and ‘obama, healthcare’. Also, the modeling of relation can in turn facilitates the topic extraction. The relation between ‘obama’ and ‘healthcare’ makes it much easier to identify ‘obama’ as a ‘politician’, and ‘healthcare’ as a ‘policy’. In this work, we will first extract the semantic dependent word pairs through mutual information and then model the dependency within the word pairs.

By considering the semantic dependency between two words, we propose two ways to establish our topic modeling. The first way is to model the related words as a whole unit; the other way is to model each word as

separate units with dependency constraints. The dependencies between the word pairs (if the words are treated separately) and their underlying topics are modeled simultaneously. With dependencies incorporated, it is natural to discover topics hidden in the contexts and find out the evolution trajectories and transition matrix for all discovered topics.

Therefore, the novelties of this part of thesis are as follow.

• Documents are treated as structured data with relations, represented by a bag of word pairs, to facilitate the modeling of the document representation.

• Two different ways of topic modeling with semantic dependencies between words are proposed to characterize the word pair structure and then to capture the pairwise relationship embedded in the structured data.

• We have conducted a thorough experimental study on the news data and literature data to test the performance of topic discovery and then empirically evaluate the evolution and transition among the discovered topics.

This chapter is organized as follows. The second section covers the related work. In the third Section, we propose to examine the document representation, and in Section 4 we describe all the models in details. We will elaborate the model inference process for the proposed models in the fifth Section. Section 6 discusses how the PTM can help facilitate word/topic relation analysis. The experiment and evaluation will be included in Section 7 and Section 8 respectively; while in Section 9, we will draw the conclusion and discuss about future work.

## 5.2 Related Work

Over the years, topic evolution and transition have been studied intensively Chang & Blei (2009), Jo et al. (2011), Wang & McCallum (2006). Miscelleous methods are applied to detect more informative and distinctive topics. Most of the work explores the probabilistic topic modeling over text Blei et al. (2003), Griffiths & Steyvers (2004), Hofmann (2001) and further to integrates topic modeling over text with time series analysis Blei & Lafferty (2006), Wang & McCallum (2006) to obtain the topic evolution. But the pre-defined time granularity makes these time-sensitive models unreliable unless the time interval is appropriately chosen. He et al. (2009) and Wang et al. (2013) leverage citations to find the topic evolution for literature papers. But they

only use citation information, the text information is ignored.

Therefore, in this chapter, we try to find a more expressive topic modeling to explore the topic transition and evolution. The main concern for traditional topic modeling Blei et al. (2003) is its 'bag of words' assumption. Researchers try different methods to overcome the restriction. One of the early efforts to model the relationship among the topics is the CTM model Blei & Lafferty (2007). Instead of drawing the topic proportions of a document from a Dirichlet distribution, CTM model uses a more flexible logistic normal distribution introduce the covariance among the topics. However, this model could only examine if the two topics are related without showing the direction and degree of relatedness. Rather than modeling the correlation implicitly from the topic generation process, most work models the relationship explicitly, either by modeling the relationship between the topics Gruber et al. (2007), Wang et al. (2011), or between the words Wang et al. (2007), Wallach (2006). These models assume the relation between the words or sentences in sequential order. However, the words in sequential order don't necessarily relate to each other semantically, making the assumption unreasonable. Also, they can not model the words and topics simultaneously.

Except to model the sequential words, Chen et al. (2009) or the syntactic information Boyd-Graber & Blei (2009) also model the position and syntactic structure. Among all these methods, none explores the relation between word pairs.

Therefore, in our work, we will investigate the topic dependency among semantically dependent words. By firstly extracting the potential dependent word pairs, we are more confident to capture meaning dependency relationship and furthermore meaningful topic transition and evolution.

## 5.3 Document Representation

The pairwise topic model is to model the data composed of word pairs and links between them. It embeds the word pairs in a latent space that explains both the word and the topic relationship. We will give more insight into the document space of word pairs in this section.

**Document Manipulation with Mutual Information.** Different from the traditional topic modeling of manipulating the individual words, the pairwise topic model takes word pairs as input. The topic model with semantic relationship will be effective only when the processed word pairs do have significant topic dependencies between each other. Accordingly, we extract prominent word pairs out of the documents. We

measure the semantic dependency between two words through mutual information. The mutual information $I(w_1, w_2)$ between two words $w_1$ and $w_2$ is defined as:

$$I(w_1, w_2) = p(w_1, w_2) \log \frac{p(w_1)p(w_2)}{p(w_1, w_2)} \propto \frac{N_s^{w_1} N_s^{w_2}}{N_s^{(w_1, w_2)}} \tag{5.1}$$

Where $N_s^{w_1}$ and $N_s^{w_1}$ are the number of times word $w_1$ and word $w_2$ appear respectively in one sentence, while $N_s^{(w_1, w_2)}$ are the times $w_1$ and $w_2$ appear in the same sentence.

Therefore, we assume the two words have semantic dependency when their mutual information exceeds a pre-defined threshold. Hence, we change the representation of the unstructured documents into the structured data in form of word pairs, and then model the explicit dependency in word pairs.

By taking the semantic dependency, namely relationship, between two words, this model will offer us more insight for document analysis.

## 5.4 Pairwise Topic Model

The pairwise topic model(PTM) is a generative model of document collections. Different from previous work, it is to examine the dependency between the words and their underlying topics via the word pairs, assuming the dependency within the word pair and independency among the word pairs. Thus, the key part is to model the dependency between two words and theirs corresponding topics. We propose two models: PTM-1 and PTM-2 to examine the semantic dependency in detail.

The first model, the integrated pairwise topic model (PTM-1), arises from the intuition that two words and their link together represent a semantic unit. Two words with the link form one unit generated by the whole topic pair, with the second topic dependent on the first one.

The second model, the separated pairwise topic model (PTM-2), treats each word as one individual unit and explicitly model the relationship between words. In PTM-2, the generation of the second word is determined not only by its topic, but also by the first word and its corresponding topic. Both PTM-1 and PTM-2 allow each document to exhibit multiply topic transition with different proportions. We use the following terminology and notation in Table 5.1 to describe the data, latent variables and parameters in the PTM models.

Specifically, the pairwise topic model assumes that a document arises from the following generative

**Table 5.1:** Annotations in the generative process for topic evolution model.

| Notation | Description |
|---|---|
| $D$ | Number of the documents |
| $V$ | Number of the words |
| $V_p$ | Number of the word pairs |
| $w_p(w_1, w_2)$ | Word pair |
| $z_p(z_1, z_2)$ | Underlying topic pair for each word pair |
| $\alpha_k$ | Dirichlet prior for $\theta_d$ |
| $\alpha'_k$ | Dirichlet prior for $\theta_{d,k}$ |
| $\beta_w$ | Dirichlet prior for $\Phi_k$ |
| $\beta_{w_p}$ | Dirichlet prior for $\Phi_{k_p}$ |
| $\beta'_{w_p}$ | Dirichlet prior for $\Phi_{k_p,w}$ |
| $\theta_d$ | Topic distribution for document $d$ |
| $\theta_{d,k}$ | Topic transition distribution from topic $k$ for document $d$ |
| $\Phi_k$ | Word distribution for each topic $k$ |
| $\Phi_{k_p}$ | Word pair distribution for each topic pair $k_p$ |
| $\Phi_{k_p,w}$ | Word distribution given the topic pair and the first word is $k_p$ and $w$ respectively. |

processes.

For PTM-1, the generative process is as follow:

1. For each topic pair $k_p(k_1, k_2)$ ($k_1$, $k_2$ = 1, 2, 3, . . . K),

(a) Draw a (topic pair - word pair) distribution for each topic pair $k_p$:

$$\Phi(k_p) \sim \text{Dirichlet}(\beta_{w_p})$$

2. For each document d (d $\in$ 1, 2,. . . , D),

(a) Draw a document specific topic distribution:

$$\theta_d \sim \text{Dirichlet}(\alpha_k)$$

(b) Draw a document specific topic transition distribution for each topic k:

$$\theta_{d,k} \sim \text{Dirichlet}(\alpha'_k)$$

(c) For each word pair

(i) Draw the first topic from the document-topic distribution:

$$z_1 \sim \text{Categorical}(\theta_d)$$

(ii) Draw the second topic from the topic transition probability conditioned on the first topic:

$$z_2 \sim \text{Categorical}(\theta_{d,z_1})$$

(iii) Draw the word pair from the topic pair-word pair distribution:

$$w_p \sim \text{Categorical}(\Phi_{z_p})$$

During the generative process, the word pair is treated as one unit and generated from the dependent topic pair. This model is to check if two words as a text unit is sufficient to capture the topic transition.

Next, we propose PTM-2 to simulate the more intricate relationship between the words of a pair adding the dependency between the words. The generative process for PTM-2 is:

1. For each topic k (k = 1, 2, 3, . . . , K),

(a) Draw a topic-word distribution for each topic:

$$\Phi_k \sim \text{Dirichlet}(\beta_w)$$

2. For each topic pair $k_p(k_1, k_2)$ ($k1, k2$ = 1, 2, 3, . . . , K) and a word w (w = 1, 2, 3,. . . , V)

(a) Draw a word distribution:

$$\Phi_{k_p,w} \sim \text{Dirichlet}(\beta'_w)$$

3. For each document d (d = 1, 2,. . . , D),

(a) Draw a document specific topic distribution:

$$\theta_d \sim \text{Dirichlet}(\alpha_k)$$

(b) Draw a specific topic transition distribution from topic k:

$$\theta_{d,k} \sim \text{Dirichlet}(\alpha'_k)$$

(c) For each word pair,

(i) Draw the topic of the first word from the document-topic distribution:

$$z_1 \sim \text{Categorical}(\theta_k)$$

(ii) Draw the first word from the topic-word distribution:

$$w_1 \sim \text{Categorical}(\Phi_{z_1})$$

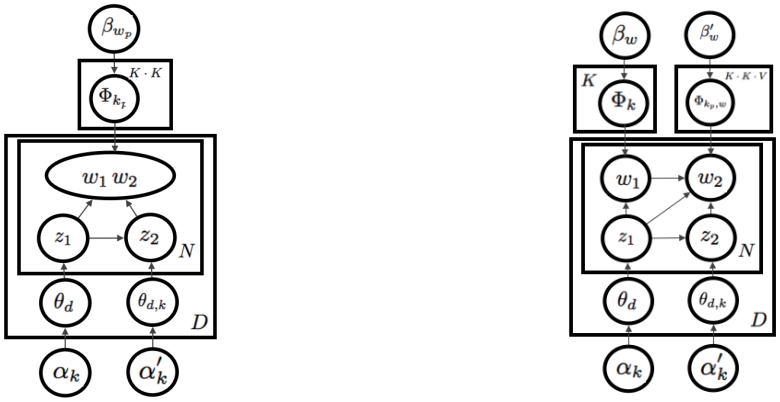
(iii) Draw the topic of the second word from the topic transition distribution conditioned on the first topic:

$$z_2 \sim \text{Categorical}(\theta_{z_1})$$

(iv) Draw the second word from (topic pair, word) $\sim$ word distribution:

$$w_2 \sim \text{Categorical}(\Phi_{(z_p,w_1)}))$$

PTM-2 treats every individual word in the pair as the text unit. In addition to the topic of the second word, the first word and its corresponding topic also contribute to the generation of the second word.




(a). Integrate PTM. (b). Separate PTM.

**Figure 5.1:** Two ways for graphical representation for pairwise topic modeling.

The joint probability of PTM-1 can be illustrated as following:

$$
\begin{aligned}
& p(W, Z, \theta_d, \theta_{d,k}, \Phi_{z_p} | \alpha_k, \alpha'_k, \beta_{w_p}) \\
=& \prod_{d=1}^{D} \frac{\Gamma(\sum_{z=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha_k - 1} \prod_{d=1}^{D} \prod_{k=1}^{K} \frac{\Gamma(\sum_{k'=1}^{K} \alpha'_{k'})}{\prod_{k'=1}^{k} \Gamma(\alpha'_{k'})} \prod_{k'=1}^{K} \theta_{d,k'}^{\alpha_{k'|k} - 1} \\
& \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \frac{\Gamma(\sum_{w_p=1}^{V_p} \beta_{w_p})}{\prod_{w_p=1}^{V_p} \Gamma(\beta_{w_p}) \prod_{w_p=1}^{V_p}} \Phi_{k_p}^{\beta_{w_p} - 1} \\
& \prod_{d=1}^{D} \prod_{k_1=1}^{K} \theta_d^{n_{k_1}} \prod_{d=1}^{D} \prod_{k_1}^{K} \prod_{k_2=1}^{K} \theta_{d,k_1}^{n_{d,k_2|k_1}} \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \prod_{w_p=1}^{V_p} \Phi_{k_p}^{n_{k_p,w_p}} \\
=& (\frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)})^D \prod_{d=1}^{D} \prod_{k=1}^{D} \theta_d^{\alpha_k + n_k - 1} \\
& (\frac{\Gamma(\sum_{k'=1}^{K} \alpha_{k'})}{\prod_{k_\prime=1}^{K} \Gamma(\alpha_{k'})})^{DK} \prod_{d=1}^{D} \prod_{k=k_1=1}^{K} \prod_{k'=k_2=1}^{K} \theta_{d,k}^{\alpha_{k'|k} + n_{d,k_2|k_1} - 1} \\
& (\frac{\Gamma(\sum_{w_p=1}^{V_p} \beta_{w_p})}{\prod_{w_p=1}^{V_p} \Gamma(\beta_{w_p})})^{KK} \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \prod_{w_p=1}^{V_p} \Phi_{k_p}^{\beta_{w_p} + n_{k_p,w_p} - 1}
\end{aligned}
\tag{5.2}
$$

The joint probability of PTM-2 is:

$$
\begin{aligned}
& p(W, Z, \theta_d, \theta_{d,k}, \Phi_k, \Phi'_{k_p} | \alpha_k, \alpha'_k, \beta_w, \beta'_{w_p}) \\
=& \prod_{d=1}^{D} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{d,k}^{\alpha_k - 1} \prod_{d=1}^{D} \\
& \prod_{k=1}^{K} \frac{\Gamma(\sum_{k'=1}^{K} \alpha'_{k'})}{\prod_{k'=1}^{k} \Gamma(\alpha'_{k'})} \prod_{k'=1}^{K} \theta_{d,k'}^{\alpha_{k'|k} - 1} \\
& \prod_{k=1}^{K} \frac{\Gamma(\sum_{w=1}^{V} \beta_w)}{\prod_{w=1}^{V} \Gamma(\beta_w) \prod_{w=1}^{V}} \Phi_k^{\beta_w - 1} \\
& \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \frac{\Gamma(\sum_{w'=1}^{V} \beta'_{w_p})}{\prod_{w'=1}^{V} \Gamma(\beta'_{w_p}) \prod_{w'=1}^{V}} \Phi_{k_p, w'}^{\beta'_{w_p} - 1} \\
& \prod_{d=1}^{D} \prod_{k_1=1}^{K} \theta_d^{n_{k_1}} \prod_{d=1}^{D} \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \theta_{d,k_1}^{n_{d,k_2|k_1}} \\
& \prod_{k_1=1}^{K} \prod_{w_1=1}^{V} \Phi_{k_1}^{n_{k_1}} \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \prod_{w_1=1}^{V} \prod_{w_2=1}^{V} \Phi_{k_p, w_1}^{n_{k_p, w_p}} \\
=& (\frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)})^D \prod_{d=1}^{K} \prod_{k=1}^{D} \theta_d^{\alpha_k + n_k - 1} \\
& (\frac{\Gamma(\sum_{k'=1}^{K} \alpha_{k'})}{\prod_{k'=1}^{K} \Gamma(\alpha_{k'})})^{DK} \prod_{d=1}^{D} \prod_{k=k_1=1}^{K} \prod_{k'=k_2=1}^{K} \theta_{d,k}^{\alpha_{k'|k} + n_{d,k_2|k_1} - 1} \\
& (\frac{\Gamma(\sum_{w=1}^{V} \beta_w)}{\prod_{w=1}^{V} \Gamma(\beta_w)})^K \prod_{k=1}^{K} \prod_{w=1}^{V} \Phi_k^{\beta_w + n_{k,w} - 1} \\
& (\frac{\Gamma(\sum_{w=1}^{V} \beta'_{w_p})}{\prod_{w=1}^{V} \Gamma(\beta'_{w_p})})^{KKV} \prod_{k_1=1}^{K} \prod_{k_2=1}^{K} \prod_{w_1=1}^{K} \prod_{w_2=1}^{V} \Phi_k^{\beta'_{w_p} + n_{k_p, w_p} - 1}
\end{aligned}
\tag{5.3}
$$

Overall, both the models can capture the topic dependency of a word pair and find the topic transition and be more expressive than traditional topic models.

The two generative processes are illustrated as probabilistic graphical models in Figure 5.1.

## 5.5 Inference

We use Gibbs sampling to perform model inference. Due to the space limit, we leave out the derivation details and only show the sampling formulas. The notations for the sampling formulas are as shown in Table 5.2.

**Table 5.2:** Notations for the inference of topic evolution model.

| Notation | Description |
|---|---|
| $n_{d,z_i}^{\neg i}$ | Number of words assigned to topic $z_i$ in document $d$ except for the current word |
| $n_{d,z_2\|z_1}^{\neg i}$ | Number of the second entity assigned to topic $z_{e_2}$ given the topic of the first entity is $z_{e_1}$ except for the current entity pair |
| $n_{z_i,e_i}^{\neg i}$ | Number of entity pair $e_i$ assigned topic zi except for the current entity pair |
| $n_{(z_{i1},z_{i2}),e}^{\neg i}$ | Number of entity pair e assigned to topic pair $(z_{e_1}, z_{e_2})$ except for the current entity pair |
| $n_{d,z_{i2}\|z_{i1}}^{\neg i}$ | Number of topic $z_{i1}$ transformed from topic $z_{e_1}$ to topic $z_{e_2}$ except for the current entity pair |
| $n_{(z_{i2},e_{i1}),e_{i2}}^{\neg i}$ | Number of the second entity $e_{i2}$ assigned to $z_{e_2}$, given the first entity is $e_{i1}$. |

For PTM-1, we have the following sampling formula:

$$\begin{aligned} & p(z_{i1}, z_{i2}|W, Z_{\neg i}, \alpha_k, \beta_{w_p}, \alpha'_k) \\ \propto & \frac{\alpha_k + n_{d,z_{i1}}^{\neg i}}{\sum_{k_1=1}^{K} \alpha_k + \sum_{k_1=1}^{K} n_{d,k_1}^{\neg i}} \\ & \frac{\alpha'_k + n_{d,z_{i2}|z_{i1}}^{\neg i}}{\sum_{k_2=1}^{K} \alpha'_k + \sum_{k_2=1}^{K} n_{d,k_2|z_{i1}}^{\neg i}} \\ & \frac{\beta_{w_p} + n_{z_i,w_i}^{\neg i}}{\sum_{w_p=1}^{V_p} \beta_{w_p} + \sum_{w_p=1}^{V_p} n_{z_i,w_p}^{\neg i}} \end{aligned} \tag{5.4}$$

For RTM-2, the sampling formula is:

$$\begin{aligned} & p(z_{i1}, z_{i2}|W, Z_{\neg i}\alpha_k, \beta_w, \alpha'_k, \beta_{k_p,w}) \\ \propto & \frac{\alpha_k + n_{d,z_{i1}}^{\neg i}}{\sum_{k_1=1}^{K} \alpha_k + \sum_{k_1=1}^{K} n_{d,k_1}^{\neg i}} \\ & \frac{\alpha'_k + n_{d,z_{i2}|z_{i1}}^{\neg i}}{\sum_{k_2=1}^{K} \alpha'_k + \sum_{k_2=1}^{K} n_{d,k_2|z_{i1}}^{\neg i}} \\ & \frac{\beta_w + n_{z_{i1},w_{i1}}^{\neg i}}{\sum_{w_1=1}^{V} \beta_w + \sum_{w_1=1}^{V} n_{z_{i1},w_1}^{\neg i}} \\ & \frac{\beta'_{w_p} + n_{(z_i,w_{i2}}^{\neg i}}{\sum_{w_2=1}^{V} \beta'_{w_p} + \sum_{w_2=1}^{V} n_{z_i,(w_{i_1},w_2)}^{\neg i}} \end{aligned} \tag{5.5}$$

## 5.6 Model Analysis

PTM models focus on the study of relation analysis through word pairs. The following information can be obtained from the models.

1. Parameters obtained from PTM-1:

a. Topic & topic transition distribution for each document $d(d \in D)$: $\theta_d$, $\theta_{d_k}$.

b. Word pair distribution for each ordered topic pair: $\Phi_{k_p}$.

2. Parameters obtained from PTM-2:

a. Topic & topic transition distribution for each document $d(d \in D)$: $\theta_d$, $\theta_{d_k}$.

b. Word distribution for each topic $k(k \in T)$: $\Phi_k$.

c. Word distribution given topic pair and previous word: $\Phi_{k_p,w}$.

The focus of pairwise topic models are to model the word/topic relatedness. Two types of relatedness can be obtained based on the parameters of the models.

The first is the transition probability for each ordered topic pair. Both PTM-1 and PTM-2 can obtain the topic transition probability for each document.

The second is the word pair distribution of each topic pair. For PTM-1, it can be obtained from $\Phi_{k,p}$ and for PTM-2, it can be obtained from $\Phi_k$ and $\Phi_{k_p,w}$. The calculations are provided as follows.

PTM-1:

$$\begin{aligned} & p((w_1, w_2)|(z_1, z_2)) = p((w_2, w_1)|(z_1, z_2)) \\ & = \Phi_{(z_1,z_2)} \end{aligned} \tag{5.6}$$

PTM-2:

$$\begin{aligned} & p(w_1, w_2|(z_1, z_2)) = p(w_2|w_1, z_1, z_2)p(w_1|z_1) \\ & = \Phi_{z_1}\Phi_{(z_1,z_2),w_1} \end{aligned} \tag{5.7}$$

In the following section, we will demonstrate how the modeling of the relation can facilitate the topic extraction and show how the relatedness model can help us analysis the word/topic relations.

**Table 5.3:** Dataset Statistics for topic evolution model

| Datasets | News | Literature |
|---|---|---|
| Size before Processing | 12.7M | 10.4M |
| # of Documents | 2,000 | 16,000 |
| # Words before preprocessing | 1,182,152 | 1165,305 |
| # Vocabulary before preprocessing | 41,181 | 211,351 |
| # Words after preprocessing | 1,323,777 | 1,561,677 |
| # Vocabulary after preprocessing | 8,380 | 12,575 |
| # Unique Word pairs after prepro | 27,100 | 525,513 |
| Size after Processig | 10.2M | 11.2M |

## 5.7 Experiment and Evaluation

### 5.7.1 Datasets

We run our experiments based on two large document collections: one is online news webpages, and the other one is literature of research papers. We compute and compare with different evaluation metrics for both topic models of PTM-1 and PTM-2 on both datasets. For each dataset, we randomly held-out 80% of the documents for training and 20% of the documents for testing.

For the news documents, we use the documents related with several famous topics published by popular news agencies such as CNN, BBC, and ABC news, etc Yan et al. (2011a) Yan et al. (2011b). The topics include 'BP Oil Spill', 'Influenza H1N1' and 'Arab Spring'.

For the literature of research papers, we use data from Tang et al. (2008), which is extracted from academic search and mining platform ArnetMiner[1]. It covers 1,558,499 papers from major Computer Science publication venues and has gathered 916,946 researchers for more than 50 years (from 1960 to 2010).

**Data Preprocessing.** The number of words pairs is extremely huge if we treat each and every two words in one sentence as word pairs. Thus, we expect the measured word pairs to be important and have higher relevance with a large probability. Therefore, we first filtered out stop words and other insignificant words by calculating tf-idf scores and discarding words with low scores. The final step is to remove all irrelevant word pairs through the mutual information. The statistics of the two corpus are shown in Table 5.3. The same word in original text may be repeated in different word pairs, making the number of words after processing larger than the number of words before processing.

[1]Downloaded from `http://arnetminer.org/citation`.

### 5.7.2 Comparison Methods

Here we compare with two traditional, but very popular topic models: LDA Blei et al. (2003)(Latent Dirichlet Allocation) and CTM Blei & Lafferty (2007) (Correlated Topic Model). LDA is also a generative model with each word corresponding to one topic. However, it ignores the topic correlation residing in the words. In the generative process of one document, the LDA model first select a topic from a document specific topic distribution, and then select a word from the word distribution of the selected topic. Thus, in LDA, one document consists of a proportion of topics represented by a number of words. The CTM, as mentioned in section 2, follows the same generation strategy as LDA, except that it uses a logistic normal distribution instead of Dirichlet distribution for the topic distribution.

### 5.7.3 Evaluation Metric and Parameter Setting

We will show in this section the topics discovered by the topic modeling methods empirically: top-8 topic words for each topic. Furthermore, we will provide more illustration graphs for topic evolution and transition as supplementary investigation.

Another set of experiments involves intrinsic evaluation of the 'perplexity' approach. Perplexity is to measure how well a probability model predicts a sample, and is a widely-used metrics to compare the probability models. The perplexity of the whole corpus is defined as:

$$\text{Perplexity(D)} = \exp\left( - \frac{\sum_{n=1}^{N_d} \log \text{p}(w_n)}{\sum_{d=1}^{D} N_d} \right) \tag{5.8}$$

where $N_d$ is the number of words in document $d$, and $w$ denotes one individual word unit.

## 5.8 Result and Analysis

We will first show the topics discovered by the topic modeling methods empirically: top-8 topic words for each topic. Further more, we will provide more illustration graphs for topic evolution and transition as supplementary investigation. As we will see in the later perplexity result, the perplexity doesn't change much for different number of topics. Therefore, we choose the number to be 50 empirically.

**Table 5.4:** Top 10 Topic Words within the Sample Topics by LDA

| NEWS-2 | NEWS-8 | NEWS-10 | NEWS-19 | NEWS-20 |
|---|---|---|---|---|
| space:0.034 | flu:0.197 | health:0.125 | president: 0.017 | bp: 0.031 |
| science:0.027 | swine:0.098 | care:0.119 | foreign: 0.016 | spill: 0.020 |
| log:0.024 | pet:0.040 | insurance:0.082 | minister: 0.012 | gulf: 0.016 |
| nasa:0.021 | melamine:0.030 | chinese:0.046 | secretary: 0.012 | rig: 0.015 |
| cosmic:0.015 | pandemic:0.026 | china:0.031 | international: 0.010 | day: 0.012 |
| yle:0.013 | virus:0.022 | reform:0.024 | bush: 0.009 | coast: 0.011 |
| mr:0.012 | gluten:0.020 | healthcare:0.024 | ceasefire: 0.009 | leak: 0.008 |
| galaxy:0.012 | protein:0.019 | medicare:0.022 | council: 0.008 | mexico: 0.008 |
| msnbc:0.012 | wheat:0.017 | coverage:0.018 | ban: 0.008 | guard: 0.007 |
| planet:0.012 | fda:0.015 | payer:0.014 | storage: 0.006 | high: 0.004 |
| space exploration | flu | healthcare | government | oil spill |
| NEWS-21 | NEWS-28 | NEWS-33 | NEWS-38 | NEWS-40 |
| president: 0.017 | bahrain: 0.044 | human: 0.024 | al:0.052 | gaza:0.160 |
| foreign: 0.016 | government: 0.025 | myanmar: 0.024 | mccain:0.022 | israeli:0.088 |
| minister: 0.012 | al: 0.018 | council: 0.018 | peninsula:0.019 | israel:0.083 |
| secretary: 0.012 | opposition: 0.014 | situation: 0.016 | gcc:0.018 | hamas:0.062 |
| international: 0.010 | protest: 0.012 | government: 0.015 | lisa:0.016 | hama:0.047 |
| bush: 0.009 | people: 0.012 | special: 0.014 | doctor:0.016 | palestinian:0.044 |
| ceasefire: 0.009 | shia: 0.011 | rights: 0.013 | emanuel:0.016 | rocket:0.021 |
| council: 0.008 | sunni: 0.010 | international: 0.012 | baby:0.013 | ashkelon:0.017 |
| ban: 0.008 | bahraini: 0.010 | rrb: 0.012 | shield:0.013 | strip:0.016 |
| storage: 0.006 | model: 0.008 | guidance: 0.001 | bahrain:0.012 | rockets:0.013 |
| government | protest | human rights | medication | conflict area |
| PAPER-0 | PAPER-4 | PAPER-6 | PAPER-14 | PAPER-15 |
| user:0.148 | learning:0.219 | agent:0.187 | web: 0.037 | design:0.221 |
| interface:0.133 | structure:0.111 | computing:0.096 | based: 0.036 | evaluation:0.098 |
| ability:0.090 | training:0.076 | reasoning:0.057 | service: 0.031 | optimization:0.067 |
| step:0.072 | resources:0.046 | coordination:0.025 | management: 0.016 | metrics:0.051 |
| techniques:0.062 | terms:0.034 | logic:0.024 | distributed: 0.015 | change:0.047 |
| interfaces:0.059 | relation:0.031 | negotiation:0.017 | business: 0.013 | robot:0.041 |
| variety:0.046 | behaviour:0.029 | transaction:0.014 | user: 0.011 | principles:0.030 |
| underlying:0.014 | probability:0.028 | team:0.013 | grid: 0.010 | improving:0.017 |
| interpretation:0.014 | machine:0.013 | belief:0.012 | process: 0.009 | robots:0.016 |
| peer:0.012 | distributions:0.012 | multi:0.012 | architecture: 0.009 | designers:0.015 |
| interface | learning | theory | web service | design |
| PAPER-31 | PAPER-37 | PAPER-41 | PAPER-43 | PAPER-49 |
| security: 0.058 | patterns:0.214 | software:0.162 | model: 0.05 | field:0.115 |
| students:0.016 | planning:0.085 | project:0.085 | based: 0.02 | interaction:0.088 |
| science: 0.014 | goal:0.083 | impact:0.068 | object: 0.01 | location:0.075 |
| university: 0.009 | classification:0.068 | engineering:0.063 | modeling: 0.01 | mapping:0.073 |
| education: 0.006 | domains:0.055 | development:0.047 | language: 0.01 | interactions:0.048 |
| secure: 0.006 | analyzed:0.036 | effort:0.033 | systems: 0.01 | sense:0.046 |
| program: 0.006 | chain:0.036 | projects:0.025 | approach: 0.00 | map:0.040 |
| teaching: 0.005 | action:0.034 | nature:0.024 | analysis: 0.00 | evidence:0.031 |
| technology: 0.004 | storage: 0.006 | production:0.021 | oriented: 0.00 | map:0.024 |
| high: 0.004 | storage: 0.006 | developers:0.012 | formal: 0.00 | element 0.022 |
| security | pattern recognition | software management | model | location detection |

### 5.8.1 Topic Demonstration

In this section, we will compare the topic words obtained from both the LDA and PTM-2. For the topic demonstration, the CTM follows the same topic generation strategy as LDA, thus only the topic words from LDA are shown here. For PTM-1 and PTM-2, we only show the results from PTM-1. To find the representative topics over the years, we select 8 topics among the topics of top strength each year and list the top 10 words discovered by our proposed method against LDA in Table 5.4-5.5.

As we could see, the topics obtained by PTM-2 has more distinctive power. For example, both the LDA and PTM-2 get the topic ‘user interface’ out of text, but the ‘user interface’ obtained by LDA have both the

**Table 5.5:** Top 10 Topic Words within the Sample Topics by PTM

| NEWS-11 | NEWS-13 | NEWS-15 | NEWS-18 | NEWS-25 |
|---|---|---|---|---|
| nasa:0.106 | swine:0.258 | reactor: 0.114 | health:0.16 | israeli:0.184 |
| cosmic:0.076 | flu:0.224 | plant: 0.101 | care:0.094 | gaza:0.152 |
| sep:0.059 | pig:0.024 | radiation: 0.100 | insurance:0.074 | israel:0.116 |
| planetary:0.026 | viru:0.024 | nuclear: 0.055 | south:0.059 | palestinian:0.052 |
| brain:0.026 | outbreak:0.012 | japan: 0.048 | medicare:0.022 | hama:0.048 |
| rover:0.025 | influenza:0.012 | tepco: 0.039 | payer:0.021 | hamas:0.029 |
| galaxy:0.019 | pandemic:0.010 | fuel: 0.038 | hayward:0.019 | jazeera:0.028 |
| orbit:0.018 | vaccine:0.009 | power: 0.026 | launch:0.019 | humanitarian:0.028 |
| tripoli:0.015 | baby: 0.006 | fukushima: 0.017 | resistant:0.010 | correspondent:0.017 |
| rocket:0.014 | obama: 0.006 | radioactive: 0.016 | patient:0.009 | strip:0.016 |
| space exploration | influenza | nuclear | health care | conflict |
| NEWS-27 | NEWS-41 | NEWS-45 | NEWS-46 | NEWS-47 |
| department: 0.112 | haiti: 0.210 | bahrain: 0.521 | bp:0.195 | people:0.201 |
| secretary: 0.106 | earthquake: 0.123 | medical: 0.057 | spill: 0.102 | council:0.099 |
| safety:0.081 | chile: 0.076 | doctor: 0.053 | gulf: 0.081 | rights:0.065 |
| obama: 0.068 | quake: 0.054 | hospital: 0.025 | coast: 0.053 | washington:0.057 |
| operation: 0.020 | port: 0.053 | unrest: 0.022 | operation: 0.033 | ban:0.044 |
| canadian: 0.014 | food: 0.042 | report:0.017 | rig: 0.022 | boat:0.036 |
| america: 0.013 | company: 0.040 | ambassador: 0.008 | drill: 0.021 | injury:0.024 |
| administration: 0.013 | tsunami: 0.01 | protection: 0.007 | leak: 0.019 | european:0.024 |
| aerial: 0.013 | offshore: 0.013 | twitter: 0.007 | containment:0.013 | military:0.021 |
| assembly: 0.010 | dr: 0.01 | kingdom:0.005 | drilling:0.011 | main:0.020 |
| government | earthquake | medical support | oil spill | council |
| PAPER-1 | PAPER-2 | PAPER-6 | PAPER-11 | PAPER-21 |
| interface:0.083 | concepts:0.071 | task:0.066 | data: 0.118 | services: 0.193 |
| context:0.044 | reasoning:0.068 | experiment:0.050 | security: 0.073 | commerce: 0.059 |
| challenge:0.041 | values:0.053 | extraction:0.047 | secure: 0.044 | mobile: 0.049 |
| designing:0.039 | concept:0.039 | tasks:0.039 | networks: 0.037 | environment: 0.030 |
| learning:0.038 | description:0.028 | conducted:0.036 | electronic: 0.023 | internet: 0.030 |
| developing:0.037 | attributes:0.026 | size:0.028 | tools: 0.0234 | web: 0.029 |
| multimedia:0.037 | transfer:0.025 | speed:0.028 | protocols: 0.018 | network: 0.023 |
| understanding:0.036 | methods:0.021 | form:0.027 | structure:0.018 | trust: 0.020 |
| interactions:0.034 | intelligence:0.018 | noise:0.026 | perform: 0.016 | market: 0.019 |
| complex:0.030 | forms:0.018 | text:0.023 | key 0.012 | customers: 0.013 |
| interface | concept | experiment | network security | web service |
| PAPER-30 | PAPER-31 | PAPER-36 | PAPER-37 | PAPER-41 |
| logic:0.071 | model :0.052 | pattern:0.137 | project:0.083 | sensor:0.061 |
| architecture:0.047 | theory: 0.042 | methods:0.052 | construction:0.080 | phase:0.053 |
| mapping:0.039 | digital:0.040 | metrics:0.043 | step:0.068 | levels:0.042 |
| design:0.039 | agent: 0.026 | sense:0.040 | aspects:0.041 | location:0.035 |
| engine:0.038 | engineer: 0.020 | text:0.034 | people:0.0379 | fault:0.035 |
| core:0.035 | optimal: 0.019 | translation:0.033 | goal:0.037 | markov:0.033 |
| architectures:0.035 | space: 0.015 | effort:0.033 | flow:0.034 | cluster:0.030 |
| consists:0.030 | similarity:0.015 | learning:0.032 | activities:0.031 | activity:0.029 |
| employed:0.028 | interaction: 0.015 | structure:0.029 | access:0.027 | respect:0.027 |
| supports:0.021 | cognitive: 0.015 | parameters:0.025 | output:0.021 | details:0.027 |
| archetecture | model | pattern recognition | project management | location detection |

'design' part and 'user experience' part in the same topic, while the topics obtained by PTM-2 have the two parts separated as 'interface' and 'user scheme' respectively in topics PAPER-1 and PAPER-3.

### 5.8.2 Topic Pair Demonstration

This section show how we can analyze the word/topic relation from topic pairs through PTM-2 on the Literature dataset. First we show the self-transition of 'Web Service' (topic 21) and 'user interface' (topic 1) in Table 5.6. The top five word pairs are selected. We also show the top word transition pairs from 'web service' (topic 21) to 'project management' (topic 37) and from 'interface' (topic 1) to 'web service' (topic 21).

From the word pairs under the topic pairs, it is easier to understand how the topics are related.

**Table 5.6:** Top Word Transition Pair under Topic Transition Pair for Literature

| PAPER-21 → PAPER-21 | PAPER-1 → PAPER-1 |
|---|---|
| web → services:0.122 | interface → pilot:0.021 |
| service → tcp:0.036 | range → inference:0.007 |
| resource → architecture:0.019 | control → sampling:0.006 |
| service → perspective 0.018 | control → values:0.006 |
| business → language:0.017 | interfaces → range:0.004 |
| web service → web service | interface → interface |
| **PAPER-1 → PAPER-37** | **PAPER-1 → PAPER- 21** |
| interface → project:0.0153 | interface → web:0.0351 |
| interface → program:0.115 | interface → protocols:0.03 |
| control → processes:0.009 | interface → networking:0.01 |
| control → interface-:0.009 | interface → search:0.01 |
| interface → operations:0.008 | control→ operations:0.009 |
| interface → project management | interface → web service |

Next some complementary evidence to illustrate the model performance will be provided.

### 5.8.3 Topic Strength

Fifty topics are too many to demonstrate in one graph, and we manually cluster the fifty small topics into five big categories for news corpus and six for literature corpus, and will use the categories as the big topics. The five categories for news are Politics, Space Exploration, Health and Medication, Military and Entertainment; while the six research topics for literatures are Modeling and Algorithm, Learning, Object Detection, HCI, Engineering and Network. We choose the top 7 topics each year for the news and 5 topics each year for the literature, and list their topics and corresponding categories in Table 5.7-5.8.

Next, we will show the change of topic strength over the years. Since we can not present the 50 topics all in the same graph, we will show the change of strength for each category accordingly. The category strength of each year is defined as the cumulation of the topics proportion under that category:

$$\text{Strength(category Z}|\text{Year Y)} = \sum_{z \in Z, d \in D_Y} \theta_{d,z} \tag{5.9}$$

As shown in Figure 5.2, we can see that the politics has always been a hot topic over the years, while for the research paper, the object detection dominates the early years of research.

**Table 5.7:** Topic Category for News by Topic Evolution Model

| Topic Number | Topic | Category |
|---|---|---|
| 1 | Entertainment | Entertainment |
| 2 | Resolution | Politics |
| 9 | Photography | Entertainment |
| 11 | Nasa Space Exploration | Space Exploration |
| 12 | Conflict | Military |
| 13,24,48 | Influenza | Health and Medication |
| 15 | Nuclear | Politics |
| 18 | Health Care | Politics |
| 20 | Scandal | Polictics |
| 22,26,41 | Protest | Politics |
| 25 | Conflict Area | Military |
| 27 | Government | Politics |
| 29 | Finance | Politics |
| 31,33 | Abuse | Politics |
| 32 | Showbiz Event | Entertainment |
| 37 | Music | Entertainment |
| 38 | Celebrities | Entertainment |
| 39 | Politician | Politics |
| 42 | Earthquake | Politics |
| 45 | Medical Support | Health and Medication |
| 46 | Oil Spill | Politics |
| 47 | Council | Politics |
| 49 | Media | Politics |

### 5.8.4 Topic Transition

In this section, we will present the transition matrix in the form of transition graph. The transition graph shows the top seven news topics of year 2009 and five topics of literature corpus for the year 2001. The topic transition each year here is defined as the accumulation of document-specific topic transition over all the documents of that specific year. The topic transition obtained from the model is thus illustrated in figure 5.3. Here, different color refers to different category, and the direction of the arrow denotes the direction of topic transition. The number on the arrows shows the transition probability. The most probable transitions with probability higher than 0.02 are presented in bold.

We could see for the news topic transition, the most probable topical transition is from council (topic 47) to conflict area (topic 25). This means over the year 2009, the topic ‘council’ is quite related to the topic ‘conflict area’, and the ‘conflict area’ is mostly arisen from the topic ‘council’, which is in consistent with our understanding about the Gaza War. Another topic pair that has strong correlation is ‘politician’ (topic 39) and

**Table 5.8:** Topic Category for the Literature by Topic Evolution Model

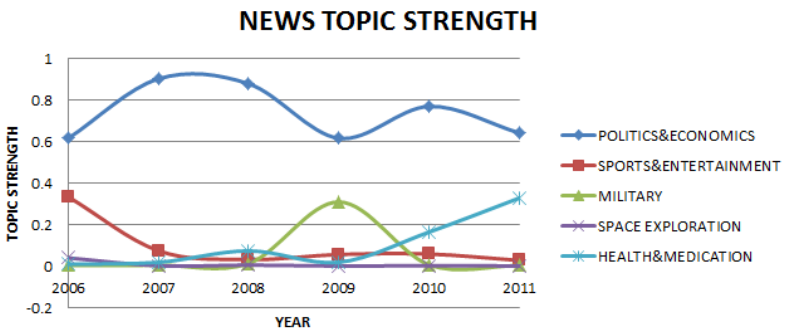
| Topic Number | Topic | Category |
|---|---|---|
| 1 | Interface | Human Computer Interaction |
| 2 | Concepts Reasoning | Modeling and Algorithm |
| 3 | User Scheme | HCI |
| 6 | Experiment | Learning |
| 7 | Estimation | Learning |
| 8 | Technique | Software Engineering |
| 10 | Detection | Object Detection |
| 11 | Network Security | Network |
| 15 | Learning | Learning |
| 21 | Web Service | Network |
| 25 | Semantic Mining | Algorithm and Modeling |
| 30 | Logic Architecture | Architecture |
| 31 | Model | Algorithm and Modeling |
| 32 | Motion Control | Object Detection |
| 36 | Pattern Recognition | Algorithm and Modeling |
| 37 | Project Management | Engineering |
| 39,40 | Distributed Computing | Engineering |
| 41 | Location sensing | Object Detection |
| 42 | Integration | HCI |
| 43 | Web Server/Clinet | Network |
| 45 | Security Management | Engineering |
| 48 | User Experience | HCI |

'health care' (topic 18). The direction from 'politician' to 'healthcare' show the politicians may have dispute over the 'health care' policy.

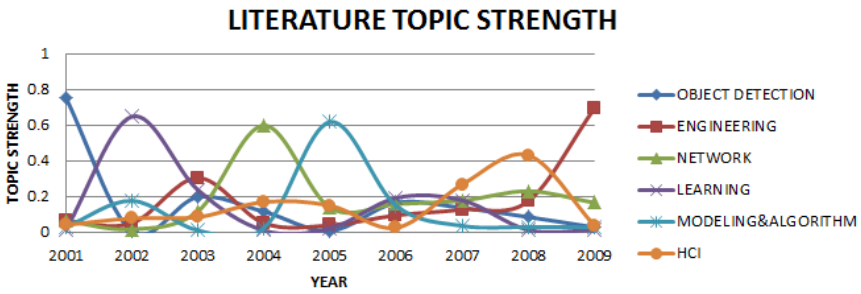
For the literature topic transition, 'interface' (topic 1) has high transition probability to both Experiment (topic 6) and Web Service (topic 21). We can see that the study of 'user interface design' are highly related to 'network' and 'model learning', indicating 'user interface' is a highly interdisciplinary subject. The topic transition graph gives us a better idea of the topic relation and the strength of the relatedness.

### 5.8.5 Topic Evolution

With the information about the topic strength and topic transition, we now can better explain the whole corpus through the topic evolution graph here in Figure 5.4. For each year, the top seven topics for the news and top five topics for the literature are selected and demonstrated. Only the transitions with probability higher than 0.02 are shown in the graph. Also, the relatedness of the topics from different years are calculated using KL distance, each topic being the word distribution. For two topics with probability distribution $z_i$(w) and $z_j$(w),




(a). News topic evolution for Separate Model




(b). Literature topic evolution for Separate Model

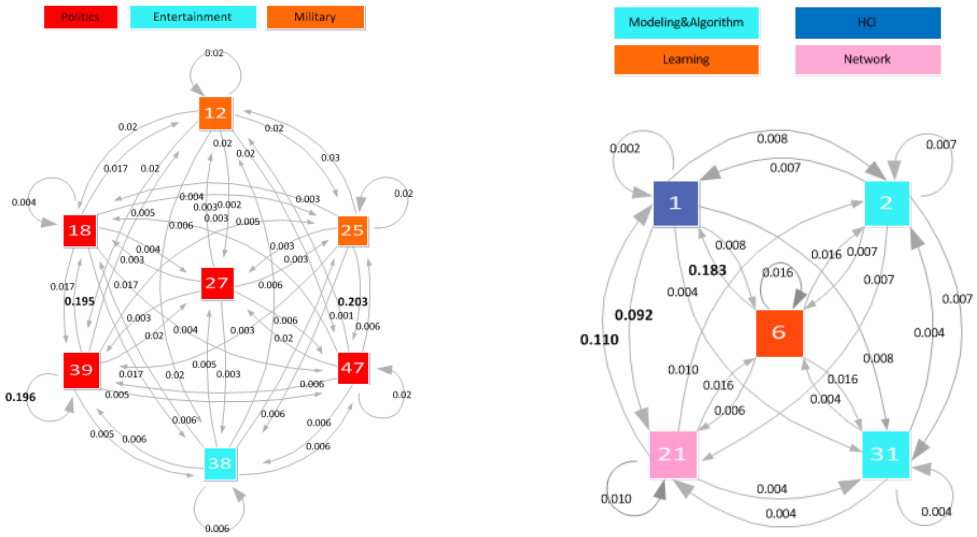
**Figure 5.2:** Topic transition

the KL distance is calculated as:

$$\begin{aligned} &\text{KL-Distance}(\text{topic}z_i,\text{topic}z_j) \\ &= \sum_{w\in V} z_i(w)\log\frac{z_i(w)}{z_j(w)} + \sum_{w\in V} z_j(w)\log\frac{z_j(w)}{z_i(w)} \end{aligned} \tag{5.10}$$

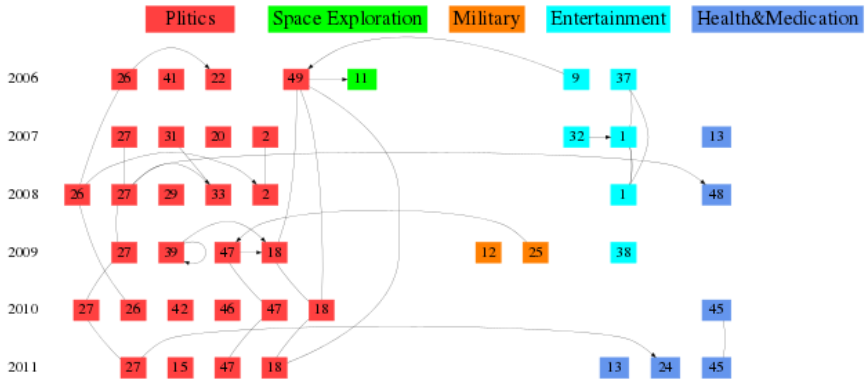
Thus, we can see the evolvement through the years.

The arrows between topics within one year tell the prominent topic transition with probability higher than 0.02. The links between topics of different year indicate the topic corelatedness. The two topics linked together over different years are either the same topic or the topics with high similarity measured by the KL distance. We can see topic 27 'government' stays as a stable one all through the years. It confirms with our common sense that the government action is always a focus of the public. From year 2009 to year 2011, the topic 18 'healthcare' stays the top topic. Also, we could see that the topic 11 'nasa space exploration' is the hottest topic for 2006, the 46th topic 'oil spill' is the news for 2010. Both the topic 31 and topic 33 have the very similar topic 'abuse'.

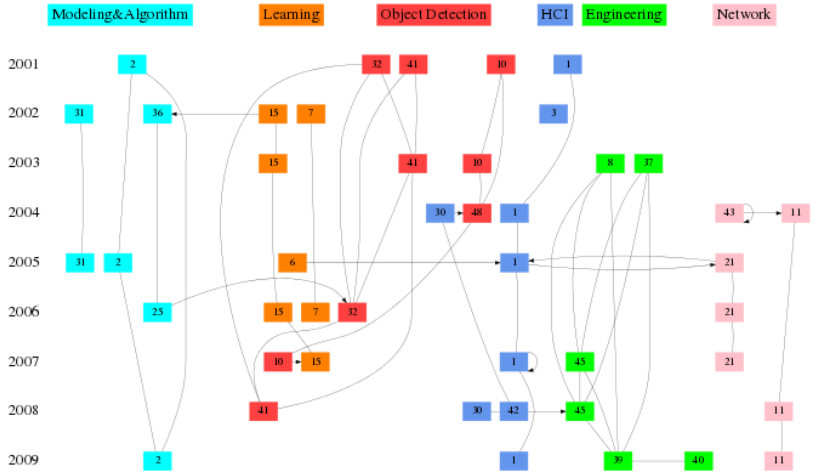



(a). News topic transition. (b). Literature topic transition.

**Figure 5.3:** Topic transition




(a). News topic evolution for Separate Model.




(b). Literature topic evolution for Separate Model.

**Figure 5.4:** Topic transition

**Table 5.9:** Overall Perplexity by Topic Evolution Model

| Methods | LDA | | CTM | | PTM-1 | | PTM-2 | |
|---|---|---|---|---|---|---|---|---|
| Topics | News | Paper | News | Paper | News | Paper | News | Paper |
| 50. | 1550.42 | 1743.49 | 1449.05 | 1239.35 | 1128.99 | 1424.97 | 117.18 | 154.13 |
| 60. | 1439.48 | 1678.41 | 1319.57 | 1232.74 | 1103.11 | 1389.76 | 115.18 | 150.18 |
| 70. | 1370.46 | 1583.18 | 1311.43 | 1235.21 | 1041.18 | 1339.36 | 114.69 | 146.90 |
| 80. | 1361.96 | 1512.24 | 1295.89 | 1241.40 | 983.92 | 1286.04 | 110.42 | 143.91 |
| 90. | 1354.97 | 1458.33 | 1288.68 | 1235.21 | 919.894 | 1236.77 | 108.35 | 141.18 |
| 100. | 1310.83 | 1409.27 | 1288.19 | 1238.92 | 893.29 | 1204.46 | 103.34 | 138.54 |

Also, from the literature topic evolution graph, we could see that motion control (topic 32), location sensing(topic 41) and detection (topic 10) respectively dominate the early years of research. The implication is in consistent with our corpus distribution, since the main conferences in the year 2001 are 'Agents' and 'ASP-DAC'. Also, the evolution graph shows the recent trend in the network area, including 'web server/client' (topic 43), 'web service' (topic 21) and 'network security' (topic11).

Thus, from the topic evolution graph, we can see the whole picture of a collection of documents, including the dying and emerging of topics. The dynamic change of the topic evolution gives us a lot of insight into the topic structure of the corpus and thus a better and full understanding of the whole corpus.

### 5.8.6 Perplexity

Finally, we compare the language perplexity for our two methods against both LDA and CTM in Table 9.

Obviously, our PTMs, especially the PTM-2, provide a better fit than traditional topic models. The comparison between PTM-1 and PTM-2 shows that the use of word pairs as text units is not as expressive as the use of words as text units. That is, the dependency of the words within a word pair should be modeled explicitly.

## 5.9 Conclusions & Future Work

In this chapter, we proposed a new way to explore the topic transition and evolution. A more expressive probabilistic pairwise topic model is proposed to facilitate identifying the topics and further capturing the topic transition and evolution. Instead of assuming the topic dependency in the sequential text units, we show the extracted word pairs have more topical correlation and thus a better choice of where to mine the topic transition.

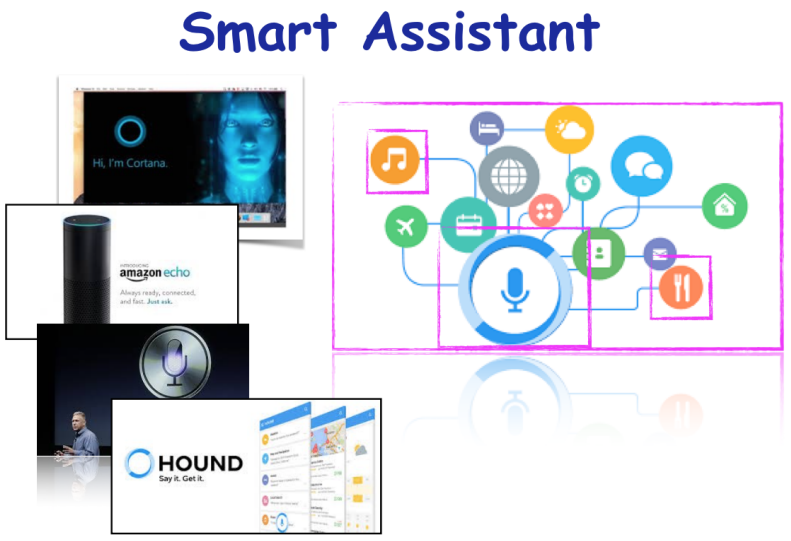
Both the pairwise topic models proposed (PTM-1 and PTM-2) have more expressive power than the traditional topic models, both empirically and quantitatively. We found our model was competitive in identifying meaningful topics and the topic words obtained were more distinctive. Further, we find the modeling of complete dependency between both the words and topics are more expressive than just modeling the dependency between topics.

Overall, our models provide a better way to represent the document so as to model the topic relatedness.

Although the topic modeling through word pairs has been proved to be effective, the model is too complex to be scaled up. We need to simplify the model to lower the time and space complexity. Therefore, the future efforts will focus on the simplification of the model and its scalability.

Also, the pairwise topic model can be extended to find the relation between heterogeneous data, such as the relation between image pixels and text words. Its usage in other types of data needs further exploration.

# Chapter 6: Spoken Language Analysis




With the rampant usage of smart phones, the processing and further understanding of user spoken language become an essential part to build the engine for intelligent assistant. Different from traditional natural language understanding, the aim of spoken language understanding is for the phone to understand the human spoken language and provide corresponding services to the user. For example, if you want your smart phone to guide you to a Starbucks in San Jose, you may say 'Navigate me to Starbucks in San Jose.'. The smart phone is expected to open up the google map service with 'Starbucks' as destination. To provide the corresponding service, the smart phone needs to know the user's intention and the related parameters for the intention. In this case, the user intention is 'navigation' and the parameter is 'Starbucks' as 'destination'.

Therefore, Spoken Language Understanding (SLU) is the ability to process what a user says and figure out how it maps to actions the user intents. The SLU result can then be passed to an application that takes the appropriate action. The main task of spoken language understanding is to map the words the users say to desired actions supported by your application.

The understanding of spoken language then is to extract its intention and the related parameters. For the aforementioned example, given the user spoken language 'Navigate me to Starbucks', the spoken language

understanding engine is to output the intent as ‘navigate’ and the parameter ‘Starbucks’ as ‘destination’.

Although the related parameters are different from ‘entity’ in general sense, we refer to the parameters in this thesis as the ‘entity’. Different intentions are related to different entity types. The ‘navigation’ intention is related to entity type ‘destination’, while the ‘music’ intention relates to entity type ’song’, ‘artist’, ‘genre’ and etc..

Therefore, the function of spoken language is quite restricted compared to natural language.

To better understand the spoken language, we need first to examine its characteristics and how it is different from normal texts. Figure 6.1 and Figure 6.2 show the clips for the sample sentences from spoken languages.

| ID | Parking |
|---|---|
| 104 | get me a parking spot for less than 20 |
| 105 | find me affordable places where i can park |
| 106 | find me parking on hollywood blvd in downtown |
| 107 | parking near american diner |
| 108 | where can i park overnight |
| 109 | take me to the cheapest parking garage |
| 110 | find me the garage nearest to where i am |
| 111 | are there any places to park downtown under 20 |
| 112 | show me the map of parking at oklahoma university |
| 113 | lowest price parking |
| 114 | navigate me to the cheapest parking |
| 115 | locate nearest parking garage to lake theater |
| 116 | find parking near merchandise mart |
| 117 | where can i park for under 500 |
| 118 | does the parking garage stay open from midnight to 2 am |
| 119 | i would like to find a parking service that does not charge more than five dollars |
| 120 | are there any other lots near the stadium |

**Figure 6.1:** Data for Parking

From the observation, we see the syntactic structure corresponds to its semantic structure. That is, the spoken language also falls into two parts of intent expression and entity expression syntactically. For example, for the sentence with id number 106 ‘find me parking on hollywood blvd in downtown’, the ‘find me parking’ is the expression to show the intent, while ‘hollywood blvd’ is its related entity.

To verify our assumption, we manually label the 1477 sentences from ‘parking’ and 1488 sentences from ‘music’ with different entity types. The sample annotations are shown in Figure 6.3-6.4.

First, we examine the number of words that are in intent phrases. The intent phrases present most of the time as repeated patterns. Therefore, we count the number of repeated patterns exceeding a certain threshold as the intent phrases. Since most of the repeated one or two words phrases consist of stop words, we count

| ID | Music |
|---|---|
| 401 | songs by britney spears |
| 402 | album by green day |
| 403 | lets play jason derulo in my head |
| 404 | listen to john mayer |
| 405 | listen to new songs by drake |
| 406 | play u2s albums |
| 407 | music by jennifer lopez |
| 408 | music by michael jackson |
| 409 | music by tom petty |
| 410 | music from maroon five |
| 411 | play a song by celine dion |
| 412 | play adele latest album |
| 413 | play fall out boys |
| 414 | play john denver |
| 415 | play some smashing pumpkins |
| 416 | put on madonna first album |

**Figure 6.2:** Data for Music

| ID | Sentences | Parking_Type | Calendarx | Location | Location_Reference | Distance_Reference | Distance_Number | Distance_Unit | Price_Reference |
|---|---|---|---|---|---|---|---|---|---|
| 1152 | street parking close to my work | street | | | my work | | | | |
| 1153 | street parking near diner | street | | diner | | | | | |
| 1154 | take me somewhere to park within view | | | | | | | | |
| 1155 | take me somewhere with super cheap parking | | | | | | | | cheap |
| 1156 | take me to a garage within 5 miles | garage | | | | | 5 | miles | |
| 1157 | take me to a parking garage | garage | | | | | | | |
| 1158 | take me to a parking garage that is closest to the space needle in seattle | garage | | space needle in seattle | | closest | | | |
| 1159 | take me to a place with parking under 10 | | | | | | | | |
| 1160 | take me to a parking lot with really cheap parking | lot | | | | | | | cheap |

**Figure 6.3:** Ground Truth Building for Parking Data

| ID | Sentences | title | artist | album | genre | playlist |
|---|---|---|---|---|---|---|
| 1004 | play drive to work playlist | | | | | drive to work |
| 1005 | can i hear bad blood | bad blood | | | | |
| 1006 | play that trap house album | | | trap house | | |
| 1007 | start taylor swifts music | | taylor swift | | | |
| 1008 | what type of jazz music do you have | | | | jazz | |
| 1009 | play folk music | | | | folk | |
| 1010 | let me hear the whole follow the leader album | | | follow the leader | | |
| 1011 | play red by taylor swift | red | taylor swift | | | |
| 1012 | play nine inch nails | | nine inch nails | | | |
| 1013 | play in utero by nirvana | | nirvana | in utero | | |

**Figure 6.4:** Ground Truth Building for Music Data

only the patterns with length longer than 3. Table 6.1 and 6.2 show the pattern statistics for both the normal and spoken language.

From the above observation, we can see although spoken language is much shorter in length than normal

**Table 6.1:** Pattern Statistics for both Normal & Spoken Language I

| | **Normal Text** | **Spoken Language** | |
|---|---|---|---|
| | News Text | Parking | Music |
| **# of sentences** | 1400 | 1447 | 1479 |
| **# of words** | 29502 | 11453 | 7731 |
| **# average length of sentence (words)** | 21 | 8 | 5 |

**Table 6.2:** Pattern Statistics for both Normal & Spoken Language II

| | **Normal Text** | **Spoken Language** | |
|---|---|---|---|
| Threshold | News Text | Parking | Music |
| 30 | 114(0.38%) | 1502(13.1%%) | 582(7.5%) |
| 25 | 114(0.38%) | 1967(17.1%) | 582(7.5%) |
| 20 | 114(0.38%) | 2432(21.2%) | 708(9.15%) |
| 15 | 213(0.72%) | 2836(24.8%) | 900(12%) |
| 10 | 384(1.3%) | 3736(32.6%) | 1137(14.7%) |

text, it has much more repeated patterns.

Second, we examine the distribution of entities in spoken language compared to normal texts. The statistics are shown in Table 6.3.

**Table 6.3:** Entity Statistics for both Normal & Spoken Language

| | **Normal Text** | **Spoken Language** | | | |
|---|---|---|---|---|---|
| | Normal Text | Parking | Music | Message | Call |
| # of words | 29502 | 11453 | 7731 | 1209 | 576 |
| # of entities | 856 | 2330 | 1750 | 255 | 156 |
| percentage | 2.9% | 19.4% | 22.6% | 21% | 27% |

The statistics shows entities are a great part for spoken language than for normal texts.

Therefore, the spoken language is different from normal text from the following perspectives.

First, semantically, spoken language consists of two parts to show the intention and the related information. The semantic structure of spoken language is different from normal texts, due to different purposes that different normal texts and spoken language serve. The functions for spoken language are more restricted to providing the service, while the purpose of the normal texts are countless.

Second, syntactic structure for spoken language corresponds to its semantic structure, and the intention phrases and the related information phrases are separated according to its semantic structure.

Therefore, spoken language is a subset of natural language both semantically and syntactically. In the following chapters, we will formally define the spoken language as the intent specific sub-language, and design a chunker to parse the spoken language into two parts: intent phrases and related information phrases. Finally, we develop a statistical method to classify the intent phrases and entity phrases into different topics.

## 6.1 Intent Specific Sub-language

In this section, we will define the intent specific sub-language in detail.

Let $L$ be the set of all natural language. For each natural language $l \in L$, $P(l)$ is the power set of $l$, representing all possible language clips for $l$. Then $P(L) = \{p(l)|p(l) \in L\}$ represents all possible natural language clips.

For example, for one sentence $l =$ 'Take me to Starbucks', all its possible language clips are 'Take me to Starbucks', 'Take me to', 'me to Starbucks', 'Take me', 'me to', 'to Starbucks', 'Take', 'me', 'to', 'Starbucks'.

For the intent specific sub-language, assume the intent sets are $D$, and the number of intents is $|D|$. For each intent $d \in D$, there are $n$ related entity types $\{d_i|i \in \{1, 2, ...n\}\}$. For each entity type , we denote all possible expressions for that entity type as $E_{d_i}$ and the overall entity expressions can be denoted by $E_d = \{E_{d_i}|i \in \{1, 2, ...d_n\}\}$. Similarly, we denote the set of intention expressions as $I^d$.

For example, for the parking intent, $d$ = parking, the related entity types may include 'parking type', 'location', 'time', 'price'. Then $E_{parking\,type}$ may include {street, off street, on street, valet, ...}, and $E_{location}$ may be {Starbucks, San Jose, airport, restaurant, ...}.

Thus, for the intent $d \in D$, we have $|E^d|$ entity interpreters $\{\boldsymbol{E}_{d_i}|(i \in \{0, 1, 2, ..., |E^d|\})\}$ and one intention interpreter $I$. Given a spoken language $q \in P(L)$, we have

$\boldsymbol{E}^d_{E_{d_i}}(q) = \cup_s s(s \in q \text{ and } s \in E_{d_i}))(i \in \{0, 1, 2, ..., |E^d|\})$

$\boldsymbol{I}^d(q) = \cup_s s(s \in q \text{ and } s \in I_d))$

For example, for the user request 'find me a parking near San Jose around 5pm',

We have $q$ = 'find me a valet parking near San Jose airport at 7pm', to implement $\boldsymbol{E}$ and $\boldsymbol{I}$ to the query, we have

$\boldsymbol{E}^{parking}_{parking\,type}(q) = \{valet\}$

$\boldsymbol{E}^{parking}_{location}(q) = \{San\ Jose\ airport\}$

$\boldsymbol{E}^{parking}_{time}(q) = \{5pm\}$

$\boldsymbol{E}^{parking}(q) = \{\{valet\}_{type}, \{San\ Jose\ airport\}_{location}, \{5pm\}_{time}\}$

and $\boldsymbol{I}^{parking}(q) = \{find\ me\ a\}$

Therefore, we define the sub-language for each intent as

$S^d(q) = \cup_{l_i \in q} \boldsymbol{E}^d(l_i) \cup \imath^d(l_i \in P(L))$

The 'parking' intent specific sub-language for $q$ then is:

$SL(q) = \{\boldsymbol{I}^d_q : \boldsymbol{E}^d_q\} = \{\{find\ me\ a\ parking\}, \{\{valet\}_{parking\ type}, \{San\ Jose\ airport\}_{location}, \{5pm\}_{time}\}$

As we can see from the definition, sub-language is a intent specific formalism. It is to interpret the user intention from different intent perspectives. Different from other formalism to find the underlying semantic representation through syntax features, the sub-language is to interpret user intention from intent specific entities and expressions.

The word clips do not have to have the semantic meaning or syntactic structure to form the sub-language. For the intent specific expressions, they are not necessarily to be grammatically correct. From the aforementioned example, in the user request 'Navigate me to Starbucks', the 'Navigate me to' may not be grammatically segmented into one group, but it is really important feature to express 'navigate' intention.

## 6.2 Problem Formulation

Having defined the intent specific sub-language, we now formally define our sub-language extraction problem as the extraction of sub-language for a given a spoken language $q$.

## 6.3 Spoken Language Processing Pipeline

Since the spoken language is a subset of natural language, the processing for spoken language should be different for spoken language than for natural language.

For the natural language, the pipeline generally includes text segmentation, tokenization, Part-Of-Speech Tagging and syntactic parsing. The text segmentation is for the language rather than English to get the word segmentation, the second part is the tokenization to chop the sentences into pieces and perhaps throw away certain words. The part-of-speech tagger and syntactic parsing are the intermediate stages for further tasks, such as entity extraction, text summarization, and topic extraction, etc.

Since spoken language is a subset of natural language, the processing pipeline is tailored to its specific characteristics. Figure 6.5 shows the simplified processing pipeline for spoken language.

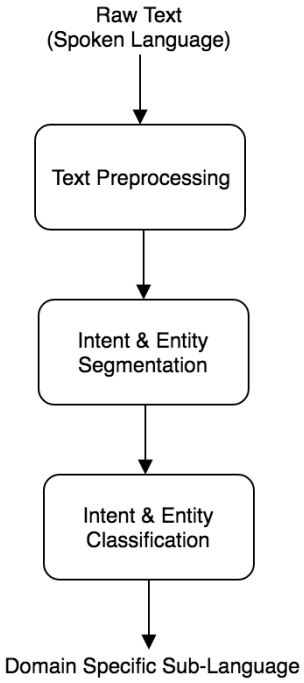



**Figure 6.5:** Spoken Language Processing Pipeline

There are mainly three components in this pipeline:

•Text preprocessing. The text is normalized in this part, removing the punctuation. Stop words are not removed here. The stop words sometimes play semantic or syntactic roles in such short texts.

•Intent and Entity Segmentation. In this part, the intent and entity phrase will be separated.

•Intent and Entity Classification. The segmented parts are labeled with intent and entity types. The output should be the intent specific sub-langauge.

In the following two chapters, we will focus on intent and entity segmentation, and intent and entity classification. In Chapter 7, we will further examine some characteristics helping us to segment the spoken language text and show the characteristics of spoken language can greatly improve the segmentation performance. In Chapter 8, we will introduce a statistical modeling method to classify the intents and entities.

# Chapter 7: Intent and Entity Phrases Segmentation

From previous chapter, we learn that spoken language consists of intent and entity phrases. In this chapter, we will explore the spoken language further to segment the sentences into the intention and entity phrases.

There are three prominent characteristics of spoken language to help segment the intention and entity.

a. More entities are preceded by preposition in spoken language than in normal texts. Table 7.1 shows entities take a larger portion in spoken language than in normal language. Therefore, for spoken language, we can use preposition as segmenters to separate the intent phrase from the entity phrases.

**Table 7.1:** Percentage the entities preceded by preposition for both Normal & Spoken Language

| | Normal Text | Parking | Music | Message | Call |
|---|---|---|---|---|---|
| # of sentences | 1400 | 1447 | 1479 | 200 | 100 |
| # of preposition | 856 | 921 | 591 | 105 | 41 |
| # of entities | 4460 | 2330 | 1750 | 255 | 156 |
| percentage | 19% | 40% | 34% | 41% | 26% |

b. There are mainly two types of entities, entities that are relatively fixed, and entities that have large variation. For example, in ‘parking’ domain, the entities in ‘parking type’ has a relatively restricted vocabulary set, whereas the entities in ‘location’ are much more free formed texts with large vocabularies.

Therefore, we can use the entity types that have limited variations as prior knowledge to segment spoken language text. Assume we have a user request as ‘could you find me a valet parking in san jose?’. The preposition can help separate the sentence into two chunks ‘could you find me a valet parking’ and ‘san jose’. In addition, the parking type ‘valet parking’ can help separate the sentence into three chunks: ‘could you find me a’, ‘valet parking’ and ‘san jose’.

c. The repeated phrases are mostly intent phrases and thus can also be treated as segmenters. For example, in ‘music’ domain, for the sentence ‘play the song yellow by coldboy.’, the yellow is separated by the repeated phrase ‘play the song’ and the preposition ‘by’.

Therefore, we use three types of segmenters: preposition, restricted entities, and patterns to separate the spoken language texts. The preposition and restricted entities are easy to obtain, but the pattern extraction

toolkit is not readily available. Section 7.1 will introduce the pattern mining method exclusively for text mining. In Section 7.2, we will show the segmentation algorithm and the result.

## 7.1 Semantic Pattern Mining

Pattern mining is an important data mining problem with broad applications. Multiply studies have been proposed for mining interesting patterns in transaction database, such as frequent pattern mining Agrawal et al. (1993), Agrawal et al. (1994)Mannila et al. (1994) Agarwal et al. (2001),Pei et al. (2001), closed pattern miningPasquier et al. (1999)Liu et al. (2003), and maximal pattern mining Bayardo Jr (1998). The patterns can be item sets, or item sequencesAgrawal & Srikant (1995).

Frequent pattern mining is the most basic pattern mining technique. Frequent patterns are patterns that appear in a data set with frequency no less than a user specified threshold. For example, beer and diaper may appear frequently together in a transaction data set to form a frequent pattern {beer, diaper}. However, frequent pattern mining can yield many redundant patterns. For example, in a transaction dataset, if both {beer} and {beer, diaper} appear 200 times, it means beer is bought with diaper all the time, and we have no need to keep {beer} in our frequent pattern set.

Therefore, closed pattern is proposed. Closed patterns are frequent patterns with no longer frequent patterns having the same frequency. From the aforementioned example, {beer} as a subset of {beer, diaper} is excluded from the closed pattern. Closed pattern mining is a more compact and lossless representation for frequent patterns.

Obviously, both the frequent and closed pattern mining techniques are not suitable for text mining, since they ignore both the order and word adjacency which are very important for text mining.

The sequential pattern mining tries to mine the frequently occurring ordered items or subsequences. For example, transaction records Han & Kamber (2006) may show people buy PC, then a digital camera, and then a memory card in a sequence frequently. Thus, {PC, digital camera, memory card} forms a frequent sequential pattern. Although sequential patterns are patterns with sequential order, the items in sequential order don't need to be contiguous. Another variation of pattern mining is contiguous sequential pattern mining. The contiguous constraint requires that the items should not only be in sequential order, but should also be contiguous.

Although the contiguous sequential pattern mining takes into consideration of the word order and word contiguity, for the pattern mining techniques to be used in text mining, the existing algorithms simply treat each word as an item without considering the word difference in different word surroundings. This is true for transaction data, since the item is identified by its name only. The 'cheese cake' in one transaction represents the same thing as in another transaction. However, human interpretation of texts relies on inherent grouping of terms, and a word in one word grouping may be quite different from the same word in a different word grouping. Therefore, a word itself can not identify its meaning unless we put it into its surrounding words.

For example, we have a corpus of movie reviews. Lots of reviews are talking about 'american pie', and some noisy sentences, such as 'I love eating pie' or 'pie is tasty' are also included in the collection. Let the frequency of 'american pie' and 'pie' be 200 and 202 respectively. Given the threshold to be 100, we would find 'pie' to be one closed pattern, since 'pie' appears 202 times. However, for 200 times, 'pie' together with 'american' forms a coherent semantic grouping, while 'pie' itself as a meaningful word grouping only appears twice. Therefore, the frequency of 'pie' should be 2, and be excluded from the closed patterns.

Therefore, the lossless representative pattern mining for text mining is quite different from closed pattern mining for transaction data. For closed pattern mining in transaction data, individual item in each transaction can be put into different patterns and contribute to the counts of frequency for many patterns. For example, in a transaction {beer, diaper}, the item 'beer' can be included into both {beer} and {beer, diaper} and contribute to the counts of both the item set {beer} and {beer, diaper}. However, in text mining, the word 'pie' in sentence 'american pie is my favorite movie' should be counted only into the frequency of 'american pie', but not 'pie'. That means, the individual word in a short text should only belong to one valid pattern and contribute to one count for that specific valid pattern.

Therefore, when mining the lossless representative patterns in text, instead of just simply counting the number of each pattern for each transaction, we need to find the valid pattern for each text and only add count to the frequency of the valid pattern. Therefore, this is not a trivial problem and worth investigation.

In this paper, we will propose the problem and its solution. The novelties of this paper are as follows:

•We introduce a new concept 'semantic pattern' to include both the pattern and its position information to form a two dimensional pattern identification. Based on it, we propose a theoretical framework, and examine

its properties.

•We formulate the problem of frequent semantic pattern mining based on the proposed framework. The frequent semantic pattern mining leverages the two dimensional information to find the compact and lossless representative patterns for text mining.

•We solve the frequent semantic pattern mining problem from a novel perspective via Suffix Array, which is a perfect fit for our theoretical framework. Although semantic patterns are designed to be more complicated, we prove that our algorithm can scale up linearly.

The section is organized as follows. Section 7.1.1 gives a review about the related work. In Section 7.1.2, we introduce the basic concepts. Section 7.1.3 introduces the new concept of semantic pattern and propose the problem of frequent semantic pattern mining. Section 7.1.4 formally define the problem. From section 7.1.5 to section 7.1.8, we examine in detail the algorithms to extract the frequent semantic patterns. Section 7.1.9 to section 7.1.13 are about the experiments and evaluation. Finally, we conclude our study in Section 7.1.14.

### 7.1.1 Related Work

Pattern mining as a traditional data mining technique has been studied over the years. Basic pattern mining techniques include frequent, closed and maximal pattern mining. Frequent pattern mining has no restriction that

The two basic approaches to solve the pattern mining problems are: Apriori-based approach Inokuchi et al. (2000) Kuramochi & Karypis (2001) Vanetik et al. (2002) and pattern-growth approach, Borgelt & Berthold (2002) Huan et al. (2004). Recent years have seen the explosive increase of interest in the sequential pattern mining Srikant & Agrawal (1996), Zaki (2001), Pei et al. (2004),especially for the closed sequential pattern mining Yan et al. (2003), Wang & Han (2004), Gomariz et al. (2013), Fournier-Viger et al. (2014). ZhangZhang et al. (2015) proposed a continuous sequential mining problem and solved it with CCspan algorithm. The new CCspan solved the scalability problem and obtained more compact yet lossless patterns.

Although lots of efforts have been made to extend the existing pattern mining algorithms, all of them ignore the difference between words and items. A new framework considering the context information is proposed to solve this problem.

### 7.1.2 Basic Concepts

In this section, we review some basic concepts in pattern mining and introduce the new concept of semantic pattern.

**Definition 1** (Continuous Sequential Pattern (P)). *Assume we have a word sequence $s = \{w_1, w_2, ..., w_n\}$, we identify each word in sequence by its index: $s[i] = w_i (i \in \{1, 2, ...n\})$, and each sub sequence of $s$ as $s[i : j] = \{w_i, w_{i+1}, ...w_j\} (i < j,$ and $i, j \in \{1, 2, ...n\})$.We define the continuous sequential pattern as the contiguous sequence of j terms $p = s[i : i + j - 1]$. For any sequential patterns $p_1, p_2$, if$p_1 \subseteq p_2$, $p_2$ is the super pattern for $p_1$, and $p_1$ is the sub pattern for $p_2$. Given a corpus $S$, we denote all the continuous patterns in the corpus as $P(S)$.*

The patterns discussed in this paper are all the continuous sequential patterns. Thus, we just use pattern to mean continuous sequential pattern.

**Definition 2** (Support of a Pattern). *For a collection of word sequences $S = \{s_1, s_2, ..s_n\}$, we define the support of a pattern here as the number of word sequences containing the pattern.*

**Definition 3** (Frequent Pattern (FP)). *For a collection of word sequences $S = \{s_1, s_2, ..s_n\}$, given a threshold $\lambda$, the frequent patterns are those with the support larger than $\lambda$. We denote all the frequent patterns in $S$ as $FP(S)$.*

**Definition 4** (Closed Pattern (CP)). *The closed patterns are patterns with no super pattern that has the same support.*

Assume we have a four word sequences collection $S$ as:

s[0]: american pie gave me such false hope for women

s[1]: american pie will forever be my favorite

s[2]: the american pie cast look like now

s[3]: easy as pie

Given the threshold $\lambda = 3$, we have frequent patterns $FP(S)$ as {'america pie','pie','american'}. and closed patterns $CP(S)$ as {'america pie','pie',}. Closed patterns are more compact representation than

frequent patterns. However, there are still some inappropriate pattern such as 'pie'. Although it appears four times, it appears three times in 'american pie'. Since 'pie' in 'american pie' has different meaning as in 'pie' in s[3], the 'semantic support' of 'pie' actually should be 1 instead of 4.

Therefore, although closed patterns are the lossless representation for frequent patterns for transaction data, they are not necessarily to be the lossless representation for text data. In the next section, we will define the semantic pattern to incorporate the context information to facilitate the exclusion of inappropriate patterns.

### 7.1.3 Semantic Pattern

Since the same word in different short text of different word groupings is quite different from each other, the word itself can not fully identify its meaning. We need to identify a word also by it position. Therefore, we combine the word and its position together in defining the meaning of a pattern.

**Definition 5** (Semantic Pattern (SP)). *Assume we have a collection of word sequences $S = \{s_1, s_2, ..., s_n\}$. For a pattern $p \in P(S)$, we define a semantic pattern as $SP = p : (d, s)$, where $(d, s)$ are two dimensional position information ($d$ is the word sequence id, and $s$ is the starting position of pattern $p$ in word sequence $d$). We denote the pattern of $SP$ as $SP_{pattern}$ and position of $SP$as $SP_{position}$, we have $SP_{pattern} = p$ and $SP_{position} = (d, s)$. For any two semantic patterns $SP_1 = p_1 : (d_1, s_1)$ and $SP_2 = p_2 : (d_2, s_2)$, the length of $p_1$ is $l(p_1)$ and the length of $p_2$ is $l(p_2)$, we define $SP_1 \subseteq SP_2$ if $p_1 \subseteq p_2$, and $d_1 = d_2, s_1 + l(p_1) \leqslant s_2 + l(p_2)$. We denote all the semantic patterns in $S$ as $SP(S)$.*

To keep the example simple, we use two word sequences to illustrate our new concepts. Assume we have a collection of two word sequences:

s[0]: american pie

s[1]: pie

The semantic pattern for 'american pie' in the first sentence is 'american pie: (0, 0)', meaning it appears in $s[0]$ and starts at position $0$. 'pie: (0,1)' is the sub pattern of 'american pie: (0,0)'. and $SP(S)$ = {american pie: (0,0), american:(0,0), pie: (0,1), pie:(1,0)}

Therefore, we can identify the pattern by the words it contains and its position.

**Definition 6** (Semantic Pattern Collection). *Assume we have a collection of semantic patterns $C = \{SP_1, ...SP_n\}$,*

*we denote the pattern collection for all semantic patterns C as $C_{pattern}$, we have:$C_{pattern} = \cup_{i=1}^{n} SP_{ipattern}$.For any $p \in C_{pattern}$, we define the semantic pattern collection over one pattern $p(p \in C_{pattern})$ as $C(p) = \cup SP_i(SP_{ipattern} = p)$. We denote the semantic pattern collection over patterns $P(P \subseteq C_{pattern})$ as $C(P) = \cup_{p \in P} C(p)$.*

Continuing with the previous example, the semantic pattern collection for $SP(S)$ over 'american pie' is $SP(S)$(american pie)={american pie:(0,0)}, and semantic pattern collection over 'pie' is $SP(S)$(pie)={pie: (0,1), pie:(1,0)}. The $SP(S)$ = {american pie:(0,0), pie: (0,1), pie:(1,0)}

**Definition 7** (Exclusive Semantic Pattern Collection (ESPC))**.** *Assume we have a collection of semantic patterns $C = \{SP_1, ...SP_n\}$, we define its exclusive semantic patterns over C as: $ESPC(C) = \cup SP(SP \in C$ and $\nexists SP' \in C$ that $SP \subset SP')$.*

We have $ESPC(SP(S))$={american pie: (0,0)}, since both 'pie: (0,1)' and 'pie:(1,0)' in $SP(S)$ are the sub patterns of 'american pie: (0,0)'.

**Definition 8** (Semantic Pattern Collection Combination (SPC))**.** *Assume we have a collection of semantic patterns $C = \{SP_1, ...SP_n\}$. For each $p \in C_{pattern}$, we denote all its position collection as $C(p)_{position} = \cup SP_{position}(SP_{pattern=p})$. We define the semantic pattern collection combination to combine all the positions of semantic patterns over the same pattern together as $SPC(C) = \cup\, p : C(p)_{position}(p \in C_{pattern})$, we have $SPC(C)_{pattern} = C_{pattern}$. For each $p \in SPC(C)_{pattern}$, we denote: $SPC(C)(p) = p : C(p)_{position}$, thus we have $SPC(C)(p)_{pattern} = p, SPC(C)(p)_{position} = C_{position}(p)$ and $SPC(C) = \cup SPC(C)(p)(p \in C_{pattern})$. Further, we denote the number of semantic patterns for each pattern $p \in SPC(C)_{pattern}$ as $|SPC(C)(p)|$, we have $\sum_{p \in SPC(C)_{pattern}} |SPC(C)(p)| = n$.*

The semantic pattern collection combination over $SP(S)$ should be: $SPC(SP(S))$ = {american pie:{(0,0)}, american:{(0,0)}, pie: {(0,1),(1,0)}}.

**Definition 9** (Frequent Semantic Pattern (FSP))**.** *Given a collection of word sequences $S = \{s_1, s_2, ...s_n\}$ and threshold $\lambda$, we have all the frequent patterns as $FP(S)$, and all the semantic patterns over $S$ as $SP(S)$. For the semantic pattern collection: $FPC = SP(S)(FP(S))$, we have $ESPC(FPC)$ and its combination as $SPC(ESFP(FPC))$. We define the semantic support of a pattern $p$ over $S$ as the number*

*of semantic patterns for $p$ in $SPC(ESFP(FPC))$. We denote the semantic support of pattern $p$ over $S$ as $\delta(S)(p)$, that is $\delta(S)(p) = |SPC(ESFP(FPC))(p)|$. Further, we define the frequent semantic pattern as: $FSP(S) = \cup SPC(ESFP(FPC))(p)(\delta(S)(p) > \lambda)$.*

To provide a more thorough example of how the frequent semantic patterns can be obtained, we will use four word sequences again. There are mainly four steps to find frequent semantic patterns.

The first step is to find the frequent patterns. We have $FP(S)$=$\{$american pie, american, pie$\}$.

The second step is to calculate the semantic pattern collection over the frequent patterns. $SP(S)(FP(S))$= $\{$american pie: (0,0), american pie: (1,0), american pie: (2,1),american: (0,0),american: (1,0),american: (2,1),pie: (0,1),pie: (1,1),pie: (2,2),pie:(3,2)$\}$. This step is to put all the potential frequent semantic patterns together to facilitate the exclusion of invalid patterns.

The third step is to get the exclusive semantic pattern collection. $ESPC(FPC)$= $\{$american pie: (0,0),american pie:(1,0),american pie:(2,1),pie:(3,2)$\}$. This step helps find valid pattern for one individual word. Obviously, we gives the longer pattern higher priority. As shown in this example, if a word is included in a longer pattern, it would be excluded from the shorter pattern.

The forth step is to combine all semantic patterns with the same pattern together as $SPC(ESPC(FPC))$ =$\{$american pie: $\{$(0,0),(1,0),(2,1)$\}$, pie:$\{$(3,2)$\}\}$. Since $\delta(S)(pie) < 3$, we exclude the semantic patterns with 'pie' as patterns out of the frequent semantic patterns.

Therefore, the frequent semantic patterns over $S$ is $FSP(S)$ = $\{$american pie: $\{$(0,0),(1,0),(2,1)$\}\}$.

We can actually prove theoretically that the pattern collection of frequent semantic patterns is more compact than closed pattern collection.

**Theorem 1.** *Assume we have a collection of semantic patterns $S = \{s_1, s_2, ...s_n\}$, the pattern collection over frequent semantic patterns is the subset of closed pattern, that is: $FSP(S)_{pattern} \subseteq CP(S)$.*

**Proof 1.** *Assume we have $CP(S) \subset FSP(S)_{pattern}$. $\because CP(S) \subset FSP(S)_{pattern}, \therefore \exists p \in FSP_{pattern}(S)$ that $p \notin CP(S)$. $\because p \in FSP(S)_{pattern}, \therefore \nexists p' \in FSP(S)_{pattern}$ that $p \subseteq p'. \therefore \exists p', p \subset p'$, and $\forall SP \in SPCL\ (FSP(S))(p)$ and $\forall SP' \in SPCL(FPS(S))(p')$ that $SP \subset SP'$. $\therefore SP \notin ESPC\ (FSP(S)). \therefore p \notin ESPC(FSP(S))_{pattern}. \therefore p \notin FSP(S)_{pattern}$. which is contradictory to our assumption. $\therefore FSP(S)_{pattern} \subseteq CP(S)$.*

### 7.1.4 Problem Statement

We are given a collection of word sequences $S = s_1, s_2, ..., s_n$. Each word sequence $s_i (i \in 1, 2, ..., n)$ is a non-empty sequence of words. Given a frequency threshold $\lambda$, the semantic pattern mining over $S$ is to find a collection of frequent semantic patterns embedded in $S$.

### 7.1.5 Frequent Semantic Pattern Mining via Suffix Array

Some may wonder how the semantic pattern extraction can achieve linear time complexity, given the cost of pattern mining without context information is already expensive.

We will show in Section 4 that the candidate generation for frequent semantic patterns from suffix array construction can greatly reduce the number of potential frequent semantic patterns and further make the exclusion of the inappropriate semantic patterns much more effective.

The first part of Section 4 introduces the suffix array construction, and second part of Section 4 examines the candidate generation and inappropriate pattern exclusion.

### 7.1.6 Suffix Array Construction

In this section, we will show what is the suffix array and why it can help achieve the linear time complexity in semantic pattern candidate generation.

In essence, the frequent semantic pattern mining is to find the repeated sub word sequences and their two dimensional position information. Suffix Array actually provides us with the perfect solution for both our concerns. Through suffix array, the repeated pattern are naturally grouped together and pattern location information can be easily obtained.

Suffix arrayManber & Myers (1993) is a simple and space efficient alternative to suffix trees McCreight (1976),Ukkonen (1995) . It is widely used index structures on strings and sequences.

**Definition 10** (Suffix Array for a Single String (SA(s))). *A suffix array $SA$ of a long string sequence $S$ of size N, is an array of all N suffixes, sorted alphabetically. A ith largest suffix, is a string that starts at position $SA[i]$ in the sequence and continues to the end of the string sequence.*

For example, the string $S$= 'apple' is indexed as in Table 7.1. Its suffixes are shown in Table 7.2. The suffix array $SA$ containing the starting positions of these sorted suffixes as shown in Table 7.3.

**Table 7.2:** Index for string ‘apple‘

| 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| a | p | p | l | e | $n0$ |

**Table 7.3:** Suffixes before and after Sorting

| Suffix (Unsorted) | i | Suffix (Sorted) | i |
|---|---|---|---|
| apple | 0 | apple | 0 |
| pple | 1 | e | 4 |
| ple | 2 | le | 3 |
| le | 3 | ple | 2 |
| e | 4 | pple | 1 |

There are many interesting linear algorithms to obtain the suffix array, and skew algorithm Kärkkäinen & Sanders (2003) is the most famous. We will use it for our suffix sorting.

Different from traditional suffix array construction, our algorithm is to find the repeated word sub sequence in a collection of word sequences instead of repeated character sequence in a string. Thus, we only sort the suffixes of sub word sequences. We define the $SA$ for word sequences collection as:

**Definition 11** (Suffix Array for a Collection of Word Sequences (SA(S)) )**.** *Given a collection of word sequences $S = \{s_1, s_2, ...s_n\}$, we combine all the word sequences together to form a long sequence $s_1\#s_2\#...\#s_n$ of length N, with a smallest character $\#$ in between word sequences. A suffix array $SA[i] = (d, p)$ of $S$ is an array of all $M(M < N)$ suffixes, starting with a word, sorted alphabetically.The ith largest suffix, is a sub word sequence that is in the $d$th word sequence, starting at position $p$ of the $d$th word sequence and continues to the nearest $\#$.*

Here ‘#’ are special symbols (sentinels) that are different and lexicographically less than other symbols. This is to make sure that the suffixes stop at the end of each corresponding word sequence.

The differences between suffix array of a string and suffix array of word sequence collection are the following:

**Table 7.4:** Suffix Array

| i | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| SA[i] | 0 | 4 | 3 | 2 | 1 |

a. The unit of suffixes of a string is character, while the unit of suffixes of a word sequences collection is word.

b. The suffix array for string stores one dimensional position information for the suffix in the string, while the suffix array for word sequences collection stores two dimensional information.

Continue with the example of previous four word sequence collection, we combine the word sequences as: s[0]#s[1]#s[2]#s[3]. Table 7.4 shows the index for the collection. The first three columns of Table 7.5 show the suffix array of sorted suffixes with two dimensional position information ($SA[i]$) and the suffixes represented by the position information ($Suffix$).

**Table 7.5:** Index for Four Word Sequences Collection.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | m | e | r | i | c | a | n |  | p | i | e |  | g | a |
| 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| v | e |  | m | e |  | s | u | c | h |  | f | a | l | s |
| 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |
| e |  | h | o | p | e |  | f | o | r |  | w | o | m | e |
| 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 |
| n | # | a | m | e | r | i | c | a | n |  | p | i | e |  |
| 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 |
| w | i | l | l |  | f | o | r | e | v | e | r |  | b | e |
| 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 |
|  | m | y |  | f | a | v | o | r | a | t | e | # | t | h |
| 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 |
|  | e | a | m | e | r | i | c | a | n |  | p | i | e |  |
| 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 |
| c | a | s | t |  | l | o | o | k | s |  | l | i | k | e |
| 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 |
|  | n | o | w | # | e | a | s | y |  | a | s |  | p | i |
| 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 |
| e | \0 |  |  |  |  |  |  |  |  |  |  |  |  |  |

Algorithm 1 shows how the suffix array over a collection of word sequences $S$ can be obtained.

Thanks to the lexicographical ordering, we can see after suffix sorting, the potential frequent semantic patterns are grouped together and can be extracted efficiently.

### 7.1.7 Candidate Generation and Inappropriate Semantic Pattern Exclusion

In this section, we will introduce the candidates generation from longest common prefix extraction and prove the generation process is theoretically well founded.

We first generate the candidates for frequent semantic patterns from suffix array through longest common prefix.

**Definition 12** (Longest Common Prefix ($LCP_\lambda$))**.** *The Longest Common prefix Array is an auxiliary data*

**Table 7.6:** $SA$ and $SLCP_\lambda$ Calculation

| $i$ | $Suffix$ | $SA[i]$ | $SLCP_3$ |
|---|---|---|---|
| 0 | american pie cast look like now | (2,1) | american pie: {(2,1),(0,0),(1,0)} |
| 1 | american pie gave me such false hope for women | (0,0) | |
| 2 | american pie will forever be my favorite | (1,0) | |
| 3 | as pie | (3,1) | |
| 4 | be my favorite | (1,4) | |
| 5 | cast look like now | (2,3) | |
| 6 | easy as pie | (3,0) | |
| 7 | false hope for women | (1,5) | |
| 8 | favorite | (1,6) | |
| 9 | for women | (0,7) | |
| 10 | forever be my favorite | (1,3) | |
| 11 | gave me such false hope for women | (0,2) | |
| 12 | hope for women | (0,6) | |
| 13 | like now | (2,5) | |
| 14 | look like now | (2,4) | |
| 15 | me such false hope for women | (0,3) | |
| 16 | my favorite | (1,5) | |
| 17 | now | (2,6) | |
| 18 | pie | (3,2) | pie: {(3,2),(2,2),(0,1)} |
| 19 | pie cast look like now | (2,2) | pie:{(2,2),(0,1),(1,1)} |
| 20 | pie gave me such false hope for women | (0,1) | |
| 21 | pie will forever be my favorite | (1,1) | |
| 22 | such false hope for women | (0,4) | |
| 23 | the american pie cast look like now | (2,0) | |
| 24 | will forever be my favorite | (1,2) | |
| 25 | women | (0,8) | |

The second column shows the suffixes after sorting, and the third column is the two dimensional position information of the suffixes stored in $SA[i]$.

*structure to the suffix array. For a suffix array $SA[i], i \in \{1, .., M\}$, given a threshold $\lambda$, the array stores the longest common prefixes among $\lambda$ consecutive suffixes in suffix array.*

$$LCP_\lambda(i)$$
$$= \begin{cases} LCP(S[SA[i]:], ..., S[SA[i+\lambda-1]:]) & i \in \{0, ...M-\lambda-1\} \\ \emptyset & i \in \{M-\lambda, ...M\} \end{cases}$$

where $LCP$ is to get the longest common prefix of multiple suffixes.

For example, for the suffix array in Table 7.6, given threshold 3, we have $LCP_3(0) = LCP(S[SA[0]:], S[SA[1]:], S[SA[2]:]) = LCP$(american pie cast look like now, american pie gave me such false hope for women, american pie will forever be my favorite) = american pie.

Accordingly, we define the semantic longest common prefix as:

**Algorithm 1:** Suffix Array Construction

SAConstruction $S$;
**Input** : A Collection of $n$ Word Sequences $(S)$
**Output** : Suffix Array for $S : SA$
**Method:**
Initialize $\boldsymbol{ComboS}$ to store the combination of all word sequences.
Initialize $\boldsymbol{SA}$ to store the starting position of each suffix in $ComboS$.
Initialize $\boldsymbol{Mapping}$ to store the correspondence between one dimensional position information in $ComboS$ and two dimensional position information in $S$.
**Step 1**. Combine all word sequences in S with '#' in between each word sequence to form a long sequence and we denote the total length of the sequence to be N. At the same time,
a) Initialize $\boldsymbol{SA}$ to be a vector of integers from 0 to N-1. Let each integer denote a suffix starting with a word in the $\boldsymbol{ComboS}$.
b) Store the correspondence between one dimensional position in $\boldsymbol{ComboS}$ and two dimensional position information in $S$.
$M = 0$ //Initialize the length of $SA$ to be 0.
$k = 0$ //Initialize the index of $SA$ to be 0.
$N = 0$ //Initialize the total length of $S$ to be 0.
**for** $i \in \{1, 2, ...n\}$ **do**
    $s = S[i]$
    $ComboS = ComboS + S[i] + \#$
    $L$ is the length of $S[i]$
    **for** $j \in \{1, ..., L\}$*)* **do**
        **if** $j$ *is a starting position of a word* **then**
            $\boldsymbol{SA}[k] = j + N$
            $k++$
            $Mapping[SA[k]] = (i, j)$
        **end**
    **end**
    N = N + L
**end**
M = k
**Step 2**. Leverage the Skew Algorithm to sort all the suffixes in initial $\boldsymbol{SA}$ and the resulted $\boldsymbol{SA}$ stores the one dimensional position information of sorted suffixes.
**Step 3**. Replace the one dimensional position information for each $SA[i]$ with the two dimensional position information.
**for** *i in* {*1,2,...M*} **do**
    $SA[i] = Mapping[SA[i]]$
**end**

**Definition 13** (Semantic $LCP_\lambda$ $(SLCP_\lambda)$)**.** . *Assume we have a collection of word sequences* $S = \{s_1, s_2, ..s_n\}$*, and its* $LCP_\lambda$ *is* $LCP_\lambda(i)(i \in \{1, 2, ..., M\})$*. We define the semantic longest common prefix*

*as:*

$$SLCP_\lambda(i)$$
$$= \begin{cases} LCP_\lambda(i) : \cup_{k=i}^{i+\lambda-1} SA[k] & i \in \{0, ...M-\lambda-1\} \\ \emptyset & i \in \{N-\lambda, ...M\} \end{cases}$$

The last column of Table 7.6 shows the $SLCP_3(i)$ for the previous collection of four word sequences. From Table 7.6, we can see both 'american pie' and pie' are the candidate frequent semantic patterns, since they all exceed the threshold 3.

In the following, we will prove how the frequent semantic pattern can be obtained from semantic longest common prefix.

The following lemma shows the longest common prefixes are all the frequent patterns.

**Lemma 1.** *Given a word sequences collection $S$ and frequency threshold $\lambda$, $\forall LCP_\lambda(i) \neq \emptyset$, $LCP_\lambda(i) \in FP(S)$ .*

**Proof 2.** *If $LCP_\lambda(i) \neq \emptyset$, there are at least $\lambda$ suffixes with prefix $LCP_\lambda(i)$. Therefore, $LCP_\lambda(i)$ is the frequent pattern.*

The following theorem will show us that the combination of all the semantic longest common prefixes is a subset of semantic pattern collection combination over frequent patterns.

**Theorem 2.** *Assume $S$ is a collection of word sequences, we have $\cup_{i\in\{1,2,...,M\}} SLCP_\lambda(i) \subseteq FPC$.*

**Proof 3.** *$\forall LCP_\lambda(i) \neq \emptyset (i \in 1, 2, ...M)$, we know from Lemma 2 that $LCP_\lambda(i) \in FP$. $\because LCP_\lambda(i) \in FP$. $\therefore \forall SLCP_\lambda(i)(i \in \{1, 2, ..n\}), SLCP_\lambda(i) \subset SP(S)(FP)..\therefore \cup_{i\in\{1,2,...,M\}} SLCP_\lambda(i) \subseteq SP(S)(FP) = FPC$.*

From Theorem 2, the candidates can be generated through $\cup_{i\in\{1,2,...,M\}} SLCP_\lambda(i)$ by scanning the suffix array only once.

The following theorem will show although $\cup_{i\in\{1,2,...,M\}} SLCP_\lambda(i)$ is a subset of $FPC$, the exclusive semantic pattern collection over $\cup_{i\in\{1,2,...,M\}} SLCP_\lambda(i)$ is the same as $FPC$, which proves that the frequent semantic patterns can be obtained through the above calculation.

**Theorem 3.** *Assume $S$ is a collection of word sequences, we have $ESPC(\cup_{i\in\{1,...M\}}SLCP_\lambda(i)) = ESPC(FPC)$.*

**Proof 4.** *From Theorem 2, $\because \cup_{i\in\{1,...,M\}}SLCP_\lambda(i) \subseteq FPC. \therefore ESPC(\cup_{i\in\{1,...M\}}SLCP_\lambda(i)) \subseteq ESPC(FPC)$.*

*At the same time, $\forall SP \in ESPC(FPC), \because SP \in ESPC(FPC)$, and $\nexists SP' \in ESPC(FPC), SP \subseteq SP', \therefore SP_{pattern} \in \cup_{i\in\{1,...M\}}LCP_\lambda(i), \therefore SP \in \cup_{i\in\{1,...M\}}SLCP_\lambda(i)$, and $\nexists SP' \in \cup_{i\in\{1,...M\}}SLCP_\lambda(i)$, $\therefore SP \in ESPC(\cup_{i\in\{1,...M\}}SLCP_\lambda(i)), \therefore ESPC(FPC) \subseteq ESPC(\cup_{i\in\{1,...M\}}SLCP_\lambda(i))$, $\therefore ESPC(\cup_{i\in\{1,...M\}}SLCP_\lambda(i)) = ESPC(FPC)$.*

Theorem 3 shows that we can eliminate the inappropriate semantic patterns from $\cup_{i\in\{1,2,...,M\}}SLCP_\lambda(i)$ to get the frequent semantic patterns.

Table 7.7 shows an example of how to get the frequent semantic pattern from $SLCP_\lambda(i)s$ $(i \in \{1, 2, ...M\})$.

**Table 7.7:** Frequent Semantic Pattern Calculation

| | |
|---|---|
| $SLCP_\lambda$ | american pie: {(2,1),(0,0),(1,0)}<br>pie: {(3,2),(2,2),(0,1),(1,1)} |
| $ESPC$ | american pie: {(2,1),(0,0),(1,0)}<br>pie: {(3,2),~~(2,2)~~,~~(0,1)~~,~~(1,1)~~} |
| $FSP$ | american pie: {(2,1),(0,0),(1,0)} |

Finally, the frequent semantic pattern extraction algorithm is shown in algorithm 2.

### 7.1.8 Complexity Analysis

The running time of Algorithm 1 and Algorithm 2 are mainly affected by three parts: the suffix array construction of Algorithm 1, the candidate pattern generation and pattern filtering of Algorithm 2. The skew algorithm for suffix array construction is linear. Both the candidate generation and inappropriate pattern exclusion from suffix array via longest common prefix can achieve linear time complexity. Therefore, although more complicated context information is integrated, the complexity of our semantic pattern mining algorithm is still $O(N)$, which is linearly scalable in terms of the total word sequences length.

### 7.1.9 Experiments

We will compare our algorithm against CCspan algorithmZhang et al. (2015). CCspan is an algorithm to mine contiguous sequential closed patterns, and it outperforms all the other sequential closed pattern algorithms.

**Algorithm 2:** Frequent Semantic Pattern Extraction

FSPExtraction $(S, \lambda)$;
**Input** : The Suffix Array for the Collection of Word Sequences and the Threshold $(SA, \lambda)$
**Output** : Frequent Semantic Patterns
Initialize $\underline{LCP}$ to store $LCP_\lambda(i)$.
Initialize $\underline{SLCP}$ to store $SPC(\cup_i SLCP_\lambda(i))$.
Initialize $\underline{ESPC}$ to store $ESPC$
Initialize $\underline{FSP}$ to store $FSP$
**Step1**: Calculate $LCP_\lambda(i)$.
**for** $i \in \{0, 1, 2, ..., M\}$ **do**
    calculate $LCP_\lambda[i]$ and store in $\underline{LCP}$.
**end**
**Step2:** Calculate $SPC(\cup_i SLCP_\lambda(i))$ and store in SLCP
**for** $i \in \{0, 1, 2, ..., M\}$ **do**
    calculate $SLCP_\lambda[i]$
    **if** $SLCP_\lambda[i]_{pattern} in \underline{SLCP}$ **then**
        add $SLCP_\lambda[i]_{position}$ to $\underline{SLCP}[SLCP_\lambda[i]_{pattern}]$
    **end**
**end**
**Step3:** Calculate $ESPC(\underline{SLCP})$ and store in $\underline{ESPC}$
**Step4:** Calculate $FSP(\underline{ESPC})$ and store in $\underline{FSP}$
Return $\underline{FSP}$

We will examine the result in three different ways. First, we will illustrate the semantic patterns against CCspan patterns. Second, we will show quantitatively to what degree the semantic patterns are more compact than CCspan patterns. Finally, we will examine how the reduction in the number of patterns effect the performance of the classifier built on patterns as features.

### 7.1.10 Data Set

We use two datasets to evaluate our results. The two datasets are mobile data and question data. The first dataset is a collection of mobile user requests. We collect the data from Merchanical Turk [1]. We ask the user to provide the query they would use to ask the mobile device for domain specific services. For example, the user may want to find the nearest Chinese restaurant, and he would say: 'take me to the nearest restaurant'. The data is from six domains: navigation, parking, music, phone call, text message and TV. For the second dataset, we use the question and answer dataset 'ydata-yanswers-all-questions-v1_0'[2]. We only extract the questions from food, news, travel, and cars categories out of 27 categories. There are 11318 sentences for mobile data, while 277617 sentences for question data.

[1] https://www.mturk.com/mturk/welcome
[2] https://research.yahoo.com/Academic_Relations

**Table 7.8:** Pattern Demonstration

| CCspan Patterns | Semantic Patterns |
|---|---|
| rock and roll, and roll, roll | rock and roll |
| where can i find, can i find | where can i find |
| reserve a parking spot, a parking spot | a parking spot |
| romantic comedy, comedy | romantic comedy |
| find me the cheapest, me the cheapest | find me the cheapest |
| turn up the volume, up the volume | turn up the volume |
| turn the volume up, the volume up | turn the volume up |
| turn the channel to, channel to | turn the channel to |
| add to the message, to the message | add to the message |

### 7.1.11 Pattern Demonstration

We show in Table 7.8 the comparison between semantic patterns and CCspan patterns for the mobile data set. We set the threshold to be 5. The semantic patterns exclude the CCspan patterns that can not be meaningful text units. For example, although 'up the volume' appears frequently, most of time it appears with 'turn up the volume'. The times it appears as an independent text unit doesn't exceed the threshold. Therefore, it is excluded from the semantic patterns.

### 7.1.12 Compactness Examination

Although we have proven theoretically that the frequent semantic patterns are more compact than CCspan patterns, we will show in this section the compactness of semantic patterns quantitatively.

**Table 7.9:** Compactness Demonstration

| corpus | $\lambda$ | # of $CCspan$ | # of $FSP$ | Reduced Percentage |
|---|---|---|---|---|
| Mobile | 5 | 3308 | 2710 | 18% |
| | 10 | 1460 | 1259 | 13.8% |
| | 15 | 948 | 822 | 13.3% |
| | 20 | 692 | 610 | 11.8% |
| | 25 | 559 | 492 | 11.6% |
| | 30 | 474 | 417 | 12% |
| Yahoo | 5 | 132310 | 109870 | 16.9% |
| | 10 | 58061 | 49548 | 14.6% |
| | 15 | 37511 | 32285 | 13.9% |
| | 20 | 27605 | 23849 | 13.6% |
| | 25 | 21909 | 18968 | 13.4% |
| | 30 | 18281 | 15847 | 13.3 |

Overall, the semantic patterns are on average more than 13% compact than CCspan patterns as shown in

Table 7.9.

### 7.1.13 Classification

One typical application of patterns is to build classifiers by using these patterns as features. In this section, we will evaluate the power of the semantic patterns in terms of both precision and recall. We use only the mobile data. We build a binary classifier. For each part, we randomly select 90% as training and 10% as testing data.

As we can see from Table 7.10, most of the time, as the frequency threshold goes higher, the precision and recall increase, and the precision and recall are the same or higher using semantic patterns than using closed patterns.With more than 13% less features, semantic patterns are the same and more representative than closed patterns.

**Table 7.10:** Classification Comparison

| $\lambda$ | CCspan Patterns | | Semantic Patterns | |
|---|---|---|---|---|
| | precision | recall | precision | recall |
| 5 | 0.97 | 0.886 | 0.963 | 0.886 |
| 10 | 0.949 | 0.873 | 0.977 | 0.873 |
| 15 | 0.978 | 0.886 | 0.978 | 0.886 |
| 20 | 0.978 | 0.886 | 0.978 | 0.886 |
| 25 | 0.978 | 0.886 | 0.978 | 0.886 |
| 30 | 0.978 | 0.88 | 0.978 | 0.893 |

### 7.1.14 Conclusion

To incorporate the context information into pattern mining for text mining, we introduce a new concept as semantic continuous sequential pattern. A novel semantic pattern mining problem is proposed and solution is provided. By including the position information in defining the semantic pattern, our algorithm obtains only the semantic independent text units as patterns, through excluding all the text segments of no independent meaning. Our algorithm provides a novel perspective to generate the candidate patterns with suffix array and longest common prefix, which is perfectly compatible with our theoretical framework and our proposed problem.

Although more information is incorporated, the algorithm is still running in linear time. It is proved that the semantic patterns are more compact and representative than the state of the art continuous sequential patterns.

**Algorithm 3:** Spoken Language Separation

Spoken Language Segmentation $S$;
**Input** : A Collection of $n$ Short Sentences $(S)$
**Output** : Sentences Segmentation $Seg(S)$
**Method:**
Initialize $Seg(S)$ to be a list.
**Step 1**. Extract the patterns $P(S)$ from $S$.
**for** $i \in \{1, 2, ..., n\}$ **do**
    a. Separate $s$ with restricted intent specific entities and get $s_1$.
    b. Separate $s_1$ with prepositions and get $s_2$.
    c. Separate $s_2$ with patterns $P(S)$ and get $s_3$. and add $s_3$ to $Seg(S)$.
**end**

## 7.2 Segmentation

In this section, we will leverage the patterns mined from the previous section to help segment the spoken language into intent and entity phrases. The algorithm is as following.

The result is shown in Table 7.11. We use the mobile data as in section 7.1.

**Table 7.11:** Spoken Language Segmentation

| | Navigation | Music |
|---|---|---|
| Accuracy | 0.82 | 0.83 |

# Chapter 8: Intent Entity Topic Model

In this chapter, we will introduce a new method to classify the text into intents and find the intent related entities.

With the rampant expansion of smart phone usage, the spoken language understanding, represents itself as a new research area to be explored. The spoken language is the user request asking artificial intelligence to provide services to user requests. For example, the user may try to find a nearby restaurant to have lunch. The answer to the user's spoken language 'find me a nearby restaurant' could be a list of yelp pages with restaurant information. Thus, the intention mining is service oriented. Therefore, the spoken language understanding engine must have the ability to process what a user says and map it to the actions the user intent to take. The result can then be passed to an application that takes the appropriate action.

Since the user query is relatively short, rendering less features to do text analysis, most of the engine right now requires annotation of sample sentences that the users might say. The annotation indicates the function to be performed (e.g., navigate to a destination, or play a song) and the entities related to that function (e.g., the destination, the song name, etc.). For example, the sentence 'take me to the nearest Starbucks' will be annotated as 'navigation' intention, with 'starbucks' being annotated as 'destination'.

However, the annotation is labor intensive and there are countless ways a user might choose to navigate to a destination. We need to find new ways to automatically find the intentions and entities.

To learn more about the spoken language, we collect data from intents 'navigation' and 'music'. Table 8.1 shows the data collected for parking intent.

Although the query is relatively short, we found the service-oriented short query has prominent patterns embedded in the dataset. First, users try to use similar words/phrase to express their intention. For example, when asking for navigation, the users prefer to use 'could you find' or 'navigate me to', but when asking for a song, they would use 'I want to hear' or 'play'. Second, entities play an important role in delivering the user intent and also contribute to the detection of user intent. For example, 'starbucks', 'targets' or 'coffee shop' are more likely to appear in 'navigation' intent frequently, while 'taylor swift' or 'classic music' appears

**Table 8.1:** Sample Sentences

| navigation | music |
|---|---|
| show me any top rated restaurants in yelp | i want to hear a song from my playlist |
| find me a gas station | can you put on californication |
| is there a mcdonalds nearby | play album 10 |
| where is the closest jc penney store | i want to listen to bad blood |
| locate a restaurant with free valet service | listen to yellow |
| where is the nearest bank | go to michael jackson collection now |
| change destination to chicago | listen to the album welcome to the black parade |
| where is the closest starbucks located | i want to hear the whole album 21 |
| i need to go to the closest publix to indian river blvd | play songs by adele from album 25 |
| go to the nearest dunkin donuts | play some jazz music |
| find me a cheap hotel here | please put on some classical music for me |
| take me to a coffee shop nearby | put on a jazz playlist |
| is there a starbucks within 10 miles from here | play something by the beatles |
| search for top rated pizza | play taylor swift |
| show me the best pizza places in hudson | i want to hear classical music |
| find me dollar oysters | i want to hear my deep sleep playlist |
| find a starbucks near the mall | music rolling stones |
| search for pizza near lexington | please play album favorites |
| take me to closest park | put on 80s music |
| start navigation to 2870 zanker road in san jose california | please turn on all highly rated metal |

more in 'music' intent. Therefore, the user request mainly consists of two parts: the user intent expression and entities. Obviously, the intent expression and entities serve each other to show the user intent.

Therefore, in this chapter, we will propose a topic model designed specifically for user requests, leveraging the spoken language texts characteristics. In our model, the topic is not the distribution of words, but a distribution of intent expression and entity expressions. The separation of the intent and entities expressions can help better understand the user intention and further better provide the services.

The novelties of this model are:

a. It is tailored specifically to spoken language by leveraging its specific characteristics.

b. It incorporates free available entity database and the patterns extracted as prior knowledge.

## 8.1 Related Work

Few work has been done to investigate spoken language with topic modeling. Topic models show some drawbacks when dealing with short sentences. Due to the lack of word in each sentence, the use of topic modeling on short texts is not effective as the use of topic modeling on long documents. Some models enrich the short texts with pre defined topical knowledge Yang et al. (2015). For the other models, one topic distribution is generated for the whole corpus instead of one for each document.

Previous research has shown the effectiveness of adding prior knowledge to topic modeling. The topic model with topic-in-set knowledge Andrzejewski & Zhu (2009) encourages the recovery of topics which are

more relevant to user modeling goals. Previous work also shows the use of the patterns are most effective for language representation and understanding. Single words have no correlated semantic meanings, and people utilize the combination of single words to solve the problem of semantic ambiguity. In general, phrases carry more specific content than single words. Data mining techniques were applied to text mining and classification by using word sequences as descriptive phrases (n-Gram) from document collections Cavnar et al. (1994) Fürnkranz (1998). But the performance of n-Gram is restricted due to its low frequency in documents. Although phrases and n-Gram are stronger at interpreting semantic meaning, they perform less well with statistical properties in matching representations with documents when compared with term-based representation. In order to balance the statistical and semantic properties, researchers propose to extract pattern-based features.

This part of the thesis leverages the characteristics of short texts to model the intent words and entity words separately. Also, we will add the external knowledge of entity databases to give us more effective semi-supervised topic model exclusively for spoken language.

## 8.2 Intent Entity Topic Model

Conventional topic model learns no distinction between intent phrase/words and entities, which can not capture the special characteristics of the spoken language. To tackle this problem, our model extended traditional topic model to capture both intent words/phrases and opinion words. Our model can produce simple and meaningful intent words and entity words. Specifically, we assume there are $T$ intentions in a given collection of user queries.

To understand how we model the opinion words, let us first look at two sample sentences from the navigation intent:

Find me a gas station.

Take me to gas station.

I want to hear hip hop.

I want to hear classical music.

We can see that there is a strong association between ‘find me‘/‘take me to’ and ‘gas station’, and ‘i want to hear’ and ‘hip hop‘/‘classical music’. ‘find me’ and ‘take me to’ are intent words for ‘navigation’ intent,

while ‘gas station’ is the entity words for ‘navigation’ intent. ‘i want to hear’ are intent words and ‘hip hop’ and ‘classical’ are entities for ‘music’ intent. If we know ‘find me a’ is the intent phrase for ‘navigation’ intent, ‘gas station’ is entity word. Then the association between ‘find me a’ and ‘gas station’ and ‘take me to’ and ‘gas station’ can help us to identify ‘take me to’ is also intent word to express ‘navigation’ intention. If we know ‘hip hop’ is the entity for intent ‘music’, ‘i want to hear’ is intent phrase. The association between ‘i want to’ and ‘hip hop’ and ‘i want to’ and ‘classical music’ can help us to infer ‘classical music’ is the entity word for music intent. We therefore introduce an intent entity model to capture the association between the intent and entity words.

We now describe the generative process of the model. First, we draw two multinomial word distributions for each topic from a symmetric Dirichlet prior with parameter $\beta$: an intent word distribution $\Phi_{i_z}(z \in \{1, 2..., Z\})$ and entity word distribution $\Phi_{e_z}(z \in \{1, 2, ..., Z\})$. For each topic, the intent and entity words all have different vocabularies. Second, we draw a topic distribution $\theta_z$ and type distribution $\theta_t$ over the whole corpus, where $\theta_z \propto Dir(\alpha_z)$ and $\theta_t \propto Dir(\alpha_t)$. For each user request, we draw a topic assignment $z_d \propto \theta_z$.

Now for each word in query $d$, we have two choices: the word may be an intent word or an entity word. To distinguish between the two choices, we introduce an indicator variable $t_{d,n}$. For the $n_{th}$ word $w_{d,n}$, we draw $t$ from a multinomial distribution over $\{0,1\}$. It determines whether $w_{d,n}$ is a background word, intent word or entity word. If a word is chosen to be an intent word, it is generated from the topic-intent word distribution of topic $z_d$: $\Phi_{i_{z_d}}$; while if a word is chosen to be an entity word, it should be generated from the topic-entity distribution $\Phi_{e_{z_d}}$.

Figure 8.1 shows the graphical illustration for this model and Table 8.2 shows the annotations through the generative process.

The full conditional equation used for sampling individual $z_i$ and $t_i$ values from the posterior is given by

$$P\{t_i = 1, 2|t_{\neg i}, z\} = \frac{\alpha + n_{t_i}^{\neg i}}{3\alpha + \sum_{t=1}^{T} n_t^{\neg i}} \frac{\beta + n_{z_i,t_i,w_i}^{\neg i}}{\sum_{w=1}^{W} n_{z_i,t_i,w_i}^{\neg i} + V\beta} \tag{8.1}$$

where $n_{t_i,w_i}^{\neg i}$ is the number of times word $w_i$ is assigned as type $t_i$, $n_{t_i}^{\neg i}$ is the number of times a word is assigned as type $t_i$. $n_{z_i,t_i,w_i}^{\neg i}$ is the number of times word $w_i$ is assigned to type $t_i$ and topic $z_i$. The $\neg i$

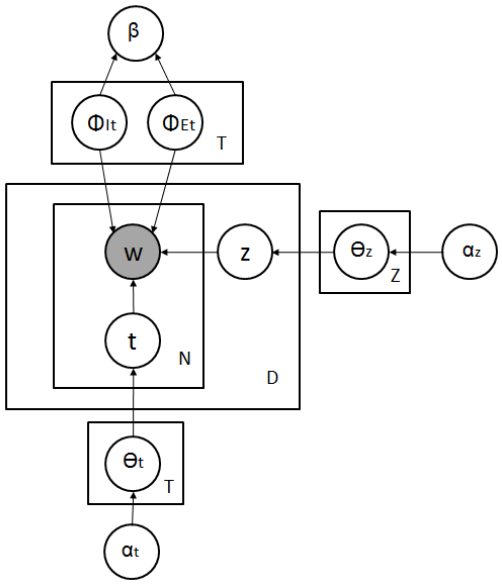



**Figure 8.1:** Pattern Entity Topic Model

**Table 8.2:** Annotations in the generative process.

| Notation | Description |
|---|---|
| $D$ | Number of spoken language texts |
| $N$ | Number of words in each text |
| $T$ | Number of topics |
| $w$ | Word |
| $z$ | Underlying topic for each word |
| $t$ | Underlying type (either it be an entity word or be an intent word) |
| $\alpha_z$ | Dirichlet prior for $\theta_z$ |
| $\alpha_t$ | Dirichlet prior for $\theta_t$ |
| $\beta$ | Dirichlet prior for $\Phi_I, \Phi_E$ and $\Phi_B$ |
| $\theta_z$ | Topic distribution for the whole corpus |
| $\theta_t$ | Type distribution for the whole corpus |
| $\Phi_{It}$ | Word distribution for each topic $t$ |
| $\Phi_{Et}$ | Word distribution for each topic $t$ |

notation signifies that the counts are taken omitting the value of $z_i$ and $t_i$.

$$P\{z_q|z_{\neg q}, t\} = \frac{\alpha + n_{z_i}^{\neg q}}{T\alpha + \sum_{z=1}^{Z} n_z^{\neg q}} \left(\frac{\Gamma(n_{t_i,z_i}^{\neg q} + V\beta)}{\Gamma(n_{t_i,z_i}^{\neg q} + n_{d,t_i,z_i}^{\neg q} + V\beta)} \prod_{v=1}^{V} \frac{\Gamma(n_{t_i,z_i,w_i}^{\neg q} + n_{d,t_i,z_i,w_i}^{\neg q} + \beta)}{\Gamma(n_{t_i,z_i,w_i}^{\neg q} + \beta)}\right) \tag{8.2}$$

where $n_{z_i}^{\neg q}$ is the number of times a word $w$ is assigned to topic $z_i$, $n_{t_i,z_i}^{\neg q}$ is the number of times a word is assigned as type $t_i$ and topic $z_i$. $n_{d,t_i,z_i}$ is the number of time a word in query $d$ is assigned to type $t_i$ and topic $z_i$. $n_{t_i,z_i,w_i}^{\neg q}$ is the number of times word $w_i$ is assigned as topic $z_i$ with type $t_i$. $n_{d,t_i,z_i,w_i}^{\neg q}$ is the number

of times word $w_i$ in query $d$ is assigned topic $z_i$ with type $t_i$. The $\neg q$ notation signifies that the counts are taken omitting the value of $z$ and $t$ in query $q$.

We show the intermediate result in Table 8.3. In the next chapter, we will further improve the model by leveraging both the entity databases and pattern extraction method.

**Table 8.3:** Intermediate Result

| navigation | | music | |
|---|---|---|---|
| **intent** | **entity** | **intent** | **entity** |
| find 0.0382790459092 | restaurant 0.0408047067755 | play 0.123688458434 | song 0.0653896961691 |
| i 0.0346242626314 | store 0.0328335547542 | music 0.0520581113801 | some 0.0396301188904 |
| show 0.0171197743011 | gas 0.0294173467451 | playlist 0.0494350282486 | yellow 0.0204755614267 |
| closest 0.0157091561939 | traffic 0.0237236667299 | album 0.0401533494754 | beatles 0.0178335535007 |
| want 0.0135932290331 | starbucks 0.0227747200607 | listen 0.0359160613398 | coldplay 0.0171730515192 |
| nearest 0.0135291100282 | home 0.0208768267223 | hear 0.0312752219532 | adele 0.0125495376486 |
| go 0.0132085150038 | restaurants 0.0191687227178 | start 0.0203793381759 | rock 0.0125495376486 |
| where 0.0127596819697 | chinese 0.0168912507117 | put 0.0145278450363 | classical 0.0118890356671 |
| near 0.0123108489356 | food 0.0167014613779 | songs 0.0141242937853 | nirvana 0.0112285336856 |
| get 0.0102590407797 | shop 0.0163218827102 | taylor 0.0133171912833 | justin 0.0112285336856 |

## 8.3 Entity Databases

Since there are lots of entity databases available online, such as MusicBrainz[1], we try to add these knowledge to our model. By constraining some seed entities to appear only in restricted sets of topics, these terms will be concentrated in only certain topics of entity type. The split within those set of topics may be different from previous topic model will produce, thus revealing new information within the data. Therefore, the supervision is used to encourage the recovery of topics which are more relevant to user modeling goals.

Let

$$P\{t_i|t_{\neg i}, z\} = \frac{\alpha + n_{t_i}^{\neg i}}{2\alpha + \sum_{t=1}^{T} n_t^{\neg i}} \frac{\beta + n_{z_i,t_i,w_i}}{\sum_{z=1}^{Z}\sum_{t=1}^{T} n_{z_i,t_i,w_i} + V\beta} \tag{8.3}$$

$$\begin{aligned} P\{z_q|z_{\neg q}, t\} = &\frac{\alpha + n_{z_i}^{\neg q}}{T\alpha + \sum_{z=1}^{Z} n_z^{\neg q}} \Big(\frac{\Gamma(n_{t_i,z_i}^{\neg q} + V\beta)}{\Gamma(n_{t_i,z_i}^{\neg q} + n_{d,t_i,z_i}^{\neg q} + V\beta)} \\ &\prod_{v=1}^{V} \frac{\Gamma(n_{t_i,z_i,w_i}^{\neg q} + n_{d,t_i,z_i,w_i}^{\neg q} + \beta)}{\Gamma(n_{t_i,z_i,w_i}^{\neg q} + \beta)}\Big) \end{aligned} \tag{8.4}$$

Assume we have topic in-set knowledge as sets of entity words $C = \{C_i\}, \{i \in 1, 2, ..., |C|\}$, where each $C_i$ is an entity set of a specific topic. For each entity, it may belong to multiply entity word sets. We set a hard constraint by modifying the Gibbs Sampling equation with two indicator functions $\delta_z = \{\delta_{z_1}, ...\delta_{z_{|}Z|}\}$ and $\delta_t = \{t_1, t_2\}$. Therefore, for each word $v \in V$, if $\exists C_i \in C$ that $v \in C^i$, $\delta_{z_i}$ takes on value 1, if $v \in C^{(i)}$ and

[1] `https://musicbrainz.org/`.

0 otherwise, and $\delta_t = \{0, 1\}$. If $\nexists C_i \in C$, then $\delta_{z_i} = \{1, 1, ..., 1\}$, and $\delta_t = \{1, 1\}$

$$P(z_i, t_i | t_{\neg i}, z_{\neg i}, w) \propto q_{i,z,t} \delta_z \delta_t \tag{8.5}$$

If we wish to restrict $z_i$ to a single value (e.g., $z_i$ = 5), this can now be accomplished by setting $C = \{5\}$. Finally, for unconstrained $z_i$, we simply set $C = \{1, 2, ..., 1\}$, in which case our modified sampling reduces to the standard Gibbs sampling.

This formulation gives us a flexible method for inserting prior intent knowledge into the inference of latent topics.

## 8.4 Pattern Mining

As we discussed in the section 8.1, phrases have more power to convey meaning. Thus, we replace the individual words with patterns. The words not belonging to a pattern remain independently. We extract patterns as in Section 7.2. Therefore, the phrase 'i want listen' will form into one unit to show the 'play music' intention.

## 8.5 Data Set

We use mobile query data set to evaluate our results.We collect the data from Merchanical Turk [2]. We ask the user to provide the query they would use to ask the mobile device for intent specific services. For example, the user may want to find the nearest Chinese restaurant, and he would say: 'take me to the nearest restaurant'. The data is from six intents: navigation, parking, music, phone call, text message and TV. Here we only use the data from navigation and music intents. The detailed statistics are shown in Table 8.4.

**Table 8.4:** Data Statistics

| Datasets | # short texts | # words | # vocabulary |
|---|---|---|---|
| Navigation | 2597 | 8737 | 3272 |
| Music | 1479 | 7731 | 1218 |

[2] `https://www.mturk.com/mturk/welcome`

## 8.6 Experiments

### 8.6.1 Pattern Entity Demonstration

This section will show the word distribution for both the intention and entity words for intent music and navigation.

We set the number of topics to be two. As we can see from the Table 8.5, the 'show me', 'take me to', 'search for' and 'where is the nearest' are all representative patterns for 'navigation' intention, while 'listen to', 'put on', 'turn on' and 'start playing' are all patterns to show 'play music' intention. The incorporation of pattern into the model makes the 'navigation' intention more understandable to users. Further, the intention entity topic model can help user to find the intention related entities. 'starbucks', 'restaurant', 'pizza', 'mc donalds' and 'coffee shop' are all all points of interests (POI) that are related to 'navigation' intention, while the songs, singers, genres such as 'taylor swift','yellow', 'beatles', and 'classical' are more likely to appear as entity words.

On the other side, the traditional topic models such as Latent Dirichlet Allocation don't distinguish between the intention and entity words. Although traditional topic modeling still figures out the word distribution for each topic, the mixture of both the intention and entity words can hardly illustrate the characteristics of the spoken language.

Obviously, the separation of the 'intention' and 'entity' for topics makes spoken language more understandable and thus makes the intention entity topic model a more expressive language model.

**Entity Identification**

One result from intent entity topic model is the entities obtained for each intent. In this section, we will demonstrate the effectiveness of the entity identification from the intention entity topic model. We select top 100 entity words for both 'navigation' and 'music' intention, and the accuracy for the obtained entities are shown in Table 8.6.

**Classification**

Another good way for evaluating topic model is through classification. In this section, we will show how the incorporation of entity and patterns into the model affects the classification result. The evaluation result on the

**Table 8.5:** Pattern & Entity

| Pattern Entity Topic Model | | | | Topic Model | |
|---|---|---|---|---|---|
| navigation | | music | | | |
| pattern | entity | pattern | entity | navigation | music |
| find_me_a | starbucks | play | song | find | play |
| 0.014050719671 | 0.0364188163885 | 0.0641781270465 | 0.0732790525537 | 0.03469884185280764 | 0.05947757363267441 |
| find_a | home | playlist | taylor_swift | show | music |
| 0.0130226182317 | 0.0333839150228 | 0.0530451866405 | 0.0347890451517 | 0.01538689408051772 | 0.025869499946730722 |
| near | restaurant | album | yellow | closest | playlist |
| 0.0106237148732 | 0.0324734446131 | 0.032416502947 | 0.0222057735011 | 0.013962709141558284 | 0.0257726467084427 |
| nearby | gas_station | music | beatles | restaurant | want |
| 0.0102810143934 | 0.0312594840668 | 0.0271774721676 | 0.019985196151 | 0.012253687214806965 | 0.02461040784898643 |
| place | store | i_want_to_listen_to | coldplay | nearest | album |
| 0.00976696367375 | 0.0288315629742 | 0.0144073346431 | 0.019245003701 | 0.012025817624573455 | 0.019767745934585322 |
| here | pizza | songs | adele | where | listen |
| 0.00839616175463 | 0.016995447648 | 0.0127701375246 | 0.0140636565507 | 0.011342208853872927 | 0.017249561739096746 |
| find | chinese_restaurant | music_by | nirvana | near | hear |
| 0.00753941055517 | 0.0142640364188 | 0.0098231827112 | 0.0125832716506 | 0.011000404468522663 | 0.015893616403064435 |
| show_me | hotel | start | eminem | store | like |
| 0.00719671007539 | 0.0142640364188 | 0.00949574328749 | 0.0118430792006 | 0.009861056517355117 | 0.010082422105783107 |
| work | mcdonalds | listen_to | rock | get | start |
| 0.00685400959561 | 0.0133535660091 | 0.00916830386379 | 0.0111028867506 | 0.009519252132004852 | 0.00979186239091904 |
| take_me_to | mall | put_on | kfix | route | song |
| 0.00633995887594 | 0.0124430955994 | 0.00884086444008 | 0.00888230940044 | 0.00934834993932972 | 0.009598155914342996 |
| area | grocery_store | turn_on | justin_bieber | go | please |
| 0.00616860863605 | 0.0118361153263 | 0.00785854616896 | 0.00888230940044 | 0.008835643361304324 | 0.008339063816598708 |
| near_me | hospital | i_want_to_hear | bad_blood | gas | taylor |
| 0.00616860863605 | 0.0112291350531 | 0.00785854616896 | 0.00814211695041 | 0.008835643361304324 | 0.007176824957142442 |
| house | cafe | play_some | red | take | put |
| 0.00599725839616 | 0.0109256449165 | 0.00720366732155 | 0.00814211695041 | 0.008721708566187569 | 0.006983118480566397 |
| where_is_the_closest | walmart | start_playing | metallica | stop | songs |
| 0.00548320767649 | 0.00880121396055 | 0.00622134905043 | 0.00740192450037 | 0.008607773771070816 | 0.006789412003990353 |
| is_there_a | bank | open | prince | traffic | rock |
| 0.0053118574366 | 0.00819423368741 | 0.00523903077931 | 0.00666173205033 | 0.0076962954101367775 | 0.006692558765702331 |
| find_me | coffee_shop | play_album | aerosmith | station | of |
| 0.0053118574366 | 0.00789074355083 | 0.0049115913556 | 0.00666173205033 | 0.0073544910247865134 | 0.005917732859398154 |
| search_for | park | play_me | rolling_stones | need | swift |
| 0.00496915695682 | 0.00758725341426 | 0.0049115913556 | 0.00666173205033 | 0.007126621434553004 | 0.00572402638282211 |
| list | bar | workout_playlist | michael_jackson | best | go |
| 0.00462645647704 | 0.00667678300455 | 0.00425671250819 | 0.00666173205033 | 0.006898751844319495 | 0.005530319906246066 |
| go_to | food | pop | beyonce | there | country |
| 0.00428375599726 | 0.00667678300455 | 0.00425671250819 | 0.00666173205033 | 0.006898751844319495 | 0.005530319906246066 |
| where_is_the_nearest | library | genre | classical | starbucks | destination |
| 0.00411240575737 | 0.00667678300455 | 0.00360183366077 | 0.00666173205033 | 0.006841784446761118 | 0.0054334666679580436 |

**Table 8.6:** Entity Identification

| Model | # of words | navigation | | music | |
|---|---|---|---|---|---|
| | | # of seed | # of correct | # of seed | # of correct |
| none | 100 | 0 | 52 | 0 | 23 |
| entity | 100 | 35 | 80 | 25 | 72 |
| pattern | 100 | 0 | 44 | 0 | 27 |
| pattern & entity | 100 | 40 | 79 | 25 | 71 |

classification of music and navigation data is shown in Table 8.7.

**Table 8.7:** Classification Comparison

| Model | Precision | Recall |
|---|---|---|
| none | 0.975 | 0.963 |
| entity | 0.969 | 0.969 |
| pattern | 0.994 | 0.963 |
| pattern &entity | 0.994 | 0.969 |

## 8.7 Conclusion

In this chapter, we propose a novel language model exclusively for service oriented spoken language understanding. The model leverages the characteristics of spoken language to separate it into intent and entity words. The separation and different treatment of the intent and entity words improve the model performance. The model is further modified through combining the pattern mining techniques and freely available entity databases.

# Chapter 9: Conclusion and Future Work

This thesis combines the exploration of language structures and topic modeling according to these structure. The thesis explores two types of texts, the long document texts and spoken language text. Although research has been done quite a lot to statistically modeling the language, there are still language structures not yet been covered. Different type of language may have different semantic and syntactic structures, and the modeling of different language should be dependent on their unique structures.

This thesis explores the characteristics of language itself and builds the topic modeling exclusively towards these language structures. Two types of texts are examined. One is the normal document texts and the other is the short spoken language texts. The normal document texts include the written texts, such as research paper, or news article. The short spoken language texts are human articulated texts dictated to machines for a specific purpose.

In Chapter 2, the basis for topic modeling is introduced. The origin and classical topic modeling methods are described in detail.

Chapter 3 to Chapter 5 cover the first part of the thesis for long documents. Chapter 3 introduces the problem of pairwise relation network for documents. Chapter 4 and Chapter 5 propose topic modeling techniques trying to extract the word and topic relation structures from the long documents. Chapter 4 applies pairwise topic model to both the biology papers and medical records, and Chapter 5 applies the pairwise topic model to the news articles and research papers for topic transition and evolution. In Chapter 4, the word pairs are extracted with the information extraction tools and Chapter 5 extend the word pairs to include all the pairs with their mutual information exceeding a certain threshold.

Chapter 6 to Chapter 8 cover the second part of the thesis dealing with spoken language. In Chapter 6, the characteristics of spoken language are studied and a new form of language (Intent Specific Sublanguage) is defined in between the spoken language and machine language. The problem of spoken language understanding is proposed based on this new form of language and a new language pipeline is introduced. Chapter 7 presents a new way of syntactic segmentation based on the specific characteristics for spoken language. Chapter 8

shows the topic modeling to automatically distinguish the intent and entity words and find topic for the intent and entity words.

Our further work includes two parts. First, for the pairwise topic model, we need to further examine the model itself to reduce the time complexity and to make it scalable. Second, for the intent entity topic model, we will integrate the result from Chapter 7 to the model in Chapter 8.

# Bibliography


Agarwal, R. C., Aggarwal, C. C., & Prasad, V. 2001, Journal of parallel and Distributed Computing, 61, 350

Agrawal, R., Imieliński, T., & Swami, A. 1993, ACM SIGMOD Record, 22, 207

Agrawal, R., & Srikant, R. 1995, in Data Engineering, 1995. Proceedings of the Eleventh International Conference on, IEEE, 3

Agrawal, R., Srikant, R., et al. 1994, in Proc. 20th int. conf. very large data bases, VLDB, Vol. 1215, 487

Andrzejewski, D., & Zhu, X. 2009, in Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing, Association for Computational Linguistics, 43

Bai, B., Weston, J., Grangier, D., et al. 2009, in Proceedings of the 18th ACM conference on Information and knowledge management, ACM, 187

Bayardo Jr, R. J. 1998, ACM Sigmod Record, 27, 85

Blei, D. M., & Lafferty, J. D. 2006, in Proceedings of the 23rd international conference on Machine learning, ACM, 113

Blei, D. M., & Lafferty, J. D. 2007, The Annals of Applied Statistics, 17

Blei, D. M., Ng, A. Y., & Jordan, M. I. 2003, the Journal of machine Learning research, 3, 993

Borgelt, C., & Berthold, M. R. 2002, in Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on, IEEE, 51

Boyd-Graber, J. L., & Blei, D. M. 2009, in Advances in neural information processing systems, 185

Caplan, L. J., & Herrmann, D. J. 1993, Zeitschrift für Psychologie mit Zeitschrift für angewandte Psychologie

Cavnar, W. B., Trenkle, J. M., et al. 1994, Ann Arbor MI, 48113, 161

Chaffin, R. 1989, in Relational models of the lexicon, Cambridge University Press, 289

Chang, J., & Blei, D. M. 2009, in International Conference on Artificial Intelligence and Statistics, 81

Chen, H., Branavan, S., Barzilay, R., & Karger, D. R. 2009, in Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 371

Firth, J. R. 1957, Papers in linguistics, 1934-1951 (Oxford University Press)

Fournier-Viger, P., Gomariz, A., Campos, M., & Thomas, R. 2014, in Advances in Knowledge Discovery and Data Mining (Springer), 40

Fürnkranz, J. 1998, Austrian Research Institute for Artifical Intelligence, 3, 1

Gomariz, A., Campos, M., Marin, R., & Goethals, B. 2013, in Advances in knowledge discovery and data mining (Springer), 50

Green, R., Bean, C. A., & Myaeng, S. H. 2013, The semantics of relationships: an interdisciplinary perspective, Vol. 3 (Springer Science & Business Media)

Griffiths, T. L., & Steyvers, M. 2004, Proceedings of the National academy of Sciences of the United States of America, 101, 5228

Gruber, A., Weiss, Y., & Rosen-Zvi, M. 2007, in International Conference on Artificial Intelligence and Statistics, 163

Han, J., & Kamber, M. 2006, Data Mining: Concepts and Techniques (2nd ed., pp. 227-283). San Francisco, USA: Morgan Kaufmann Publishers

He, Q., Chen, B., Pei, J., et al. 2009, in Proceedings of the 18th ACM conference on Information and knowledge management, ACM, 957

Hofmann, T. 2001, Machine learning, 42, 177

Hu, B., Lu, Z., Li, H., & Chen, Q. 2014, in Advances in Neural Information Processing Systems, 2042

Huan, J., Wang, W., Prins, J., & Yang, J. 2004, in Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 581

Inokuchi, A., Washio, T., & Motoda, H. 2000, in Principles of Data Mining and Knowledge Discovery (Springer), 13

Jo, Y., Hopcroft, J. E., & Lagoze, C. 2011, in Proceedings of the 20th international conference on World wide web, ACM, 257

Kärkkäinen, J., & Sanders, P. 2003, in Automata, Languages and Programming (Springer), 943

Karpathy, A., & Fei-Fei, L. 2015, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3128

Khoo, C. S., & Na, J.-C. 2006, Annual review of information science and technology, 40, 157

Kuramochi, M., & Karypis, G. 2001, in Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, IEEE, 313

Liu, G., Lu, H., Lou, W., & Yu, J. X. 2003, in Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 607

Manber, U., & Myers, G. 1993, siam Journal on Computing, 22, 935

Mannila, H., Toivonen, H., & Verkamo, A. 1994, [a]Efficient Algorithms for Discovering Association Rules, [o] AAAI Workshop Knowledge Discovery in Databases (KDD-94)

McCreight, E. M. 1976, Journal of the ACM (JACM), 23, 262

Pasquier, N., Bastide, Y., Taouil, R., & Lakhal, L. 1999, in International Conference on Database Theory, Springer, 398

Pei, J., Han, J., & Lakshmanan, L. V. 2001, in Data Engineering, 2001. Proceedings. 17th International Conference on, IEEE, 433

Pei, J., Han, J., Mortazavi-Asl, B., et al. 2004, Knowledge and Data Engineering, IEEE Transactions on, 16, 1424

Putthividhy, D., Attias, H. T., & Nagarajan, S. S. 2010, in Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 3408

Srikant, R., & Agrawal, R. 1996, Mining sequential patterns: Generalizations and performance improvements (Springer)

Srivastava, N., & Salakhutdinov, R. 2012a, in International Conference on Machine Learning Workshop

Srivastava, N., & Salakhutdinov, R. R. 2012b, in Advances in neural information processing systems, 2222

Tang, J., Zhang, J., Yao, L., et al. 2008, in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 990

Thomo, A. 2009, Victoria, Canda.[online] Available at:.[29 July 2012]

Tian, J., Huang, Y., Guo, Z., et al. 2015, Signal Processing Letters, IEEE, 22, 886

Tran, T. H., & Choi, S. 2014, in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, IEEE, 5979

Ukkonen, E. 1995, Algorithmica, 14, 249

Vanetik, N., Gudes, E., & Shimony, S. E. 2002, in Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on, IEEE, 458

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. 2015, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3156

Wallach, H. M. 2006, in Proceedings of the 23rd international conference on Machine learning, ACM, 977

Wang, C., Blei, D., & Li, F.-F. 2009, in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 1903

Wang, H., Zhang, D., & Zhai, C. 2011, in Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 1526

Wang, J., & Han, J. 2004, in Data Engineering, 2004. Proceedings. 20th International Conference on, IEEE, 79

Wang, X., & McCallum, A. 2006, in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 424

Wang, X., McCallum, A., & Wei, X. 2007, in Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on, IEEE, 697

Wang, X., Zhai, C., & Roth, D. 2013, in Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 1115

Wu, H., Min, M. R., & Bai, B. 2014, in SMIR@ SIGIR, 46

Wu, W., Lu, Z., & Li, H. 2012, Regularized mapping to latent structures and its application to web search, Tech. rep., Citeseer

Xu, X., Shimada, A., & Taniguchi, R.-i. 2013, in Multimedia and Expo (ICME), 2013 IEEE International Conference on, IEEE, 1

Yan, R., Kong, L., Li, Y., Zhang, Y., & Li, X. 2011a, in Proceedings of the 20th international conference companion on World wide web, ACM, 157

Yan, R., Wan, X., Otterbacher, J., et al. 2011b, in Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval, ACM, 745

Yan, X., Han, J., & Afshar, R. 2003, in In SDM, SIAM, 166

Yang, S., Lu, W., Yang, D., Yao, L., & Wei, B. 2015, Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies

Zaki, M. J. 2001, Machine learning, 42, 31

Zhang, J., Wang, Y., & Yang, D. 2015, Knowledge-Based Systems, 89, 1

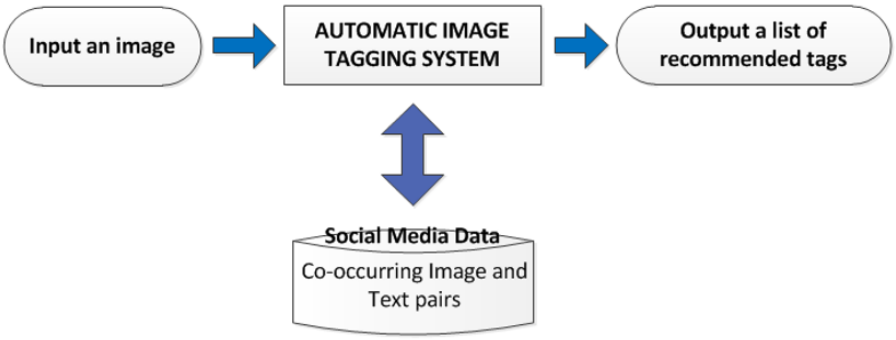
# Appendix A: Pairwise Topic Model III

## A.1 Introduction

Image tagging is a difficult and highly relevant task for many machine learning applications. Specifically, with the emerging of online photo services, the image tagging has become a pre-requisite to make the images searchable and sharable online. The essence of the tag recommendation is to learn the correspondence between the image and tags so as to provide the tag recommendation for a given image. Most state-of-art image annotation technologies use the images with human-labeled tags to learn the correlation between image and tags. However, the correspondence learning through the human-labeled images lacks the generalizability and variety, since the human-labeled images are too small in number to be representative and further the human intentionally generated tags focus on identifying the object instead of elaborating the context. We then propose and study a novel approach that can automatically recommend the tags to an image leveraging social media data. Instead of leveraging the limited number of human labeled images, we try to learn the image and text correspondence through the abundant online social media data with the co-occurring of the image and text. Therefore, the correspondence from the social media data is both generic and elaborate. This is a generic model, and can be used as tag recommendation to other entities.

Tagging, by giving key words to objects has become a popular means to annotate web resources. As more companies begin to provide the online photo services, image tagging has become an ever important component for a searchable image databases. Automatic image tagging, by definition from wiki, is the process by which a computer system automatically assigns metadata in the form of captioning or keywords to a digital image. Online social media services, such as Flickr, allow users to share their photos with other people for social interaction. The users can annotate their photos with their own tags to facilitate the search and sharing. The tagging for the images may give a semantical description for the image, or further reflect some personal perspective and context that is important to the image. However, a large fraction of images online have no tags at all and hence never be retrieved for text queries. Therefore, image tagging has become an essential machine learning task that needs extensive exploration. The existing solution to image annotation problem relies too

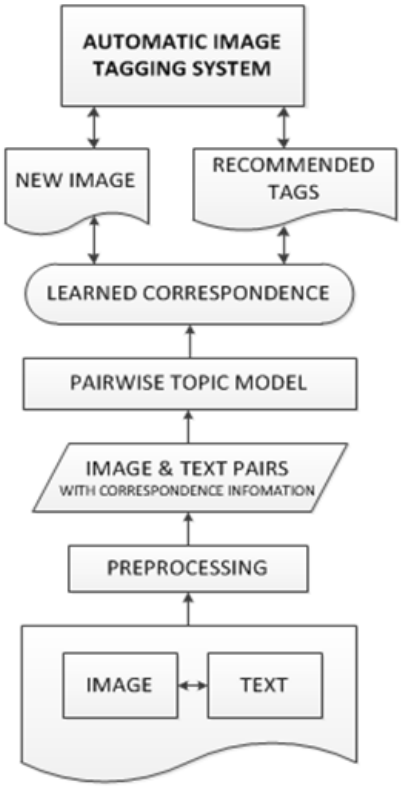
much on the human-labeled data. These methods try to learn the dependency between the image and tag through the human-labeled images. The tag and image correspondence learning through human-labeled images still need to address the following problems. First, human-labeled images may be limited in number and probably not be the representative subset of the whole image population, thus some important correspondence information cannot be captured. Second, in most cases, the intentionally labeled tags lack the semantic and contextual information which is also very important to image tagging. In the human labeled annotations, the tags are mostly the annotation of the objects in the image, without much contextual information. For example, if the system is given a picture with a couple sitting around a table with beautiful kindles and a cake, it probably reminds me of words 'romantic' or 'wedding anniversary' instead of just the 'candle' and 'cake'. The users should have a better experience when they are provided with tags 'romantic' than 'candle'. However, the limited number of the labeled image data and the lack in the variety of the vocabulary in the human labeled data may not suffice enough to capture the correspondence on the atmosphere and contextual lever. That is why we need the unlimited freely available social media data to learn the correspondence between the image and tags. The large amount of data online can thus guarantee that we have enough resources to learn from and have rich contextual information.




Data Flow of the Automatic Tagging System

The automatic image annotation scenario of our system is depicted in Figure 1. When the user uploads an image as a photo or picture, the system outputs the relevant tags related to the image. A detailed automatic tagging system is illustrated in Figure 2. The system is to leverage the co-occurred image and text data to find the correspondence between the features of image and text. With the co-occurring image and text pairs, the representative features for both the image and text need to be extracted respectively. Then the highly relevant

image and text feature pairs can be obtained through dependency measures, such as mutual information. The next step is to use the pairwise topic model to learn the correspondence between the feature pairs. Finally, the tags for the new input images can be obtained via the learned correspondence.




Detailed Automatic Tagging System

## A.2 Related Work

The image annotation problem has been studied extensively. The two widely used methods include the topic modeling and deep learning. Topic Modeling Most works are to learn the image annotation through the image data sets with labeled captions. Almost all the works Putthividhy et al. (2010) Wang et al. (2009) Tian et al. (2015) Tran & Choi (2014) Xu et al. (2013) down this line views the image and text as multi-modal data, each describing the same thing from different perspectives. Thus, they assume both the image and the text share the same latent topic space, and the topics from the image are the same as those from the text. However, the correspondence between the topics of Hu et al. (2014)the image and the text doesn't mean the topics are the same as in the aforementioned example. Further, for the correspondence mining from co-occurred image and text, the image and text may have their own individual line of storytelling. Thus,

some topics from the image and text may overlap, and some may not. Therefore, the existing models lack both the ability to learn the relatedness in a correspondence sense and detect whether correspondence exists between topics of image and text respectively. Deep Learning Model Deep Learning arose as the dominant machine learning techniques in recent years. The deep architect Wang et al. (2009)ure mostly related to our study involves multi-modal deep learning. The two main multi-modal models include the Multimodal learning with Deep Boltzmann Machine Srivastava & Salakhutdinov (2012b) and Multimodal Deep Belief Network Srivastava & Salakhutdinov (2012a). Both models train an energy network to find the correspondence between multi-modal data and thus. Although both models can help retrieve the missing data of one modality given another, the main purposes of these two models are to find the joint representation for the multi-modal data. Another deep learning architecture quite related to our work is to generate image description Karpathy & Fei-Fei (2015)Vinyals et al. (2015) . Karpathy combines the Convolutional Neural Network (CNN) over image regions and the Recurrent Neural Network (RNN) over sentences, and then aligns two modalities through the multimodal embedding. Other deep learning methods to find correspondence between two objects include the deep matching between short texts Bai et al. (2009) Wu et al. (2014). Except for the aforementioned two methods, the other methods to find the correspondence between two heterogeneous objects include to find correspondence between two text clips Wu et al. (2012). The deep learning can learn very complicated relationship; while it is very hard to get the explanation behind it. Further, comparing with the topic modeling, the training of deep learning methods, to large extent, is more complicated than the topic modeling methods.

## A.3 Problem Formulation

Although image annotation has been studied extensively over the years, the main focus is to learn the tag and image correspondence through the human labeled images. To release the dependency on the human-labeled data, we here propose an automatic image annotation problem via existing online social media data. The problem can be formulated as follows: Suppose $P = \{(x, y)|x \in A, y \in B\}$ denotes all the social media resources with co-occurring images and text, where $A$ represents all the images, $B$ represents all the texts, and x and y are two instances from $A$ and $B$. Further, suppose $A_f$ and $B_f$ are the features sets of $A$ and $B$, while $a_f$ and $b_f$ ($a_f \subseteq A_f, b_f \subseteq B_f$) are the feature sets for instances $a$ and $b$ from $A$ and $B$ respectively. Given $P$, we aim to retrieve a list of tags for each new image $x_new$ ($x_{new} \notin A$) and rank them in order of relevance

probability.

## A.4 Methods

This part, we examine in detail the system step by step. As shown in Figure 2, there are mainly two crucial steps in learning the image and text correspondence. The first step is data preprocessing to obtain the correspondence information embedded in the co-occurring image and text. The second step is to explore in detail how they are related. The following two sections will examine in detail how the two steps work.

### A.4.1 Correspondence Extraction via Mutual Information.

The system is to learn the corresponding relationship between the image and text through the social media data. Actually, both the image and text are represented by their features. The correspondence between the image and text is essentially the matching between two types of features of the image and text. To make the illustration simpler, we use the 'word' to refer to the text feature and the 'codeword' to denote the image feature. The 'codeword' is coined from Wang et al. (2009), where each codeword represents each group resulted from the employment of KNN to raw image features with the number of clusters set to the number of codewords needed. We obtain here the social media documents with the co-occurring image and text. Although the co-occurrence of the codeword and word may not guarantee they are related, it is still reasonable to assume that the word co-occur most frequently with a specific codeword may have a higher chance of corresponding to that specific codeword. Accordingly, the modeling of the correspondence within the most frequently co-occurring feature pairs has a better chance to capture the real correspondence. Here we extract the potential codeword and word correspondence using the mutual information. The mutual information between each codeword and word is defined as follows. Mutual Information (codeword, word) =

Thus, the mutual information is calculated as the number of the cooccurring codeword and word divided by the product of the number of codeword and number of text word. Thus, we assume the correspondence relationship exists within one 'codeword,word' pair when the mutual information between the codeword and word is higher than a threshold. In the following generative model, we show how the correspondence can be learned.

### A.4.2 Pairwise Topic Model to capture the image and text correspondence.

For this part, we propose a pairwise topic (PTM) model to learn the image and text correspondence from the co-occurred image and text pairs. Intuitively, the co-occurred image and text should have their own topic distribution respectively, since they are complementary rather than completely the same with each other. Thus, some topics in the text may not have their correspondence in the image and vice versa. Also, the correspondence relationship, instead of modeling as within the same topic space, is to be modeled as the topic pair distribution, with the probability of each pair demonstrating its degree of relatedness. Take the following image and text pair as an example, it is obtained from the CNN news website: http://edition.cnn.com/2015/02/13/health/gallery/outbreak-preparedness/index.html.

The following figure shows one example of co-occurred image and text. Each circle represents the codeword or text word, each rectangular denotes the topic. The black line indicates the correspondence between the image and text topics. The line with color explains the codeword/word distribution within one topic. As we can see the image may contain the topic of the background sky, the fence, the people, the animals, while the text may include the topics of farm, animal, flu and covering. The correspondences exist both between the same topics for animal and covering, and between pairs within different topics, such as fence and farm, fence and animal, covering and flu. Notice the image topic sky has no correspondence with any topic in text, while there is also no corresponding image topic to text topic 'study'.

The following examines the model in detail. Table 1 shows the notation used in the generative process.

1. For each image topic k (k = 1, 2, 3 . . . K),

Draw a topic-codeword distribution for each image topic.

$$\Phi_i \sim \text{Dirichlet}(\beta_1)$$

2. For each text topic k (k = 1, 2, 3 ' K),

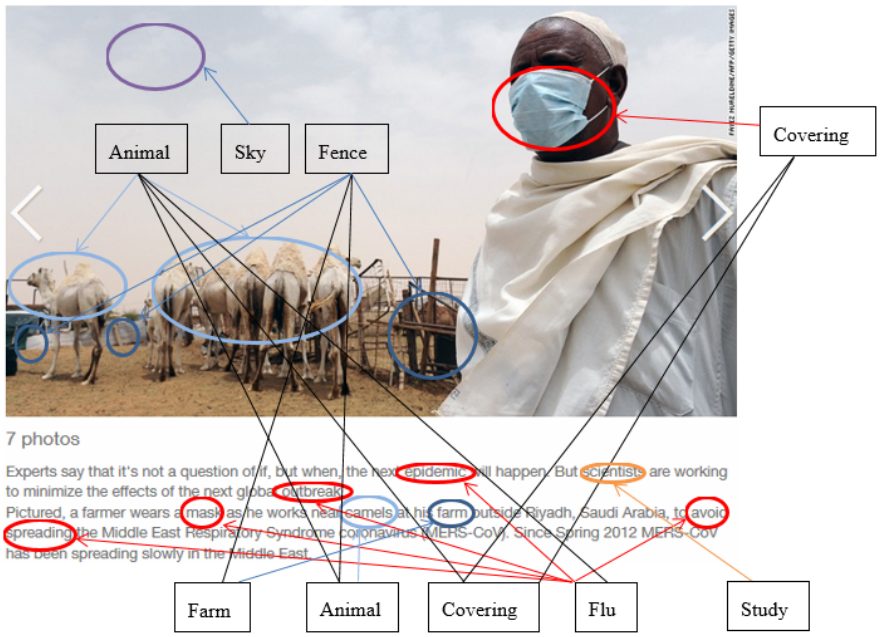
Draw a topic-word distribution for each text topic.

$$\Phi_t \sim \text{Dirichlet}(\beta_2)$$

3. For each image and text topic pair , draw a topic pair-word pair distribution for each pair.

$$\Phi_{it} \sim \text{Dirichlet}(\beta_i t)$$

4. For each document with co-occurred image and text pair d (d = 1, 2, ' D),




Co-occurred Image and Text

**Table A.1:** Annotations in the generative process for co-clustering model.

| Notation | Description |
|---|---|
| $V_1$ | Number of image codewords |
| $V_2$ | Number of text words |
| $K_1$ | Number of topics for image |
| $K_2$ | Number of topics for text |
| $D$ | Number of documents |
| $w_p(w_1, w_2)$ | (codeword,word) pair |
| $z_p(z_1, z_2)$ | Underlying topic pair for each (codeword,word) pair |
| $\alpha_1$ | Dirichlet prior for $\theta_d$ |
| $\alpha_2$ | Dirichlet prior for $\theta_{d,k}$ |
| $\beta_1$ | Dirichlet prior for $\Phi_1$ |
| $\beta_2$ | Dirichlet prior for $\Phi_{k_1,k_2,w}$ |
| $\theta_d$ | Image Topic distribution for document $d$ |
| $\theta_{d,k}$ | Topic transition from image topic $k_1$ to each text topic for document$d$ |
| $\Phi_k$ | Word code distribution for each image topic. |
| $\Phi_{k_p,w_1}$ | Word distribution for each topic given the previous image code word and its topic are w and k respectively. |

a. Draw an image specific topic distribution

$$\theta_i \sim \text{Dirichlet}(\alpha_1)$$

b. Draw an text specific topic distribution

$$\theta_t \sim \text{Dirichlet}(\alpha_2)$$

c. Draw an image and text topic pair distribution

$$\theta_{it} \sim \text{Dirichlet}(\alpha')$$

d. For each image codeword and text word pair,

aa. Assign the correspondence variable $x$ according to the mutual information between the image codeword and the text word.

bb. If $x = 1$, then

Draw the image and text topic pair from the following distribution.

$$z_{it} \sim \text{Categorical}(\theta_i, \theta_t, \theta_{it}) \propto \text{Categorical}(\theta_i)\text{Categorical}(\theta_t)\text{Categorical}(\theta_{it})$$

If $x = 0$, then

Draw the topic of the image codeword from the document-image topic distribution

$$z_1 \sim \text{Categorical}(\theta_1) \quad .$$

Draw the topic of the text word from the document-text topic distribution.

$$z_2 \sim \text{Categorical}(\theta_t)$$

cc. If $x = 1$, then

Draw the 'codeword, word' pair from the following distribution.

$$(w_1, w_2) \sim \text{Categorical}(\Phi_i, \Phi_t, \Phi_{it}) \propto \text{Categorical}(\Phi_i)\text{Categorical}(\Phi_t)\text{Categorical}(\Phi_{it})$$

If $x = 0$, then

Draw the image codeword from the image topic-codeword distribution.

$$w_1 \sim \text{Categorical}(\Phi_i)$$

Draw the text word from the text topic-word distribution

$$w_2 \sim \text{Categorical}(\Phi_t)$$

In the generative process, we introduce a variable x to determine whether the pair mage 'codeword, text word' has the correspondence relationship. We decided here the two words have correspondence if their mutual information exceeds a threshold, otherwise, the two words is independent with each other. Further, for the image codewords with no related text words and the text words without image corresponding words, their underlying topics are generated independently. Also the image codeword and the text word themselves

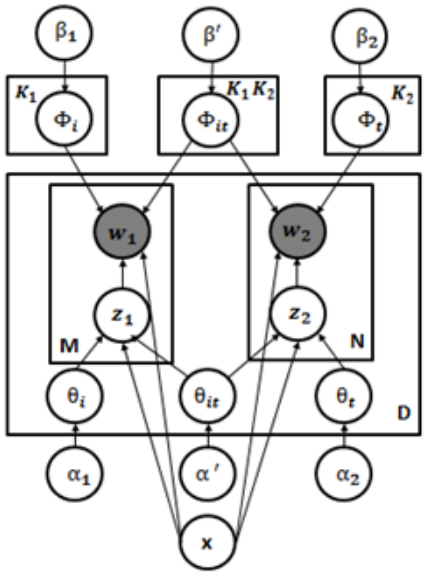
are generated from their hidden topics respectively. Second, when the image codeword and text word have correspondence relationship, their underlying topics are determined not only by the image and text topic distributions, but also their relatedness, which is captured by the image topic and text topic pair distribution.

The joint probability of the aforementioned model can be illustrated as following.

$$
\begin{aligned}
& p(W_p, Z_p, \theta_1, \theta_2, \theta_1 2, \Phi_1, \Phi_2, \Phi_{12} | \alpha_1, \alpha_2, \alpha_{1,2}) \\
=& \prod_{d=1}^{D} \frac{\Gamma(\sum_{k_1=1}^{K_1} \alpha_1)}{\prod_{k_1=1}^{K} \Gamma(\alpha_1)} \prod_{k_1=1}^{K_2} \theta_i^{\alpha_1 - 1} \prod_{d=1}^{D} \frac{\Gamma(\sum_{k_2=1}^{K_2} \alpha_2)}{\prod_{k_2=1}^{K} \Gamma(\alpha_2)} \prod_{k_2=1}^{K_2} \theta_t^{\alpha_2 - 1} \prod_{d=1}^{D} \frac{\Gamma(\sum_{k_1=1,k_2=1}^{K_1,K_2} \alpha_{12})}{\prod_{k_1=1,k_2=1}^{K_1,K_2} \Gamma(\alpha_{12})} \prod_{k_1=1}^{K_1} \prod_{k_2=1}^{K_2} \theta_m^{\alpha^{12} - 1} \\
& \prod_{k_1=1}^{K_1} \frac{\Gamma(\sum_{w_2=1}^{W_2} \beta_2)}{\prod_{w_2=1}^{W_2} \Gamma(\beta_2)} \prod_{k_1=1}^{K_1} \prod_{k_2}^{K_2} \frac{\Gamma(\sum_{w_1}^{W-1} \sum_{w_2}^{W_2} \beta_{12})}{\sum_{w_1=1}^{W_1} \sum_{w_2=1}^{W_2} \Gamma(\beta_{12})} \prod_{w_1=1}^{W_1} \prod_{w_2=1}^{W_2} \Phi_m^{\beta_{12} - 1} \\
& \prod_{d=1}^{D} \prod_{k_1=1}^{K_1} \theta_1^{n_{k_1}} \prod_{d=1}^{D} \prod_{k_1=1}^{K_1} \theta_2^{n_{k_2}} \prod_{d=1}^{D} \prod_{k_1=1}^{K_1} \prod_{k_2=1}^{K_2} \theta_m^{n_{k_1,k_2}} \\
& \prod_{k_1=1}^{K_1} \prod_{w_1=1}^{W_1} \Phi_i^{n_{k_1,w_1} - 1} \prod_{k_2=1}^{K_2} \prod_{w_2=1}^{W_2} \Phi_t^{n_{k_2,w_2} - 1} \prod_{k_1=1}^{K_1} \prod_{k_2=1}^{K_2} \prod_{w_1}^{W_2} \prod_{w_2}^{W_2} \Phi_{it}^{n_{k_p,w_p}} \\
=& (\frac{\Gamma(\sum_{k_1=1}^{K_1} \alpha_1)}{\prod_{k_1=1}^{K_1} \Gamma(\alpha_1)})^D \prod_{d=1}^{D} \prod_{k_1=1}^{K_1} \theta_i^{\alpha_1 + n_{k_1} - 1} \\
& (\frac{\Gamma(\sum_{k_2=1}^{K_2} \alpha_2)}{\prod_{k_2=1}^{K_2} \Gamma(\alpha_2)})^D \prod_{d=2}^{D} \prod_{k_2=1}^{K_2} \theta_i^{\alpha_2 + n_{k_2} - 1} \\
& (\frac{\Gamma(\sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \alpha_{12})}{\prod_{k_1=1}^{K_1} \prod_{k_2=1}^{K_2} \Gamma(\alpha_{12})})^D \prod_{d=2}^{D} \prod_{k_1=1}^{K_1} \prod_{k_2=1}^{K_2} \theta_{it}^{\alpha_{12} + n_{k_1,k_2} - 1} \\
& (\frac{\Gamma(\sum_{k_1=1}^{K_1} \beta_1)}{\prod_{w_1=1}^{W_1} \Gamma(\beta_1)})^{K_1} \prod_{k_1}^{K_1} \prod_{w_1}^{W_1} \Phi_i^{\beta_1 + n_{k_1,w_1} - 1} \\
& (\frac{\Gamma(\sum_{k_2=1}^{K_2} \beta_2)}{\prod_{w_2=1}^{W_2} \Gamma(\beta_2)})^{K_2} \prod_{k_2}^{K_2} \prod_{w_2}^{W_2} \Phi_i^{\beta_2 + n_{k_2,w_2} - 1} \\
& (\frac{\Gamma(\sum_{w_1=1}^{W}) \sum_{w_2}^{W_2} \beta_{12}}{\prod_{w_1=1}^{W} \prod_{w_2=1}^{W_2} \Gamma(\beta_{12})})^{K_1 K_2} \prod_{k_1=1}^{K_1} \prod_{k_2=1}^{K_2} \prod_{w_1}^{W_1} \prod_{w_2}^{W_2} \Phi_{it}^{\beta^{12} + n_{k_p,w_p} - 1}
\end{aligned}
\tag{A.1}
$$

The graphical illustration for the generative process is shown in the following figure.

Therefore, given the co-occurred image and text pairs, the features of the image and text and the correspondence between the two feature sets, the pairwise topic model can find: The topic distribution for image and text respectively for each image and each text. The topic pair distribution for each co-occurred image and text. The codeword distribution of each image topic and the word distribution for each text topic. The 'codeword, word' pair distribution under each topic pair between the image and the text. Overall, the PTM differs from




Pairwise Topic Model

the existing ones in that: a. the pairwise topic model can deal with the situation where the content of the image and the text are complementary rather than the same with each other. Further, b. the model captures correspondence relationship in two spaces rather than in one common space.

## A.5 Inference

We use Gibbs sampling to perform model inference. Due to the space limit, we leave out the derivation details and only show the sampling formulas.

**Table A.2:** Annotations for the inference of co-clustering model.

| Notation | Description |
|---|---|
| $n_{d,z_1}^{\neg i}$ | Number of image code words assigned to topic in document d except for the current image code word |
| $n_{d,z_2}^{\neg i}$ | Number of text code words assigned to topic in document d except for the current image code word |
| $n_{d,z_p}^{\neg i}$ | Number of image and text code word pairs assigned to topic pair in document d except for the current image code word and text word pair |
| $n_{z_1,w_1}^{\neg i}$ | Number of image codeword assigned to image topic except for the current image codeword |
| $n_{z_2,w_2}^{\neg i}$ | Number of text word assigned to text topic except for the current text word |
| $n_{z_p,w_p}^{\neg i}$ | Number of image and text word pair assigned to topic pair except for the current pair |
| $z_p(z_1, z_2)$ | Underlying topic pair for each (codeword,word) pair |

The formula we use to do the inference is: For a specific 'codeword,word' pair, if x =1,

$$
\begin{aligned}
& p(z_{i_1}, z_{i_2}|W, Z_{\neg i}, \alpha_1, \alpha_2, \beta_1, \beta_2) \\
\propto & \frac{\alpha_1 + n_{d,z_i}^{\neg i}}{\sum_{k_1=1}^{K} \alpha_1 + \sum_{k_1=1}^{K_1} n_{d,k_1}^{\neg i}} \frac{\alpha_2 + n_{d,z_2}^{\neg i}}{\sum_{k_2=1}^{K} \alpha_2 + \sum_{k_2=1}^{K_2} n_{d,k_2}^{\neg i}} \\
[h!] & \frac{\alpha_{12} + n_{d,(z_1,z_2)}^{\neg i}}{\sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2} \alpha_{12} + \sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2} n_{d,(k_1,k_2)}^{\neg i}} \\
& \frac{\beta_1 + n_{z_1,w_1}^{\neg i}}{\sum_{w_1}^{W_1} \beta_1 + \sum_{w_1=1}^{W_1} n_{z_1,w_1}^{\neg i}} \frac{\beta_2 + n_{z_2,w_2}^{\neg i}}{\sum_{w_2}^{W_2} \beta_2 + \sum_{w_2=1}^{W_2} n_{z_2,w_2}^{\neg i}} \\
& \frac{\beta_{12} + n_{z_p,w_p}^{\neg i}}{\sum_{w_1}^{W_1}\sum_{w_2}^{W_2} \beta_{12} + \sum_{w_1=1}^{W_1}\sum_{w_2=1}^{W_2} n_{z_p,w_p}^{\neg i}}
\end{aligned}
\tag{A.2}
$$

If x = 0,

$$
\begin{aligned}
& p(z_{i_1}, z_{i_2}|W, Z_{\neg i}, \alpha_1, \alpha_2, \beta_1, \beta_2) \\
[h!] \propto & \frac{\alpha_1 + n_{d,z_i}^{\neg i}}{\sum_{k_1=1}^{K} \alpha_1 + \sum_{k_1=1}^{K_1} n_{d,k_1}^{\neg i}} \frac{\alpha_2 + n_{d,z_2}^{\neg i}}{\sum_{k_2=1}^{K} \alpha_2 + \sum_{k_2=1}^{K_2} n_{d,k_2}^{\neg i}} \\
& \frac{\beta_1 + n_{z_1,w_1}^{\neg i}}{\sum_{w_1}^{W_1} \beta_1 + \sum_{w_1=1}^{W_1} n_{z_1,w_1}^{\neg i}} \frac{\beta_2 + n_{z_2,w_2}^{\neg i}}{\sum_{w_2}^{W_2} \beta_2 + \sum_{w_2=1}^{W_2} n_{z_2,w_2}^{\neg i}}
\end{aligned}
\tag{A.3}
$$

## A.6 Automatic Tagging with PTM

The PTM can help find the significant correspondence between the image and text to facilitate the image understanding. Through the PTM, we can not only find the topic distribution for both the image and text pair, but also the correspondence between the topics of image and text. To find the representative words for a given an image, we first use the topic proportion information of the image to find their most related topics of the corresponding text. Then, we use the words distribution of the topic to obtain the most representative words. The probability of the text word given the image code word is as following.

$$
\begin{aligned}
& p(w_2|w_{11}, w_{12}, \ldots, w_{1n}) \\
[h!] = & \frac{p(w_2, w_{11}, w_{12}, \ldots, w_{1n})}{p(w_1, w_{11}, w_{12}, \ldots, w_{1n})} \\
= & \frac{\sum_{k_1=1,k_2=1}^{K_1\dot{K}_2} \sum_{i=1}^{N} p(w_{i1}, w_2, z_p, \alpha_1, \alpha_2, \beta_1, \beta_2)}{\sum_{v_2=1}^{V_2} \sum_{k_1=1,k2=1}^{K_1\dot{K}_2} \sum_{i=1}^{N} p(w_{1i}, w_2, z_p, \alpha_1, \alpha_2, \beta_1, \beta_2)}
\end{aligned}
\tag{A.4}
$$

Here, we assume the independency within the text words and only consider the one-to-one correspondence

between the image code and text words. Thus, the probability for each text word given all the code words of an image can be calculated through the formula. Thus, the words can be ranked for a specific image and the word list thus can be used as the output for the automatic image annotation system of a given image.

In the Appendix, we propose a system for the automatic image annotation leveraging the social media data. Rather than using the human-labeled annotation data to extract the correspondence between two data modal describing the same object, this system uses the easily available online co-occurred image and text data to mine the correspondence. By first obtaining the co-occurred image and text from the social media, such as the news, blogs and social network, the PTM is to find the respective topics of both the image and text and their correspondence. The model learned can be used to provide a ranked list of tags to a new image. The proposed pairwise topic model can be applied to any two heterogeneous object correspondence mining.

# Vita

## Xiaoli Song

### Education

- Drexel University, Philadelphia, Pennsylvania USA
  - Ph.D., Information Science, December 2016
- Beijing Language and Cultural University, Beijing, China
  - M.S., Information Science, June 2008

### Publications

- "Semantic Pattern Mining for Text Mining", Bigdata 2016IEEEInternationalConference, 2016
- "Pairwise Topic Model and its Application to Topic Transition and Evolution", Bigdata 2016IEEEInternationalConference, 2016
- "Pairwise Topic Model via relation extraction", Bigdata 2014IEEEInternationalConference, 2014

### Teaching Experience

- **Teaching Assistant** *Sep. 2011-Sep. 2012*

  *Sep. 2011-Sep. 2012* –Nursing Informatics (INFO204)
  *Sep. 2012-Dec. 2012* – Computer Networking Technology (INFO 330)