EXTENDING TOPIC MODELS WITH SYNTAX AND SEMANTICS RELATIONSHIPS

ELNAZ DELPISHEH

A DISSERTATION SUBMITTED TO

THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

YORK UNIVERSITY

TORONTO, ONTARIO

May 2015

Abstract

Probabilistic topic modeling is a powerful tool to uncover hidden thematic structure of documents. These hidden structures are useful for extracting concepts of documents and other data mining tasks, such as information retrieval. Latent Dirichlet allocation (LDA) [16], is a generative probabilistic topic model for collections of discrete data such as text corpora. LDA represents documents as a bag-of-words, where the important structure of documents is neglected. In this work, we proposed three extended LDA models that incorporates syntactic and semantic structures of text documents into probabilistic topic models.

Our first proposed topic model enriches text documents with collapsed typed dependency relations to effectively acquire syntactic and semantic dependencies between consecutive and nonconsecutive words of text documents. This representation has several benefits. It captures relations between consecutive and nonconsecutive words of text documents. In addition, the labels of the collapsed typed dependency relations help to eliminate less important relations, i.e., relations involving prepositions. Moreover, in this thesis, we introduced a method to enforce topic similarity to conceptually similar words. As a result, this algorithm leads to more coherent topic distribution over words.

Our second and third proposed generative topic models incorporate term importance into latent topic variables by boosting the probability of important terms and consequently decreasing the probability of less important terms to better reflect the themes of documents. In essence, we assign weights to terms by employing corpus-level and document-level approaches. We incorporate term importance using

a nonuniform base measure for an asymmetric prior over topic term distributions in the LDA framework. This leads to better estimates for important terms that occur less frequently in documents. Experimental studies have been conducted to show the effectiveness of our work across a variety of text mining applications.

Furthermore, we employ our topic models to build a personalized content-based news recommender system. Our proposed recommender system eases reading and navigation through online newspapers. In essence, the recommender system acts as filters, delivering only news articles that can be considered relevant to a user. This recommender system has been used by The Globe and Mail, a company that offers most authoritative news in Canada, featuring national and international news.

To my dearest parents and my beloved Babak.

Acknowledgements

Writing this thesis has been a process which involved many individuals whom I appreciate. First, I would like to thank my supervisor, Dr. Aijun An, for her endless guidance and all the useful discussions and brainstorming sessions. Second, I would like to thank my committee members, Dr. Nick Cercone and Dr. Steven Wang for supporting this dissertation and providing intellectual inputs. I would also like to thank my examination committee members, Dr. Hui Jiang, Dr. Susan McGrath, and Dr. Charles Ling for taking the time to read my thesis and for their valuable feedbacks on this dissertation.

The members of the data mining lab has contributed to the majority of my graduate career at York University. The group has been a source of friendships as well as great collaborators. I am also thankful to the administrative and technical staff of our department for their finest support.

I am extensively indebted everything in my life to my family and friends. I would like to thank my parents who taught me faith, kindness, and hard work; my sisters and brother who taught me love. I would also like to thank my partner, Babak, for his endless love, patience, and support that have meant everything to me and made this all possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The increasing amount of electronic texts demands better tools for searching, exploring, and organizing document collections. Previously, texts were collected and stored in large text repositories and retrieved by a set of keywords. Documents were seldom analysed using their themes, because there were very few technologies to extract their thematic structures. To remedy the situation *topic detection* techniques have emerged. Major categories of these techniques include text categorization, text clustering, keywords extraction, keywords clustering, and topic modeling. In this work, we focus on *topic modeling*. Topic modeling is a powerful statistical tool to uncover hidden thematic structures of documents, also called *"topics"*. These topic models facilitate document summarization and organization in a variety of applications in information retrieval, vision, social network analysis, and text mining [11, 16, 40, 46, 77].

However, the discovered topics by existing topic modeling techniques may not always well correspond to the themes of the documents. The algorithms developed in this dissertation allow integrating semantic and syntactic structures of documents

into topic models to influence the discovered topics. These algorithms are designed as extra modules that can be incorporated into topic models.

1.1 Motivations for topic modeling

Consider the following sample text from an article of The Globe and Mail[1]:

> *"Tim Cestnick is president of Water Street Family Offices, and author of several tax and personal finance books. There's nothing like an education about money while you're still young. Aside from helping young people make wise decisions about their investments, starting young can lead to a much larger portfolio down the road. Time is an investors greatest ally. As we head into registered retirement savings plan (RRSP) season, encourage your adult children to contribute to their RRSPs. After raising eyebrows for speaking to a journalist while on a leave to seek help for his addiction issues, Toronto Mayor Rob Ford gave another media interview insisting he is undergoing treatment in a facility that costs as much as six figures. It's worth every dime, every dime, he told the Toronto Sun. A hundred grand is cheap. It's a steal. Mr. Ford's two interviews to the Sun come amid mounting questions about the sincerity of his claim that he is getting professional help. "*

Assume that our goal is to describe the common themes present in the sample text. A simple scalable approach is to consider the word frequencies throughout the text [55]. The sample text has been analyzed and the result is shown in Figure 1.1 that is a word cloud[2] of the text document, where more frequent words appear larger. Although this representation gives us a more understandable image of the text, this representation yields little insight about different themes of the sample text.

[1]The Globe and Mail offers the most authoritative news in Canada, featuring national and international news. http://www.theglobeandmail.com/

[2]http://www.wordle.net/

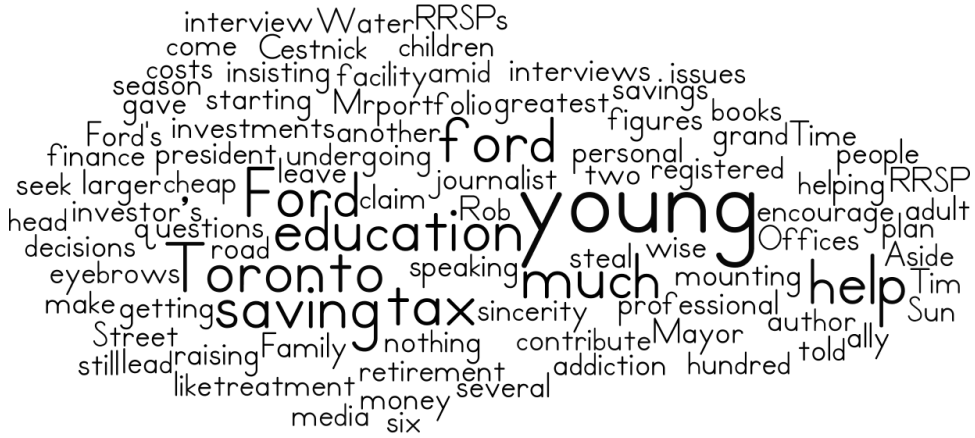

Figure 1.1: The word cloud representation of the sample text. Note that words *young*, *ford*, *tax*, and *education* are the most frequent words in the text.

A different approach is to *hard cluster* [3] the text by its word content, hoping the cluster reflects a theme of the text. However, the sample text does not fit into a single theme. For example, the sentence *"Nowadays, people are mainly concerned about their savings and the government's contribution through the Canada Education Savings Grant."* exhibits two different themes, *savings* and *education*.

Alternatively, *soft clustering*, also referred to as *fuzzy clustering*, is used to assign the sample text into several clusters, where the text has fractional membership in those clusters. Although this approach is useful in probabilistic text modeling, it is incomplete. Soft clustering represents each document in the training set as a list of numbers (the mixing proportions for themes), and there is no generative probabilistic model for these numbers. As a consequence, this approach does not accurately assign probabilities to a document outside the training set [16].

Latent topic modeling [16] solves this problem by assuming that each document is represented as a random mixture over latent themes, called *topics*, where each

topic is characterized by a distribution over words. A good probabilistic topic model of a collection of text documents assigns high probabilities to the documents of the collection as well as to other similar documents [16, 40]. If we have K topics, the probability of the ith word in a given document is

$$p(w_i) = \sum_{t=1}^{K} p(w_i|z_i = t)p(z_i = t), \tag{1.1}$$

where z_i is a latent variable indicating the topic from which the ith word is drawn and $p(w_i|z_i = t)$ is the probability of the word w_i under the tth topic. $p(z_i = t)$ is the probability of choosing a word from topic t in the current document. Intuitively, $p(w|z)$ indicates the importance of word w to topic z. $p(z)$ indicates the probability of a particular topic given a document. In the following section, we explain *Latent Dirichlet Allocation (LDA)* that is a generative latent topic model.

1.2 Latent topic modeling

Latent topic models assume a corpus is a collection of text documents. Text documents can include multiple topics, addressed by particular sets of words. Latent topic models, such as Probabilistic Latent Semantic Indexing (PLSI) [45], and Latent Dirichlet Allocation (LDA) [16] consider a document to be a weighted mixture of topics, where each topic is a multinomial distribution over words. Due to the shortcomings of PLSI, described in detail in Chapter 2, in this thesis, we focus on the Latent Dirichlet Allocation (LDA), proposed by Blei *et al.* [16]. LDA is a generative probabilistic topic model for collections of discrete data such as text corpora. For example, consider a collection of The Globe and Mail articles that appeared on

Topic 1 **(Tax)**	**Topic** 2 **(Children)**	**Topic** 3 **(Education)**	**Topic** 4 **(Mayor)**
tax	young	education	ford
income	children	school	mayor
retirement	baby	teacher	city
pension	kids	government	toronto
plan	youth	math	rob
savings	parents	student	councillor
financial	school	union	miller
money	mother	parents	doug
rrsp	age	class	campaign
contribution	boy	public	crack
...	...	...	...

Table 1.1: Top 10 terms of the most probable topics of The Globe and Mail collection. Note that labels *Tax*, *Children*, *Education*, and *Mayor* are manually assigned.

The Globe and Mail newswire during the period between January 2010 to March 2014. This corpus contains $142,163,909$ news articles. Using topics to explore the articles at a broad level reveals different aspects of the collection [11]. Some of the themes might correspond to the topics of the articles, i.e., tax, children, education, and mayor. We could zoom in on a topic of interest to review details of the topic. For example, Table 1.1 shows four most probable topics of The Globe and Mail corpus. Note, in topic *"Mayor"*, words *"ford"* and *"city"* gain high probabilities.

The word cloud representation of four most probable topics of The Globe and

Figure 1.2: The word cloud representation of topic *Tax*.

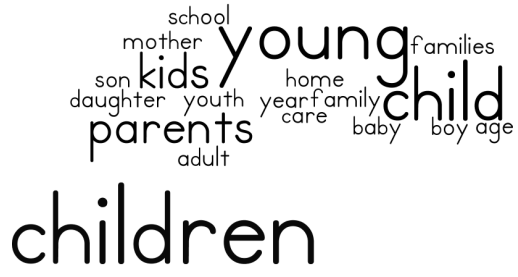

Figure 1.3: The word cloud representation of of topic *Children*.

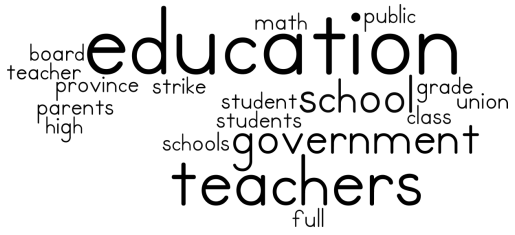

Figure 1.4: The word cloud representation of topic *Education*

Figure 1.5: The word cloud representation of topic *Mayor*.

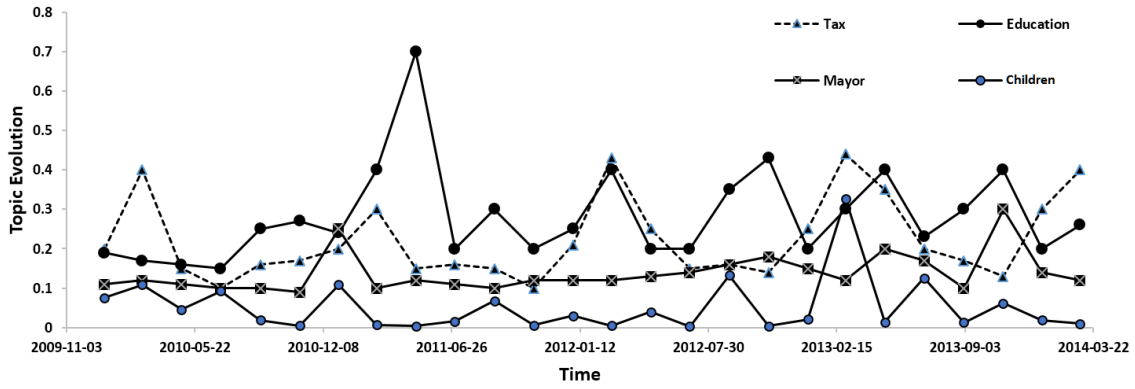

Figure 1.6: The Globe and Mail topic evolution over time. Notice the popularity of topic *Tax* in February that is the deadline of filing tax return documents in Canada.

Mail corpus is illustrated in Figures 1.2, 1.3, 1.4, and 1.5, where a word w with high probability $p(w|z)$ in a given topic (z) appears larger. Note that topic labels are manually assigned. These topics, discovered by LDA, provide a much richer understanding than the aforementioned solely word frequency representation of text documents.

In addition, we could navigate through time to reveal how these topics have evolved to see the popularity of a specific topic over a time period, as illustrated in Figure 1.6.

Besides fundamental concepts of purely exploratory analysis of probabilistic topic modeling, topic models have been applied to a wide variety of tasks in information retrieval [76, 90], vision [38], social network analysis [8, 23, 56, 69], text classification [51], machine translation [35, 89], and recommendation [48, 67, 91, 92].

1.3 Open issues and our contributions

Most topic models consider documents to be a weighted mixture of topics, where each topic is a multinomial distribution over words. Text documents are the only observed data in most conventional topic models. Some words in a discovered topic are ambiguous and can have multiple meanings. To identify the correct meaning of each word, one needs to consider other words in the topic. For example, the word *"class"* in topic 3, shown in Table 1.1, has many meanings. In one meaning, a *"class"* is a collection of things sharing a common attribute, i.e., a group of students who are taught together: *"I was late for a class."* In the second, the word refers to the system of ordering a society in which people are divided into sets based on perceived social or economic status: *"People who are socially disenfranchised by class."* Observing other words in this topic, such as *"education"*, *"school"*, and *"teacher"*, helps to identify the correct meaning of the word *"class"* that is *"a group of students who are taught together."* In order to obtain the correct meaning of the words in text documents, we need to capture relations between consecutive and nonconsecutive words. Although, the n-gram topic model [80] captures dependencies between words of a sentence, it fails to consider dependencies between nonconsecutive words with a distance; thus, the n-gram topic model is limited to capturing dependencies between consecutive words. In this thesis, we solve this problem by building a Head-Driven Phrase Structure Grammar (HPSG)-based topic model. We effectively acquire syntactic and semantic dependencies between words and incorporate them into our HPSG-based topic model. Our experimental studies show that our proposed model works considerably better than similar LDA-based topic models.

Moreover, text documents consist of words with possible conceptual similarities, called *synonyms*, defined in lexical resources like WordNet [60]. It is reasonable to expect the distribution of topics over synonymous words to be similar. For example, in topic 2, shown in Table 1.1, synonymous words such as *"kids"*, *"children"*, and *"baby"* should have similar higher probabilities, and words such as *"school"*, and *"grown-up"* should have similar lower probabilities. In this thesis, we propose an algorithm to enforce similar topic distribution over conceptually similar words.

In addition, existing topic models use fixed symmetric priors, and consider only frequencies of terms in the corpus to estimate posteriors of latent variables [82]. This representation makes sense from a point of view of computational efficiency [80], but it does not utilize additional information about how important terms are in the context of a corpus, to properly reflect the thematic structures of documents. Moreover, topics estimated by LDA for infrequently occurring words are usually unreliable [70]. As a result, most inferred topic distributions over terms contain terms that are irrelevant to the topic and should not appear with a high probability in the topic. For instance, in topic 3, shown in Table 1.1, it is reasonable for important terms such as *"student"* and *"class"* to have high probabilities, but less important terms like *"union"* to have a low probability. We build a more robust topic model by incorporating additional information about term importance in a document into a topic model framework to boost the probability of important terms and to consequently decrease the probability of less important terms.

Furthermore, we integrate our topic model using term importance into the HPSG-based topic model. The consecutive and nonconsecutive relations between words are extracted by employing syntax and semantic analysis. We further assign importance

weights to those relations using the context of the corpus or an external data source. Then, these weights are incorporated into the HPSG-based topic model to increase the probability of important relations and to consequently decrease the probability of less important relations. Experimental studies show the effectiveness of our method.

Finally, we propose a news recommender system leveraging our topic models. We build an automated recommender system that is able to filter news articles and make recommendations based on users' preferences. We use topic models to identify the thematic structure of the corpus. These themes are incorporated into a content-based recommender system to filter news articles that contain themes that are of less interest to users and to recommend articles that are thematically similar to users' preferences. This work has been done in our collaboration with the data scientists at The Globe and Mail.

In summary, to address the above open issues, the main contributions in the dissertation are as below:

- We propose a novel topic model, called *the HPSG-based Topic Model*, to consider syntactic and semantic structures of text documents in probabilistic topic models.

- We propose an algorithm to enforce similar topic distribution over conceptually similar words.

- We propose two novel generative topic models, called *Topic Model using Corpus-level Term Importance (TMCTI)* and *Topic Model using Document-level Term Importance (TMDTI)*, that do not consider a fixed distribution prior over terms

but rather we adjust the prior by employing additional information about the composition of terms that should have high or low probabilities in topics.

- We extend our HPSG-based topic model by using TMCTI and TMDTI approaches to consider importance of consecutive and nonconsecutive relations in text documents.

- We conduct extensive experiments to evaluate the proposed topic modeling techniques. Our evaluation results show that our techniques have the following benefits. First, they lead to a more robust topic model that significantly improves topic models in terms of perplexity. Second, our TMCTI and TMDTI modeling techniques lead to significantly better topic models in terms of topic coherence. Furthermore, the resultant topic models show better performance in data mining tasks, such as text classification. In addition, integrating syntax and semantics relationships into topic models enhances understandability of the discovered topics.

- We apply probabilistic topic modeling techniques to the development of a personalized content-based news recommender system for The Globe and Mail, and demonstrate that the use of topics to represent documents significantly improves the recommendation performance over the bag-of-words based document representation method.

1.4 Thesis outline

We begin this thesis by formally defining the LDA model and explaining how topics are learned from data in Chapter 2. This chapter also discusses some of the general problems and issues related to topic modeling.

In Chapter 3 various topic modeling approaches proposed in the literature, how they aim to tackle the issues of topic models, their advantages and drawbacks are explained.

Chapter 4 introduces our first proposed probabilistic topic model, the HPSG-based topic model, that enriches text documents with collapsed typed dependency relations to effectively acquire syntactic and semantic dependencies between consecutive and nonconsecutive words of text documents. In addition, in this chapter we propose to enforce coherent topic assignments for conceptually similar words by generalizing words with their synonyms. This chapter also compares our approach to the other LDA-based approaches in terms of perplexity, stability, coherence, and accuracy.

Chapter 5 presents our two proposed generative topic models, the topic model using corpus-level term importance (TMCTI) and the topic model using document-level term importance (TMDTI), that incorporate term importance into latent topic variables by boosting the probability of important terms and consequently decreasing the probability of less important terms to better reflect the themes of documents. In this chapter, we assign weights to terms by employing corpus-level and document-level approaches. We incorporate term importance using a nonuniform base measure for an asymmetric prior over topic-term distributions in the LDA framework. This leads to better estimates for important terms that occur less frequently in documents.

We show the performance of our topic models in text mining tasks. Moreover, in this chapter, we investigate the extension of the HPSG-based topic model by using phrase importance scores.

Chapter 6 we employ topic models to design a content-based news recommender system that issues the most relevant news article recommendations to users according to their personal read article history. This application has been developed in collaboration with The Globe and Mail data scientists.

Chapter 7 concludes the thesis, summarizing the contributions, and describing directions for further research.

Chapter 2

Probabilistic Topic Models

In this section, we explain fundamental probabilistic topic models for text documents. These topic models include: *the Unigram Model*, *the Mixture of Unigrams, Probabilistic Latent Semantic Indexing*, and *Latent Dirichlet Allocation*. We also highlight their key similarities and differences.

2.1 The Unigram Model

The Unigram Model assumes that a *corpus* is a collection of D documents, where each document d consists of a list of words, denoted by $d = \{w_1, w_2, \cdots, w_{|d|}\}$. This model generates documents by drawing the words independently from a single multinomial distribution [16]. Furthermore, this model assumes that the words are generated not only independently of the length of the document, but also of their positions in the document. Thus,

$$p(d) = \prod_{i=1}^{|d|} p(w_i), \tag{2.1}$$

where $p(w_i)$ is the probability of w_i, which can be estimated as the number of times word w_i occurs in a training corpus divided by the word occurrences in the corpus.

This basic model reduces each document to a vector of real numbers, each of which represents ratios of word counts in the document to the entire corpus. However, this model reveals little about inter document statistical structure. It ignores the correlation between words in neighboring positions, as well as the topic of the document. To resolve these issues, Nigam *et al.* [68] proposed to augment the model with a random topic variable, explained in the following section.

2.2 The Mixture of Unigrams

Nigam *et al.* [68] assume that every document is generated according to a probability distribution defined by a set of parameters, i.e., a random topic variable z. In the mixture of unigrams, each document is generated by first choosing a topic z and then generating $|d|$ words independently from the conditional multinomial $p(w_i|z)$. $p(w_i|z)$ is computed by dividing the number of times word w_i occurs in topic z by the number of word occurrences in topic z. The probability of the document is:

$$p(d) = \sum_{z} p(z) \prod_{i=1}^{|d|} p(w_i|z). \tag{2.2}$$

However, the assumption made by this model, each document is generated from exactly one topic, is not generally true. In reality, each document may contain multiple topics.

2.3 Probabilistic Latent Semantic Indexing

Probabilistic Latent Semantic Indexing (PLSI), proposed by Hofmann *et al.* [45], removes the simplifying assumption made in the mixture of unigrams model, that each document has only one topic. The PLSI model assumes that each document may contain multiple topics, denoted by $Z = \{z_1, z_2, \cdots, z_K\}$. For a particular document d, $p(d)$ is the probability of selecting document d, $p(z|d)$ is the probability of topic $z \in Z$ under document d, also referred to as the mixture weights of the topics for document d, and $p(w|z)$ is the probability of word w under topic z. In addition, this model assumes that a document d and word w are conditionally independent given a topic z. The PLSI model is defined as

$$p(d, w) = p(d) \sum_{z \in Z} p(w|z)p(z|d). \tag{2.3}$$

The shortcomings of PLSI come from the use of *"only"* training documents to obtain distribution of topics over words. As a result, the model learns the topic mixture only for those documents in the training set. Thus, there is no way to assign a probability to a previously unseen document. Moreover, given the fact that the number of topics is explicitly linked to the training documents, this number grows linearly with the growth of the number of training documents. The parameters for a K-topic PLSI model are K multinomial distributions of size V and D mixtures over the K hidden topics, where V is the size of the set of unique vocabulary words contained in the corpus, and D is the number of documents. This gives $KV + KD$ parameters and therefore linear growth in D. The linear growth in parameters suggests that the model is prone to overfitting [16].

These two problems are overcome by *Latent Dirichlet Allocation (LDA)* [16]. LDA as explained in the following section, is a generative model and generalizes easily to new documents. Furthermore, LDA treats the topic mixture weights as a K parameter hidden random variable rather than a large set of individual parameters which are explicitly linked to the training set. Thus, the $K + KV$ parameters in a k-topic LDA model do not grow with the size of the training corpus [16].

2.4 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA), proposed by Blei *et al.* [16], is a generative probabilistic model for collections of discrete data such as text corpora. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA also assumes that a corpus is a collection of D documents. Let $\mathcal{D} = \{w_1, w_2, \cdots, w_N\}$ represent a corpus of length N, resulting from the concatenation of the D documents which contains N words in total, where each word w_i belongs to a set of unique vocabulary words of size V [1]. LDA assumes that each word $w_i \in \mathcal{D}$ is associated with a latent topic variable z_i where $i \in \{1, 2, \cdots, N\}$. Each of these topics $t = 1 \cdots K$ is associated with a multinomial $\vec{\Phi}_t$ over V vocabulary words, such that $p(w_i|z_i = t) = \Phi_{z_i,w_i}$. Each $\vec{\Phi}_t$ is generated from a Dirichlet distribution with prior $\vec{\beta}$. Also, each document d is associated with a multinomial distribution $\vec{\Theta}_d$ over K topics, such that $p(z_i = t|d) = \Theta_{d,z_i}$, generated from a Dirichlet distribution with prior $\vec{\alpha}$. To discover the set of topics used in the

[1] This set of vocabulary words can be the set of unique words contained in the corpus with removal of stop words.

corpus $\mathcal{D}$, the objective is (1) to obtain an estimate of $\underline{\Phi}$, where $\underline{\Phi} = \{\vec{\Phi}_t\}_{t=1}^{K}$, that is the term distribution for each topic, and (2) to obtain an estimate of $\underline{\Theta}$, where $\underline{\Theta} = \{\vec{\Theta}_d\}_{d=1}^{D}$, that is the topic distribution for each document. LDA is one such model.

In LDA, each document d is generated by first drawing a distribution over K topics with parameters $\vec{\Theta}_d$, generated from a Dirichlet distribution with prior $\vec{\alpha}$. The words in the document are then generated by drawing a topic $z_i = t$ from this distribution and then drawing a word w_i from that topic according to a multinomial distribution with parameters $\vec{\Phi}_t$ generated from a Dirichlet distribution with prior $\vec{\beta}$ [16].

This procedure is a joint probability distribution over the random variables $(\mathcal{D}, \vec{z}, \underline{\Phi}, \underline{\Theta})$ given by [3]

$$p(\mathcal{D}, \vec{z}, \underline{\Phi}, \underline{\Theta}|\vec{\alpha}, \vec{\beta}) \propto \left(\prod_{t}^{K} p(\vec{\Phi}_t|\vec{\beta})\right)\left(\prod_{d}^{D} p(\vec{\Theta}_d|\vec{\alpha})\right)\left(\prod_{i}^{N} \Phi_{z_i,w_i}\Theta_{d_i,z_i}\right), \tag{2.4}$$

where $\underline{\Phi} = \{\vec{\Phi}_t\}_{t=1}^{K}$, $\underline{\Theta} = \{\vec{\Theta}_d\}_{d=1}^{D}$, Φ_{z_i,w_i} is the w_ith element in vector $\vec{\Phi}_{z_i}$, Θ_{d_i,z_i} is the z_ith element in the vector $\vec{\Theta}_{d_i}$, and d_i associates each word with a document index $d_i \in \{1, 2, \cdots, D\}$.

The LDA graphical model, and the conditional dependencies implied from the distributions are represented in Figure 2.1.

Note that words are the only observed variables. The hyperparameters $\vec{\alpha}$ and $\vec{\beta}$ are input from the user. The latent topic assignments $\vec{z}$, document distributions over topics $\underline{\Theta}$, and topic distributions over words $\underline{\Phi}$ are all unobserved. Estimation of $\underline{\Theta}$ and $\underline{\Phi}$ requires computing the latent topic assignments $\vec{z}$, $p(\vec{z}|\mathcal{D}, \vec{\alpha}, \vec{\beta})$. Unfortunately, this posterior distribution is intractable due to the coupling between $\underline{\Phi}$ and $\underline{\Theta}$ [16].

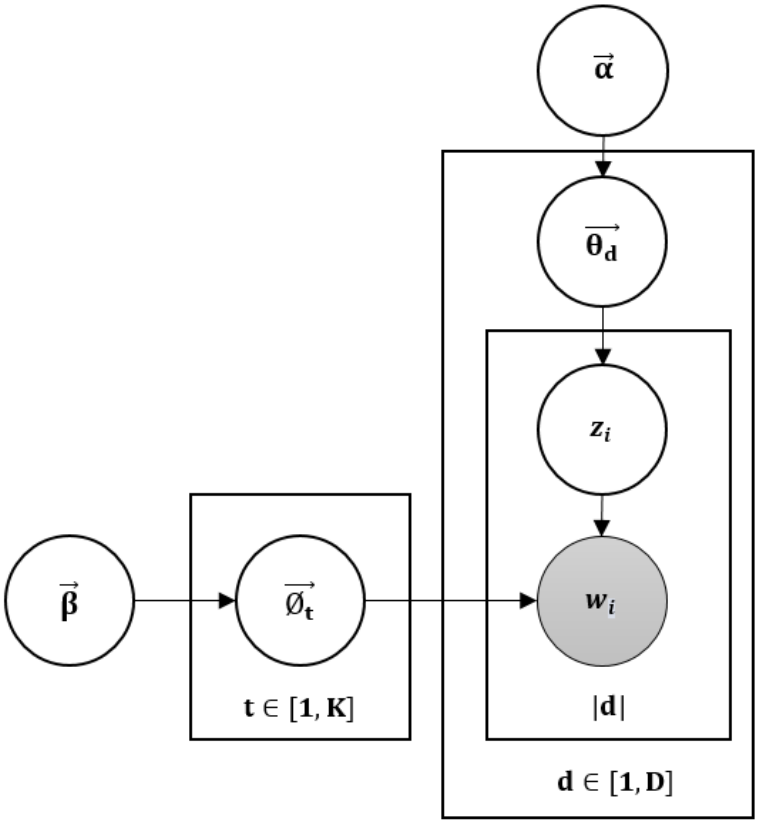

Figure 2.1: The graphical model representation of LDA. D represents the total number of documents, and $|d|$ is the length of a document d. The directed edges indicate conditional dependencies. For example, each word w depends on both the latent topic z and the topic-word multinomial $\vec{\Phi}_t$, drawn from Dirichlet($\vec{\beta}$) [16].

However, various approximate inference algorithms can be used to infer the posterior distribution. Some of these approximate inference algorithms are Laplace approximation, Variational approximation [16], Expectation-propagation [62], and Gibbs sampling [40]. These algorithms can differ in speed and accuracy. Asuncion *et al.* [5] show that these inference algorithms have relatively similar predictive performance when the hyperparameters for each method are selected in an optimal fashion. Thus, the results are significantly affected by hyperparameter settings. These hyperparameter settings can be learned from data [3, 16, 80]. However, others show that learning hyperparameters from data can have strong impact on the learned topics [82]. In our work, we focus on Gibbs sampling. Gibbs sampling is competitive in speed with other existing algorithms. However, a significant advantage of Gibbs sampling is ease of implementation in software. The following section describes Gibbs sampling and how it is used with LDA.

2.4.1 Inference via Gibbs sampling

Griffiths *et al.* [39, 40] proposed to use Gibbs sampling to obtain approximate estimates for the latent variables as well as the posterior distributions. Gibbs sampling is a special case of Markov chain Monte Carlo (MCMC) algorithm. An MCMC algorithm emulates high-dimensional probability distributions $p(\vec{z})$ by the stationary behaviour of a Markov chain. This means that one sample is generated for each transition in the chain after a stationary state of the chain has been reached, which happens after a *burn-in period* [43]. Gibbs sampling is a simple Markov chain Monte Carlo (MCMC) algorithm where the dimensions z_i of the distribution are sampled

alternatively one at a time, conditioned on the values of all other dimensions, denoted by $\vec{z}_{-i}$ [10, 43].

For example, consider the distribution $p(\vec{z}) = p(z_1, \cdots, z_N)$ from which we wish to sample. At first, we initialize each $z_i \in \vec{z}$. Each step of Gibbs sampling involves replacing the value of one of the variables, by the value drawn from the distribution of that variable conditioned on the values of the remaining variables [10]. The procedure of Gibbs sampling is summarized below:

1. Randomly initialize each $z_i \in \vec{z}$
2. For $i = 1, \cdots, N$
   (a) $z_1^{i+1} \sim p(z_1 | z_2^{(i)}, z_3^{(i)}, \cdots, z_N^{(i)})$
   (b) $z_2^{i+1} \sim p(z_2 | z_1^{(i+1)}, z_3^{(i)}, \cdots, z_N^{(i)})$
   $\cdots$
   (c) $z_N^{i+1} \sim p(z_N | z_1^{(i+1)}, z_2^{(i+1)}, \cdots, z_{N-1}^{(i+1)})$

To build a Gibbs sampler, the full conditionals $p(z_i | \vec{z}_{-i})$ is found using:

$$p(z_i | \vec{z}_{-i}) = \frac{p(\vec{z})}{p(\vec{z}_{-i})}. \tag{2.5}$$

2.4.2 The collapsed LDA Gibbs sampler

Griffiths *et al.* derive a Gibbs sampler for LDA by applying the hidden variable method from above [39, 43]. It is assumed that each document d is a multinomial distribution over K topics with parameters $\vec{\Theta}_d$. Thus, for a word in document d, $p(z_i = t|d) = \Theta_{d,t}$, where z_i is the hidden variable, denoting topic assignment to word

i, and $\vec{\Theta}_d$ is generated from a Dirichlet distribution with prior $\vec{\alpha}$. The tth topic is a multinomial distribution over V words with parameter $\vec{\Phi}_t$, generated from a Dirichlet distribution with prior $\vec{\beta}$, thus $p(w_i|z_i = t) = \Phi_{t,w_i}$ [39, 40].

In this method, the parameter sets $\underline{\Theta}$ and $\underline{\Phi}$ can be integrated out because they can be interpreted as statistics of the associations between the observed w_i and the corresponding z_i [39, 43]. The strategy of integrating out $\underline{\Theta}$ and $\underline{\Phi}$ is referred to as *collapsed* approach often used in Gibbs sampling [43].

For each topic t the distribution is given by

$$p(z_i = t|\vec{z}_{-i}, \mathcal{D}) \propto p(w_i|z_i = t, \vec{z}_{-i}, \mathcal{D}_{-i})p(z_i = t|\vec{z}_{-i}), \tag{2.6}$$

where $\vec{z}_{-i}$ and $\mathcal{D}_{-i}$ denote the $\vec{z}$ and $\mathcal{D}$ for all words other than w_i. This expression is an instance of Bayes' rule with $p(w_i|z_i = t, \vec{z}_{-i}, \mathcal{D}_{-i})$ as the likelihood of the data given a particular choice of z_i and $p(z_i = t|\vec{z}_{-i})$ as the prior on z_i. The likelihood is obtained by integrating over the parameters $\underline{\Phi}$, which results in

$$p(w_i|z_i = t, \vec{z}_{-i}, \mathcal{D}_{-i}) = \frac{n_{-i,t}^{(w_i)} + \beta}{n_{-i,t}^{(.)} + V\beta}, \tag{2.7}$$

where $n_{-i,t}^{(.)}$ is the total number of words assigned to topic t, excluding the current one, and $n_{-i,t}^{(w_i)}$ is the total number of times word w_i is assigned to topic t, excluding the current one.

Similarly, the prior is calculated by integrating over the parameter $\underline{\Theta}$:

$$p(z_i = t|\vec{z}_{-i}) = \frac{n_{-i,t}^{(d)} + \alpha}{n_{-i,\cdot}^{(d)} + K\alpha}, \tag{2.8}$$

where $n_{-i,t}^{(d)}$ is the total number of words from document d assigned to topic t, excluding

the current one, and $n_{-i,\cdot}^{(d)}$ is the total number of words in document d, excluding the current one. Then, the conditional distribution for the topic assignments is given by

$$p(z_i = t|\vec{z}_{-i}, \mathcal{D}) \propto \frac{n_{-i,t}^{(w_i)} + \beta}{n_{-i,t}^{(\cdot)} + V\beta} \cdot \frac{n_{-i,t}^{(d)} + \alpha}{n_{-i,\cdot}^{(d)} + K\alpha}. \tag{2.9}$$

The Markov Chain Monte Carlo (MCMC) algorithm is then straightforward. The z_i's are initialized between 1 and K, determining the initial state of the Markov chain. The chain is then run for a number of iterations, each time finding a new state by sampling each z_i from the distribution specified by Equation 2.9. After sufficient iterations (i.e., burn-in period) for the chain to approach the target distribution, the current values of the z_i's are recorded. However, the required length of the burn-in is one of the drawbacks with MCMC approaches. In order to check that the Markov chain has converged, experimental studies with different number of iterations are conducted. The results that lead to a fine-grained decomposition of the corpus into topics, and topics into words are selected [43][2].

With a set of samples from the posterior distributions $\underline{\Phi}$ and $\underline{\Theta}$ can be computed by integrating across the full set of samples. For any single sample we can estimate $\Theta_{d,t}$ by

$$\Theta_{d,t} = \frac{n_t^{(d)} + \alpha}{n_{\cdot}^{(d)} + K\alpha}, \tag{2.10}$$

where $n_t^{(d)}$ is the total number of words from document d assigned to topic t and $n_{\cdot}^{(d)}$ is the total number of words in document d.

Similarly, Φ_{t,w_i} is estimated by

[2]Subsequent samples are taken after an appropriate lag to ensure that their autocorrelation is low [40]

$$\Phi_{t,w_i} = \frac{n_t^{(w_i)} + \beta}{n_t^{(.)} + V\beta}, \tag{2.11}$$

where $n_t^{(w_i)}$ is the total number of times word w_i is assigned to topic t and $n_t^{(.)}$ is the total number of words assigned to topic t.

2.4.3 Estimation

The LDA trained generative models are typically used to estimate the probability of unseen test data $\mathcal{D}_{test}$, given training data $\mathcal{D}_{train}$ and hyperparameters $\vec{\alpha}$ and $\vec{\beta}$. This ability to estimate the probability of unseen data is the major difference between LDA and PLSI, mentioned in Section 2.3. Let $\mathcal{D}_{test} = \{w_1, w_2, \cdots, w_M\}$ represent a test corpus of length M. The probability $p(\mathcal{D}_{test}|\mathcal{D}_{train}, \vec{z}_{train}, \vec{\alpha}, \vec{\beta})$ for the test corpus is computed by normalizing the constant that relates the posterior distribution over $\vec{z}_{train}$ to the joint distribution over $\mathcal{D}_{test}$ and $\vec{z}_{test}$ in Bayes' rule [81]. There are many existing methods for estimating normalizing constants [83]. In this dissertaion, we use the *left-to-right* algorithm for estimating normalizing constants by sequentially approximating the marginalisation over latent topics [81, 83]. This method operates in an incremental, left-to-right fashion, where topic assignments from positions $n^{'} > n$ cannot influence the assignment at position n and words from positions $n^{'} > n$ cannot influence the probability of the word at position n.

The left-to-right algorithm decomposes $p(\mathcal{D}_{test}|\mathcal{D}_{train}, \vec{z}_{train}, \vec{\alpha}, \vec{\beta})$ as:

$$p(\mathcal{D}_{test}|\mathcal{D}_{train},\vec{z}_{train},\vec{\alpha},\vec{\beta}) = \prod_n p(w_n|\mathcal{D}_{test_{<n}},\mathcal{D}_{train},\vec{z}_{train},\vec{\alpha},\vec{\beta}) \tag{2.12}$$

$$= \prod_n \sum_{\vec{z}_{test_{\leq n}}} p(w_n,\vec{z}_{test_{\leq n}}|\mathcal{D}_{test_{<n}},\mathcal{D}_{train},\vec{z}_{train},\vec{\alpha},\vec{\beta}),$$

and uses algorithm 1 to approximate the sums over $\vec{z}_{test_{\leq n}}$.

Algorithm 1 A left-to-right estimation algorithm for topic models. The algorithm computes $l \simeq \sum_n \log \sum_{\vec{z}_{test_{\leq n}}} p(w_n, z_{test_{\leq n}}|\mathcal{D}_{test_{<n}},\mathcal{D}_{train},\vec{z}_{train},\vec{\alpha},\vec{\beta})$ using R particles [81].

initialize $l := 0$

for each position $n \in \mathcal{D}_{test}$ **do**

  $p_n = 0$

  for each particle $r = 1$ to R **do**

    for $(n^{'} < n)$ **do**

      resample $z_{n'} \sim p(z_{n'}|(\vec{z}_{test_{<n}})_{-n'},\mathcal{D}_{test_{<n}},\mathcal{D}_{train},\vec{z}_{train})$

    $p_n := p_n + \sum_t p(w_n|z_n = t,\vec{z}_{test_{<n}},\mathcal{D}_{test_{<n}},\mathcal{D}_{train},\vec{z}_{train})p(z_n = t|\vec{z}_{test_{<n}},\vec{z}_{train})$

  $p_n := p_n/R$

  $l := l + \log p_n$

  sample $z_n \sim p(z_n|\vec{z}_{test_{<n}},\mathcal{D}_{test_{\leq n}},\mathcal{D}_{train},\vec{z}_{train})$

return l

Chapter 3

Related Work

Text documents are the only observed data in most conventional topic models. However, more recent topic models extend previous models by incorporating extra information [3]. Extra information is obtained by enriching text representation to include information, such as authors of the documents [74], images associated with the text [13], style of writing and reviewers of the documents [61], and discriminative frequent patterns of the documents [38]. The aforementioned topic models represent documents as a bag-of-words, where the order of words, thus important linguistic structures of documents are neglected [16, 40].

In order to include richer linguistic structures of text documents, many methods were proposed to incorporate local word dependencies into topic models [20, 41, 42, 80, 85]. The following sections discuss current extensions to LDA.

3.1 Syntactic LDA

Topic models represent documents as a bag-of-words, where the order of words, thus, important linguistic structures of documents are neglected [16, 40]. To remedy this problem, some recent methods integrate grammatical regularities of text documents into topic models.

HMM-LDA [41] uses the states of a Hidden Markov Model to represent syntactic and semantic words. The model assumes that words are either sampled from topics randomly drawn from the topic mixture of the documents or from a syntactic class sampled from a distribution of associated syntactic classes [42]. Their model only considers local dependencies between variables of the syntactic states and fails to obtain syntactic or semantic dependencies between words.

The Syntactic Topic Model (STM) [20] was proposed to integrate grammatical regularities in the text to detect syntactically relevant topics. In STM, documents are collections of dependency parse trees, in which words in the sentence are the nodes in the graph and grammatical regularities are the edge labels [29]. The root in the dependency parse tree is used as a governor. Topic assignment of the root node affects topic assignments of all its children. Moreover, STM does not draw words from just the document distribution over topics. Rather, it draws a word from a distribution formed by the document distribution over topics weighted by the parse tree distributions. Thus, topic assignment of a word depends on both the document's theme as well as the parents of the word in the parse tree. Although, STM improves topic modeling by combining syntactic and thematic structures of documents, it does not fully distinguish topic assignment of the words that share the same parent in the

tree, i.e., children of a node. This problem specifically occurs when a root node has many children [20].

3.2 Semantic LDA

Wallach [80] proposed a probabilistic language model by incorporating both n-gram statistics and latent topic variables. They extend word generation by conditioning on n previous words. However, the n-gram topic models do not capture relations between nonconsecutive words.

Chemudugunta *et al.* [25] proposed an approach by combining semantic concepts, defined by a subset of words in the documents, and statistical learning techniques. Similar to the basic LDA, they assume that documents are distributed over topics. However, they add another assumption that documents are also distributions over concepts; and each concept is a distribution over words. A concept is restricted to assign non-zero probabilities only to words under the concept; and zero probabilities to words outside the concept. The Concept-Topic Model has the advantage of linking known concepts to the data. The main disadvantage of the model is that the provided concepts are limited to a specific domain and cannot be generalized.

Musat *et al.* [64] use ontological trees derived from Wordnet [60] to remove outliers from topic labels, resultant from a topic modeling procedure. They align the distribution of topics over words to the conceptual tree; and prune the words that are not interrelated to other words from a conceptual perspective. Their method results in a more conceptually cohesive topic words that improves topic readability. However, their method is a post-processing step for improving topic modeling results. Applying

conceptual data in earlier steps of topic modeling may result in better results.

In order to incorporate the underlying significance of terms into topic models, many methods were proposed. TagLDA [93] includes document structure knowledge into topic models in the form of tags on terms. Each term in documents is tagged based on its part-of-speech or its location in the document. Although TagLDA improves LDA in terms of test set perplexity, it does not consider term importance in topics. Also, the method describes knowledge on individual terms as opposed to a collection of terms obtained from n-grams.

Recently, some work has been done to inject domain knowledge into topic models to enforce probabilistically correlated terms to be in same topic and remove outliers [2, 64, 70]. However, none of these methods utilize term importance in topic models.

Wilson *et al.* [87] proposed a term weighting scheme for topic models. They weigh terms by measuring the information content of the terms, and compute probabilities based on the weighted counts. Although their approach has improved cross-language retrieval tasks by eliminating frequent terms from topics, their approach does not differentiate between actual low-content frequent terms and terms that occur frequently but are very important with high semantic content [27]. Moreover, they do not consider significance of terms to the theme of the corpus with respect to an external data source (i.e. Wikipedia). In addition, they use a symmetric Dirichlet prior over document-topic and topic-term distributions in the estimation process.

3.3 Information about documents

This section discusses methods proposed to incorporate information about documents in topic models. Blei *et al.* [15] proposed supervised topic models to improve inference of latent topics. They paired each document with a label. Then, they jointly model the documents and the responses to find latent topics that best predict the label. This method particularly influences the topic assignments by the model.

Blei *et al.* [13] claim that similar annotated images of texts should share similar caption words and thus similar text topics. They proposed Correspondence LDA that is a joint model of images and their corresponding text, i.e., captions. They segmented each image into regions based on their visual features, i.e., size. They assume that each document is a multinomial distribution over topics, and each topic is a multivariate Gaussian distribution over image regions to generate images; and a multinomial distribution over caption words to generate the caption words. As a result of this model, similar images will contain similar text topics. Also, This model has many applications in vision tasks, such as automatic annotation of new images [84, 86].

Markov Random Topic Fields, proposed by Daumé [28], represents a corpus by a weighted graph. Documents are the nodes of the graph, and are connected via edges, weighed based on the similarities of connected documents. Their model results in similar documents to contain similar topics.

Rosen-zvi *et al.* [74] extend LDA by considering interests of authors of the documents. They assume that each author is a multinomial distribution over topics, and each topic is a multinomial distribution over words. Each document is generated by

first drawing a distribution over authors. The words in the document are then generated by drawing a topic from this distribution and then drawing a word from that topic. Their model significantly improves topic models. However, it ignores several aspects of real world document generation, i.e., word ordering.

McCallum *et al.* [56] proposed an extension of the author topic model [74], where topics are conditioned on both the sender as well as the receiver of the documents. This extension does not require changes to the generative model of the author topic model [74], and improves its results.

3.4 Correlated topic models

Correlated topic models modify the topic modeling procedure to capture dependencies between topics. For example, a document about genetics is more likely to also be about disease than x-ray astronomy [14]. Blei *et al.* [14] proposed Correlated Topic Models (CTM) by relaxing the strong independence assumption between topics detected by LDA [16]. They achieve this correlation by assuming documents to be a logistic normal distribution over topics, allowing pairwise correlations between topics. However, topics may correlate hierarchically. Blei *et al.* proposed Hierarchical LDA (hLDA) [12] to capture hierarchical dependencies between topics. hLDA models topics with a tree-structured hierarchy over topics where topics get more specific as one moves from the root to the leaf. A document is generated by first choosing a path from the root of the tree to the leaf. Then, a vector of topic proportions is drawn from a Dirichlet distribution. After that the words are generated from a mixture of the topics along the path from the root to the leaf. However, in hierarchical LDA, doc-

uments are represented by a bag-of-words where syntax and semantics relationships of the words in documents are neglected.

In this dissertation, in order to advance the state-of-the-art, we go beyond the bag-of-words representation of text documents to incorporate syntax and semantics of text documents into topic models. We enrich text documents with syntactic and semantic dependencies between consecutive and nonconsecutive words. In addition, we use WordNet to enforce coherent topic assignments for conceptually similar words by generalizing words with their synonyms. Moreover, we use an external knowledge (Wikipedia) to obtain importance weights of the terms of documents. We further incorporate these term importance weights into latent topic variables by boosting the probability of important terms and consequently decreasing the probability of less important terms to better reflect the themes of documents.

Chapter 4

Topic Modeling using Collapsed Typed Dependency Relations

Topic modeling is a powerful tool to uncover hidden thematic structures of documents. Many conventional topic models represent documents as a bag-of-words, where the important linguistic structures of documents are neglected. In this chapter, we propose a novel topic model [32] that enriches text documents with collapsed typed dependency relations to effectively acquire syntactic and semantic dependencies between consecutive and nonconsecutive words. In addition, we propose to enforce coherent topic assignments for conceptually similar words by generalizing words with their synonyms. Our experimental studies show that the proposed model and strategy outperform the original LDA model and the Bigram topic model in terms of perplexity; and our performance is comparable to other models in terms of stability, coherence, and accuracy.

4.1 Introduction

Text documents are the only observed data in most conventional topic models. Therefore, the order of words, and thus important linguistic structures of documents, i.e. local word dependencies in a document, are typically neglected [16, 40]. Local word dependencies are either dependencies between a set of consecutive words, or a set of nonconsecutive words with arbitrary distances. For example, the term[1] *"data mining"* contains two words *"data"* and *"mining"* that are consecutively related. In addition, in sentence *"Some countries deny human basic civil rights."*, the term *"human rights"* contains two nonconsecutive words *"human"* and *"rights"* that are syntactically related. In order to incorporate sequential consecutive dependencies between words into topic models, the Bigram topic model [80] and Topical n-gram Model [85] extend word generation by conditioning not only on the topic of the word, but also on n previous words. However, the n-gram topic models only capture relations between consecutive words, ignoring the relations between nonconsecutive words.

Moreover, text documents consist of words with possible conceptual similarities, called *synonyms*, defined in lexical resources like WordNet [60]. It is reasonable to expect the distribution of topics over synonymous words to be similar.

In this chapter, a novel topic model is proposed to consider syntactic and semantic structures of text documents in probabilistic topic models. In essence, we enrich text documents with the *collapsed typed dependency relations* to circumvent obstacles in acquiring consecutive and nonconsecutive dependencies between words. In addition, we investigate the influence of enforcing similar topic distribution over conceptually

[1] A *term* consists of one or more words forming a unit of a sentence.

similar words by generalizing words with their synonyms.

The structure of this chapter is as follows: In Section 4.2, we discuss collapsed typed dependency relations and HPSG parse trees. In Section 4.3, we explain our proposed topic model incorporated with collapsed typed dependency relations. In Section 4.4, we explain our method for generalizing words using synonyms. We discuss the relationship between our topic models and other similar counterparts in Section 4.5. Section 4.6 introduces some criteria to evaluate topic models. Then, it demonstrates the effectiveness of our approach through experiments. Finally, Section 4.7 summarises the chapter.

4.2 Collapsed typed dependency relations and HPSG parse trees

The bag-of-words representation of text documents is of particular interest in most topic models. However, this representation does not contain information about the relations between words. Relations could hold over a consecutive or nonconsecutive neighborhood of a word [50].

In this work, we use the collapsed typed dependency relations to acquire syntactic and semantic structures of text documents. This acquisition enables us to further capture consecutive and nonconsecutive relations between words of text documents. Typed dependency relations are extracted from typed dependency parse trees that are respectively constructed according to the *Head-Driven Phrase Structure Grammar* (HPSG) that is explained in the following section.

4.2.1 The HPSG-based parse trees

The Head-Driven Phrase Structure Grammar (HPSG), developed by Pollard *et al.* [71], is a highly structured grammatical representation of text documents. The reason we choose the HPSG-based grammars is the high degree of its formal explicitness that effectively analyzes syntactic relations concerning multi-word constituents [30, 50]. The HPSG-based parse tree of a sentence starts from a root and ends in leaf nodes which represent words. Internal nodes of the tree represent syntactic roles of the connected leaf nodes. For example, Figure 4.1 represents the HPSG-based parse tree of the sentence *"Some countries deny human basic civil rights."*[2] In this tree, the left-most branch, node *NP* represents the role of "noun phrase" for the leaf node *"Some countries"*.

[2]Enju is used to extract the HPSG parse tree. This parser is available at http://www.nactem.ac.uk/enju.

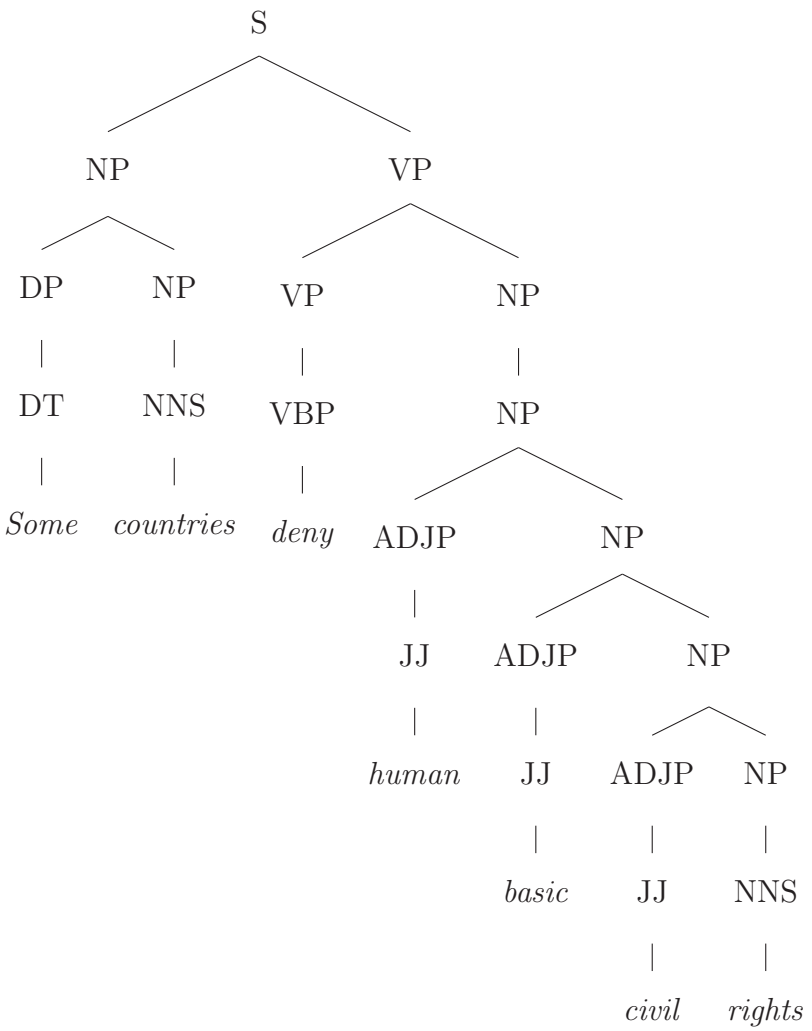

Figure 4.1: The HPSG-based parse tree of the sentence *"Some countries deny human basic civil rights."* Abbreviations that are used in this tree are as follows: *S*: sentence; *NP*: noun phrase; *VP*: verb phrase; *DP*: determiner phrase; *DT*: determiner; *NNS*: plural noun; *ADJP*: adjective phrase; *JJ*: adjective.

Grammatical Relation	**Definition**	**Example**
root	It points to the root of the sentence; and acts as the root of the tree.	"I love French fries." root(root, love)
amod(w_i,w_j)	Adjective Modifier: w_j is an adjective that changes the meaning of w_i.	"Sam eats red meat." amod(meat, red)
rcmod(w_i,w_j)	Relative Clause Modifier: w_j ia a verb in a relative clause that changes the meaning of w_i.	"I saw the man you love." rcmod(man, love)
nsubj(w_i,w_j)	Nominal Subject: w_j is a subject of a verb w_i.	"Clinton defeated Dole." nsubj (defeated, Clinton)
dobj(w_i,w_j)	Direct Object: w_j is a direct object of a verb w_i.	"They win the lottery." dobj (win, lottery)
det(w_i,w_j)	Determiner: w_j is a determiner of the head of a noun phrase w_i.	"The man is here." det (man, The)

Table 4.1: Most common grammatical relations used in typed dependency parse trees, defined in de Marneffe *et al.* [29, 30]. A comprehensive set of all grammatical relations used in this dissertation is explained in Appendix A.

The HPSG-based parse trees provide a high level syntactic representation of sentences in text documents [30]. However, we need to capture specific relations between every individual related pair of words. Thus, we need to elaborate HPSG to include additional labelled grammatical relations between words. This is achieved by constructing the following collapsed typed dependency parse trees from the HPSG-based parse trees.

4.2.2 Collapsed typed dependency relations

While the HPSG-based parse trees represent nesting of multi-word constituents, a dependency parse tree represents dependencies between individual words. A *typed dependency parse tree* of a sentence provides a tree representation of detailed grammatical relations between words in the sentence [30]. The algorithm to extract typed dependency parse trees from the HPSG parse trees has two phases [30]: dependency extraction and dependency typing. In the first phase, a sentence is parsed with a phrase structure grammar parser (HPSG), explained in Section 4.2.1. The output of this phase is arranged hierarchically and rooted with the most generic relation. In the second phase, when the relation between an internal node and its connected leaf node can be identified more precisely, more specific grammatical relations further down in the hierarchy is used. For example, Figure 4.2 shows the typed dependency parse tree for the sentence *"Some countries deny human basic civil rights."* This tree is constructed from the HPSG-based parse tree, shown in Figure 4.1. A more clear view of this tree is also shown horizontally in Figure 4.3, where the order of the words is present. Words in the sentence are nodes of the tree and grammatical relations are

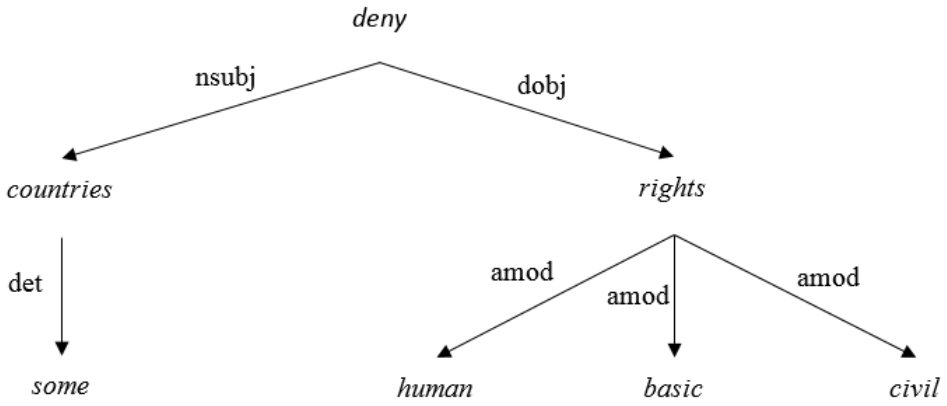

Figure 4.2: The typed dependency parse tree of the sentence *"Some countries deny human basic civil rights."* See Table 4.1 for the explanation of each relation.

the edge labels. For example, *dobj* edge label between two nodes *edge* and *label* represents that the word *rights* is the direct object of the verb *deny*. Each grammatical relation is an instance of the 48 grammatical relations mentioned in [30]. Table 4.1 shows most common grammatical relations used in typed dependency parse trees. A comprehensive set of all grammatical relations used in this dissertation is explained in Appendix A.

As illustrated in Figure 4.3, nonconsecutive relations between words with gaps, i.e. *"human rights"*, is captured under the *amod* relation. Using bag-of-words or n-gram methods to represent text documents fails to capture these relations between nonconsecutive words.

For each edge in the tree, we extract a relation $rel(w_i, w_j)$, where rel is the edge label representing a relation and w_i and w_j are two nodes of the edge. For example, the set of relations extracted from the typed dependency parse tree, illustrated in Figure 4.3, is as follows: {*root*(root, deny), *det*(countries, Some), *nsubj*(deny, countries),

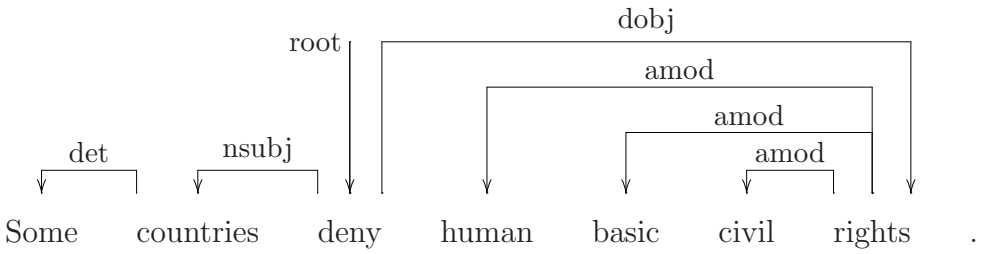

Figure 4.3: The horizontal presentation of the typed dependency parse tree of the sentence *"Some countries deny human basic civil rights."* As illustrated in this figure, the typed dependency parse tree effectively captures relations between nonconsecutive words, i.e., *amod* relation between words *human* and *rights*.

amod(rights, human), *amod*(rights, basic), *amod*(rights, civil), dobj(deny, rights)}.

The relations from typed dependency parse trees are further processed by collapsing relations involving prepositions and conjuncts to get direct dependencies between content words [30]. This collapsing is often useful in simplifying and filtering the relations. For instance, the sentence *"A company is based in LA."* contains the following relations: *prep*(based, in) and *pobj*(in, LA), where *prep* represents that *"in"* is a prepositional modifier of a verb *"based"*, and *pobj* represents that *"LA"* is the object of a the preposition *"in"*. The dependencies involving the preposition *"in"* in the aforementioned example will be collapsed into one single relation: *prep-in*(based, LA). The collapsed typed dependency parse trees are constructed using the Stanford parser toolkit that has phrase structured grammars integrated in [29, 30][3].

As a result, collapsed typed dependency relations not only capture relations between consecutive and nonconsecutive words, but they also eliminate less informative relations involving prepositions.

[3]http://nlp.stanford.edu/software/lex-parser.shtml

In our work, we use the collapsed typed dependency relations to represent the corpus. These relations enable us to better distinguish topic assignments for the relations involving the same parent. For instance, a tree including a parent with c children, will be represented by c relations, where each relation denotes the edge connecting the child and the parent. We further propose the following topic model to consider the collapsed typed dependency relations and assign a discriminate topic to each relation.

4.3 The HPSG-based topic model

In this section, we propose *the HPSG-based topic model* that enriches text documents with collapsed typed dependency relations to effectively acquire syntactic and semantic dependencies between consecutive and nonconsecutive words.

We assume that corpus $\mathcal{R}$ consists of M documents. We also assume that $\vec{R} = \{r_1, r_2, \cdots, r_R\}$ represents a corpus of R unique collapsed dependency relations between words. This set of unique vocabulary relations can be the set of unique collapsed typed dependency relations extracted from the corpus. These relations are instances of the 48 grammatical relations described in Section 4.2.2, each of which consists of two words. In addition, we assume that each relation $r_i \in \vec{R}$ is associated with a latent topic variable z_i where $i \in \{1, 2, \cdots, N\}$.

Our topic model assumes that each document d has a multinomial distribution over K topics with parameters $\vec{\Theta}_d$. Thus, for a relation r_i in document d, $p(z_{r_i} = k|d) = \Theta_{d,k}$. In our proposed model, the kth topic is represented by a multinomial distribution over R relations with parameters $\vec{\Phi}_k$, thus $p(r_i|z_{r_i} = k) = \Phi_{k,r_i}$.

Inspired from LDA [16, 39, 40], we provide a procedure to generate documents. In this procedure, each document d is generated by first drawing a distribution over topics ($\vec{\Theta}_d$), generated from a Dirichlet distribution with prior $\vec{\alpha}$. The relations in the document are then generated by drawing a topic k from this distribution and then drawing a relation from that topic according to a multinomial distribution over relations ($\vec{\Phi}_k$), generated from a Dirichlet distribution with prior $\vec{\beta}$.

Note that the only observed variables are the relations in the collection of relations $\vec{R}$. Document distribution over topics and topic distribution over relations are latent variables generated from Dirichlet distributions with priors $\vec{\alpha}$ and $\vec{\beta}$, respectively. We use Gibbs sampling to obtain approximate estimates for the latent variables. Gibbs sampling is a simple Markov chain Monte Carlo algorithm that sequentially replaces the value of one of the latent variables by a value drawn from the distribution of that variable conditioned on the values of the remaining variables [10].

We adopt Gibbs sampling algorithm proposed by Griffiths *et al.* [39, 40] to draw a topic from the conditional distribution iteratively. The complete likelihood of the model is factored as: $p(\vec{R}, \vec{z}|\vec{\alpha}, \vec{\beta}) = p(\vec{R}|\vec{z}, \vec{\beta})p(\vec{z}|\vec{\alpha})$. The first probability is an average over $\underline{\Phi}$, where $\underline{\Phi} = \{\vec{\Phi}_k\}_{k=1}^{K}$:

$$p(\vec{R}|\vec{z}, \vec{\beta}) = \int_{\underline{\Phi}} p(\vec{R}|\vec{z}, \underline{\Phi})p(\underline{\Phi}|\vec{\beta})d\underline{\Phi}. \tag{4.1}$$

The first term in Equation 4.1 is obtained as:

$$p(\vec{R}|\vec{z}, \underline{\Phi}) = \prod_{k=1}^{K}\prod_{i=1}^{R}(\Phi_{k,r_i})^{n_k^{r_i}}, \tag{4.2}$$

where $n_k^{r_i}$ is the total number of times topic k is assigned to relation r_i. By expanding $p(\underline{\Phi}|\vec{\beta})$ as a Dirichlet distribution, we obtain:

$$p(\underline{\Phi}|\vec{\beta}) = \prod_{k=1}^{K} \frac{1}{B(\vec{\beta})} \prod_{i=1}^{R} (\Phi_{k,r_i})^{\beta-1}, \tag{4.3}$$

where $B(.)$ is the Beta function. Substituting the first and the second terms of Equation 4.1 with Equations 4.2 and 4.3, and using the Dirichlet integral[4] yields:

$$\begin{aligned} p(\vec{R}|\vec{z},\vec{\beta}) &= \int_{\vec{\Phi}_k} \prod_{k=1}^{K} \frac{1}{B(\vec{\beta})} \prod_{i=1}^{R} (\Phi_{k,r_i})^{n_k^{r_i}+\beta-1} d\vec{\Phi}_k \\ &= \prod_{k=1}^{K} \frac{1}{B(\vec{\beta})} \int_{\vec{\Phi}_k} \prod_{i=1}^{R} (\Phi_{k,r_i})^{n_k^{r_i}+\beta-1} d\vec{\Phi}_k \\ &= \prod_{k=1}^{K} \frac{B(\vec{n}_k + \vec{\beta})}{B(\vec{\beta})}, \end{aligned} \tag{4.4}$$

where $\vec{n}_k = \{n_k^{r_i}\}_{i=1}^{R}$.

$p(\vec{z}|\vec{\alpha})$ remains analogous to LDA and is obtained by:

$$p(\vec{z}|\vec{\alpha}) = \prod_{d=1}^{M} \frac{B(\vec{n}_d + \vec{\alpha})}{B(\vec{\alpha})}, \tag{4.5}$$

where $\vec{n}_d = \{n_d^k\}_{k=1}^{K}$.

We can derive the full conditional distribution for relation r_i in document d generated by topic $z_{r_i} = k$:

[4] $B(\vec{\gamma}) = \int_{\vec{x}} \prod_{i=1}^{N} (x_i)^{\gamma_i - 1} d_{\vec{x}}^N$

$$p(z_{r_i} = k|\vec{z}_{-r_i}, \vec{R}) = \frac{p(\vec{R}, \vec{z})}{p(\vec{R}, \vec{z}_{-r_i})} = \frac{p(\vec{R}|\vec{z})}{p(\vec{R}_{-r_i}|\vec{z}_{-r_i})p(r_i)} \cdot \frac{p(\vec{z})}{p(\vec{z}_{-r_i})} \tag{4.6}$$
$$= \frac{B(\vec{n}_k + \vec{\beta})}{B(\vec{n}_{k,-r_i} + \vec{\beta})} \cdot \frac{B(\vec{n}_d + \vec{\alpha})}{B(\vec{n}_{d,-d} + \vec{\alpha})}$$
$$= \frac{n^{r_i}_{k,-r_i} + \beta}{\sum_{i=1}^{R}(n^{r_i}_{k,-r_i} + \beta)} \cdot \frac{n^{k}_{d,-d} + \alpha}{\sum_{k=1}^{K}(n^{k}_{d,-d} + \alpha)},$$

where $n^{r_i}_{k,-r_i}$ is the total number of times topic k is assigned to relation r_i, excluding the current one, $n^{k}_{d,-d}$ is the total number of relations in document d assigned to topic k, excluding the current assignment.

Finally, we need to calculate the multinomial parameter sets $\underline{\Phi}$ and $\underline{\Theta}$. Note that $p(\vec{\Phi}_k|\vec{R}, \vec{\beta}) = Dirichlet(\vec{\Phi}_k|\vec{n}_k + \vec{\beta})$, and $p(\vec{\Theta}_d|\vec{R}, \vec{\alpha}) = Dirichlet(\vec{\Theta}|\vec{n}_d + \vec{\alpha})$. Using the expectation of the Dirichlet distribution ($Dirichlet(\beta) = \beta_i / \sum_i \beta_i$) yields:

$$\Phi_{k,r_i} = \frac{n^{r_i}_{k} + \beta}{n^{(.)}_{k} + R\beta}, \tag{4.7}$$

$$\Theta_{d,k} = \frac{n^{d}_{k} + \alpha}{n^{(.)}_{d} + K\alpha}, \tag{4.8}$$

where $n^{(.)}_{k}$ is the total number of relations assigned to topic k, n^{d}_{k} is the number of relations from document d assigned to topic k, and $n^{(.)}_{d}$ is the total number of relations in document d.

4.4 Generalizing words using synonyms

Text documents often contain words that are synonyms. Sets of synonyms can be obtained from lexical resources like WordNet [60]. In this work, we investigate the

influence of generalizing words using a synonym on topic modeling.

Similar to LDA [16], we assume that a document is a multinomial distribution over K topics, where each topic is a multinomial distribution over V vocabulary words. We also assume that documents are represented by a sequence of words, denoted by $\mathbf{W} = \{w_1, w_2, \cdots, w_N\}$, where $w_n \in \mathbf{W}$ is the nth word in the sequence. Given the fact that a set of synonyms shares a similar concept, it is reasonable to expect them to have similar probabilities under topics. For example, if a text document is about happiness, the inferred topic should assign higher probabilities to words such as *delighted*, *blessed*, and *prosperity*; and lower probabilities to words such as *sad*, *bitter*, and *sorrow*. In order to ensure that topics are similarly distributed over synonyms, we propose the following algorithm to replace all synonyms of a word with an equivalent synonym with the highest frequency in WordNet:

1. Group the words from WordNet, based on their conceptual similarities. Each group will contain a set of synonyms.

2. For each group, find the frequency of the words in the group. The frequency of a word is the number of occurrences of the word in WordNet.

3. Select the most frequent word in the group as the *group representative*.

4. For each $w_i \in \mathbf{W}$:

   Look for a group where w_i belongs to.

   If a group is found, replace w_i with the group representative, found in Step 3;

   else, leave the word as is.

For example, consider a text document that contains the word *prosperous*. This word belongs to the following group of synonyms {*delighted*, *blessed*, *prosperous*, *happy*, *fortunate*}. Our algorithm finds the frequency of each synonym in WordNet. It selects happy as the group representative because it is the most frequent word in the group. Finally, our algorithm replaces the word *prosperous* with the word *happy*.

4.5 Relationships to other work

In this work, we go beyond the bag-of-words representation of documents to incorporate syntax and semantics of text documents into topic models. This section reviews the theoretical relationships of our contributions with previous topic models that used syntactic and semantic structures of texts.

Our proposed topic model is similar to STM [20] due to using typed dependency trees to represent syntactic structures of sentences. However, our topic model has following major differences with STM. Firstly, STM draws a word from a single distribution formed by the document distribution over topics weighted by the parse tree distributions. Thus, topic assignment of a word depends on both the document's theme as well as the parent of the word in the parse tree. However, in our model we use two distributions: document distribution over topics and topic distribution over the collapsed dependency relations. We first draw a distribution over topics; then, we select a topic from this distribution and then draw a relation from that topic distribution over the collapsed dependency relations. Secondly, STM does not fully distinguish topic assignments of the words that share the same parent in the

dependency parse tree, i.e., children of a node, as stated by Boyd-Graber *et al.* [20]. However, in our model each pair of related nodes in the parse tree introduces a discriminate relation. Thus, topic assignment to the relations involving the same parent is better distinguished. Thirdly, STM does not use labelled dependency relations and lexicalization. However, our model uses the labels of dependency relations to distinguish and further collapse relations involving prepositions and conjuncts to get direct dependencies between content words. Finally, STM computes the posterior topic distributions by Bayesian variational methods. Our model uses Gibbs sampling to infer posterior topic distributions. This final difference is complementary rather than competitive.

In addition, our proposed topic model differs from the n-gram topic models [80] in capturing dependencies between words of a sentence. Our topic model considers dependencies between nonconsecutive words with a distance; while the n-gram topic model is limited to capturing dependencies between consecutive words.

Moreover, our proposed model, uses WordNet to enforce topic similarity for words with conceptual similarities, by generalizing similar words with their synonyms. Lexical resources, i.e. WordNet, were previously used in topic models. Musat *et al.* [63] employs WordNet to improve topic models by removing unrelated words from the simplified topic descriptions. Mei *et al.* [58] used WordNet to label each topic in a multinomial topic model. Newman *et al.* [66] uses WordNet to evaluate topic coherence. None of them uses synonyms to generalize words prior to building topic models.

4.6 Experiments

We conducted experiments on two text corpora to compare the performance of four following topic models: LDA [16], LDA on generalized words using synonyms, explained in Section 4.4, the Bigram Topic Model [80], and the HPSG-based topic model, explained in Section 4.3. The four topic models were trained with 1000 iterations of Gibbs sampling [39, 40] used in the MALLET [57]. Initial values for the hyperparameters (α, β) applied to all our experiments were $\alpha = 50.0$ and $\beta = 0.01$. Note that these parameters are default parameters of most LDA-based topic models, expected to result in a fine-grained decomposition of the corpus into topics [40].

In our experiments we used Associated Press corpus[5] that consists of $2,246$ Associated Press articles, $33,872$ words, and $454,370$ collapsed typed dependency relations. In addition, we used Reuters-21578 Distribution 1.0[6] that includes $10,789$ documents, $15,996$ words, and $793,345$ collapsed typed dependency relations. Note that all dependency relations are the collapsed typed dependency relations extracted from the corpus, excluding the *"root"* relations, as explained in Section 4.2.2.

Table 4.2 illustrates top 10 terms of the most probable topics generated by aforementioned topic models on the Reuters corpus. The first column shows the words generated by LDA. Some words in this topic are ambiguous and can have multiple meanings. To identify the correct meaning of each word, one needs to consider other words in the topic. For example, the word *"share"* has many meanings. Observing other words in the topic, such as *"bank"* and *"profit"*, helps to identify the correct meaning of the word *"share"* that is *"assets belonging to an individual"*. The second

[5] http://www.cs.princeton.edu/∼blei/lda-c

[6] http://www.research.att.com/∼lewis

LDA	LDA on generalized words using synonyms	The Bigram Topic Model	The HPSG-based Topic Model
bank	financial	reconstruction plans	money funds
profit	international	debt repayment	overseas investments
foreign	net	private institute	raising stake
share	government	traders reported	foreign deposits
federal	billion	existing research	commercial banks
japanese	withdraw	payments improve	buyout transaction
policy	currency	banking office	lack assets
rates	rise	borrowing occurred	stock exchange
money	sale	federal supervisory	account balance
shares	february	bank consultancies	bank regulation

Table 4.2: Top 10 terms of the most probable topic, generated by four topic models: LDA, LDA on generalized words using synonyms, the Bigram topic model, and the HPSG-based topic model from Reuters corpus.

column shows the results of LDA on generalized words using synonyms. These words are similar to the words in the first column and still suffer from ambiguity. The terms generated by the Bigram Topic Model and the HPSG-based topic model are shown in columns three and four, respectively. These topic models have less ambiguity, given the fact that they generate terms that include pairs of words that are more descriptive than single words. In addition, as opposed to the Bigram topic model, terms generated by the HPSG-based topic model are not only limited to consecutive pairs of words of a sentence, but they also contain pairs of related words with gaps.

Given the text corpora, we compare our work with other topic models based on the following criteria:

- High likelihood on a held-out test set (perplexity) [16].
- Coherent distribution of words learned by individual topics [66].
- Accurate distribution of topics over words.
- Stable distribution of topics over words across samples [74].

These criteria and experimental results are discussed in the subsequent sections.

4.6.1 Perplexity

Perplexity is the most common criterion to evaluate the quality of topic models [47]. Perplexity measures the cross-entropy between the term distribution learned by the topic model and the distribution of terms in an unseen test document. Thus, lower perplexity score indicates that the model is better in predicting distribution of the test document [16, 25]. We evaluate perplexity as a function of number of topics for both Associated Press and Reuters corpora. We trained the topic models on 90% of the corpus to estimate the held out probability of previously unseen 10% of the corpus. We compute the perplexity of the held-out test set with respect to the HPSG-based topic model by

$$perplexity(\mathbf{R}_{test}) = exp\left(-\frac{\sum_{d=1}^{Q} logp(\vec{R}_d)}{\sum_{d=1}^{Q} |\vec{R}_d|}\right), \tag{4.9}$$

where $\mathbf{R}_{test}$ is the test corpus with Q documents, $\vec{R}_d$ denotes the set of collapsed typed dependency relations in document $d \in \mathbf{R}_{test}$, $|\vec{R}_d|$ is the total number of collapsed

typed dependency relations in document d, and $p(\vec{R}_d)$ is the probability estimate assigned to $\vec{R}_d$ by the HPSG-based topic model.

The results are illustrated in Figures 4.4 and 4.5. The x-axis shows the number of topics (K) used in each model; the y-axis shows the perplexity. These figures clearly indicate that the perplexity of our proposed topic model drastically decreases the perplexity of LDA and LDA on generalized words using synonyms. Moreover, the perplexity of our proposed topic model is slightly better than the perplexity of the Bigrams Topic Model. The improvement in perplexity is due to using the collapsed typed dependency relations instead of bag-of-words to represent the corpus. In our method, every word is followed by another word that is semantically or syntactically related to it. This representation leads to better estimates for unseen documents, and thus lower perplexity.

4.6.2 Topic coherence

Topic coherence measures the integrity or coherence of top terms in a topic generated by a topic model. In other words, top n terms generated by topic k, denoted by $\vec{\Phi}_k = \{r_1, r_2, \cdots, r_n\}$, are coherent if they are semantically similar. We use the normalized pairwise mutual information (NPMI) [49] to calculate the average sum of semantic similarity scores between every pair of top n terms of the topics generated from the Associated Press corpus. Mathematically, the NPMI of top n topic terms is computed by

$$NPMI(\vec{\Phi}_k) = \sum_{j=2}^{n} \sum_{i=1}^{j-1} \frac{\log \frac{p(r_i, r_j)}{p(r_i) \cdot p(r_j)}}{-\log p(r_i, r_j)}, \tag{4.10}$$

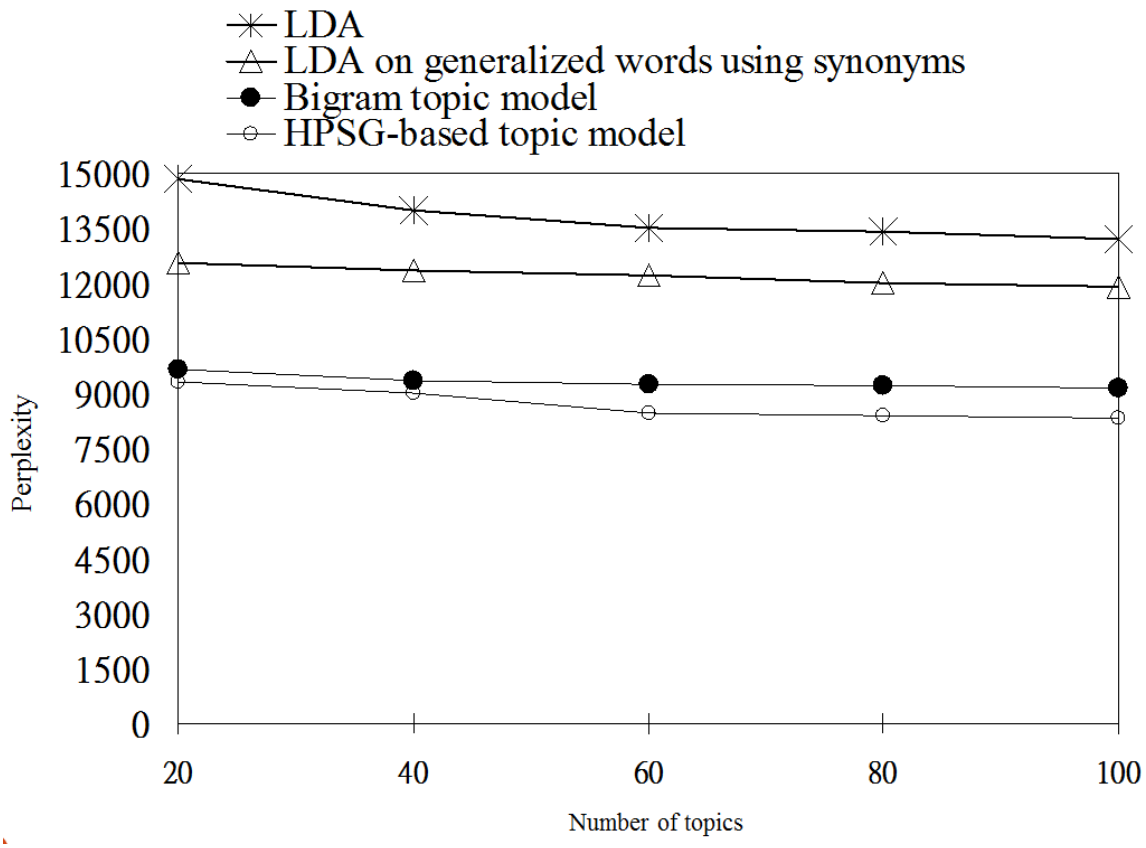

Figure 4.4: Perplexity as a function of number of topics, using LDA, LDA on generalized words using synonyms, the Bigram topic model, and the HPSG-based topic model on the Association Press corpus.

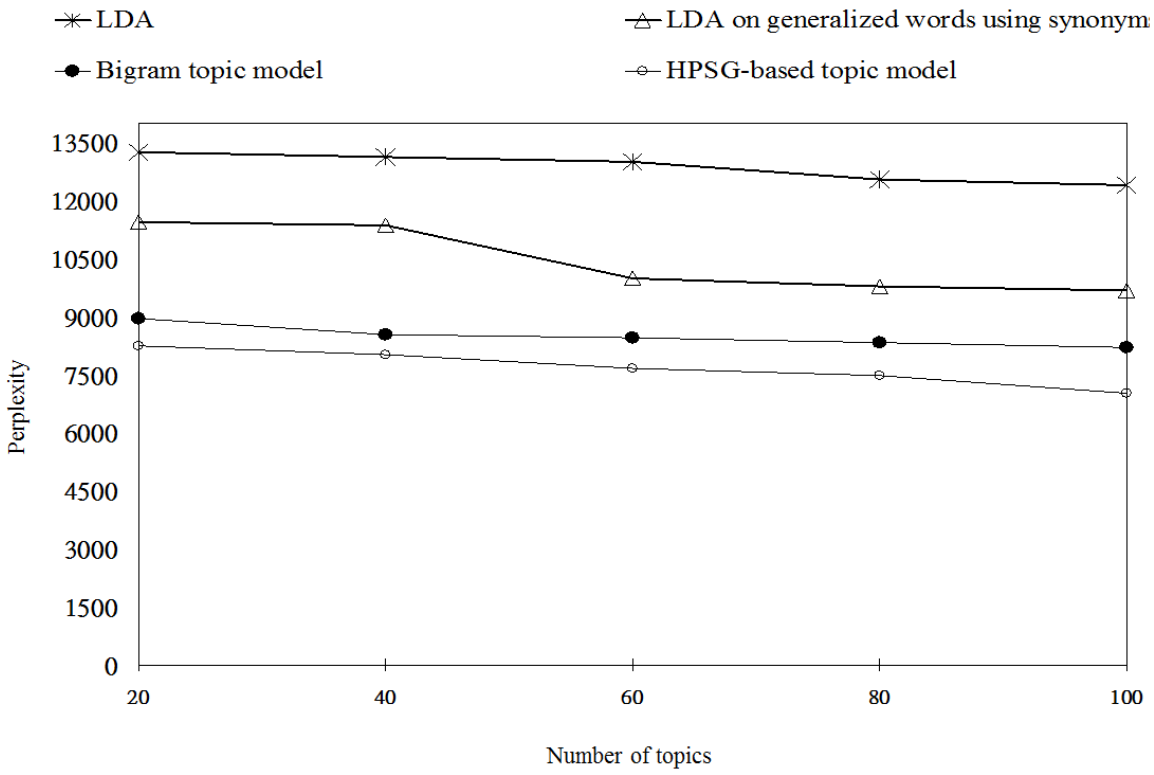

Figure 4.5: Perplexity as a function of number of topics, using LDA, LDA on generalized words using synonyms, the Bigram topic model, and the HPSG-based topic model on Reuters corpus.

where $p(x)$ is the probability that term x appears in a corpus. Wikipedia[7] is used as our training corpus. We compared the topic coherence of top 50 words from 20 topics generated by LDA, LDA on generalized words using synonyms, the Bigram topic model, and the HPSG-based topic model on Reuters corpus. The results are shown in Tables 4.3 and 4.4. LDA on generalized words using synonyms results in more coherent topic distribution over words. This coherence is due to the fact that we replaced conceptually related words with one general word, prior to modeling the topic assignments. The HPSG-based topic model generates slightly more coherent topic distributions over words than the Bigram topic model. The HPSG-based topic model performs comparable to LDA in topic coherence.

4.6.3 Accuracy

The accuracy of a topic model is the degree of closeness of the topic distribution over terms of a test corpus to actual topic distribution over terms of a topic-labeled corpus. Note that calculating accuracy depends on the availability of the topic-labeled corpus.

We assume that the test corpus $\mathbf{R}_{test}$ consists of Q documents $\mathbf{R}_{test} = \{d_1, d_2, \cdots, d_Q\}$. Each document consists of H actual topic labels, denoted by $L = \{l_1, l_2, \cdots, l_H\}$, where each $l_i \in L$ represents an actual topic label for the document. As mentioned earlier, a topic model generates K topics, where each topic is a distribution over R relations, denoted by $\vec{\Phi}_k = \{r_1, r_2, \cdots, r_R\}$. The accuracy score of the topic model is calculated by computing

[7]http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

$$Accuracy = \frac{\sum_{i=1}^{Q} min_{k=1,\cdots,K} d(\vec{\Phi}_k, L)}{Q}, \tag{4.11}$$

where $d(\vec{\Phi}_k, L)$ denotes the semantic similarity between two sets of $\vec{\Phi}_k$ and L. This semantic similarity is measured using the Lesk algorithm. The Lesk algorithm uses dictionary definitions of two terms in a pair and counts the number of terms that are shared between two definitions. The more overlapping the definitions are, the more related the terms are[8].

We compared the accuracy of LDA, LDA on generalized terms using synonyms, the Bigram topic model, and the HPSG-based topic model on a subset of Reuters corpus that contains topic labeled documents. As illustrated in Tables 4.5 and 4.6, these algorithms are comparable in terms of accuracy. However, LDA is slightly better. The HPSG-based topic model beats the Bigram topic model in terms of accuracy. The reason is due to the fact that our topic model not only considers consecutive relations between words (the Bigram topic model), but also nonconsecutive relations between words.

4.6.4 Stability

Stability is the similarity of topic distributions over relations across different samples [74]. We follow the algorithm proposed by Rosen-Zvi *et al.* [74] to find the best one-to-one topic alignment across samples. The algorithm finds the best aligned topic pair by calculating $min_{j=1,\cdots,K} d(S_1, S_2)$, where $d(S_1, S_2)$ denotes symmetrized Kullback Leibler (KL) divergences between the K topic distributions over relations from

[8]The Lesk toolkit is available at http://text-similarity.sourceforge.net

Topic model	Coherence
LDA	0.51
LDA on generalized terms using synonyms	0.54
The Bigram Topic Model	0.50
The HPSG-based Topic Model	0.52

Table 4.3: The average topic coherence of top 50 terms of 20 topics generated from Associated Press corpus.

Topic model	Coherence
LDA	0.41
LDA on generalized terms using synonyms	0.42
The Bigram Topic Model	0.39
The HPSG-based Topic Model	0.40

Table 4.4: The average topic coherence of top 50 terms of 20 topics generated from Reuters corpus.

Topic model	Accuracy
LDA	0.34
LDA on generalized terms using synonyms	0.32
The Bigram Topic Model	0.29
The HPSG-based Topic Model	0.33

Table 4.5: Average accuracy of topic distribution over terms from a subset of topic-labeled Associated Press.

Topic model	Accuracy
LDA	0.225
LDA on generalized terms using synonyms	0.220
The Bigram Topic Model	0.221
The HPSG-based Topic Model	0.223

Table 4.6: Average accuracy of topic distribution over terms from a subset of topic-labeled Reuters.

samples S_1 and S_2. KL divergence is calculated by $d(S_1, S_2) = \sum_{x \in X} S_1(x) log(S_1(x)/S_2(x))$, where X represents the set of relations in the samples [9]. We compare the stability of topic distributions over relations across samples, generated by the HPSG-based topic model and LDA on the Reuters corpus. The results, illustrated in Tables 4.7 and 4.8, show that our proposed topic model is comparably as stable as LDA in producing similar topic distributions over words across multiple samples. Similar results were obtained using the Bigram topic model.

4.7 Summary

We proposed a novel method that incorporates syntactic and semantic structures of text documents into probabilistic topic models. This representation has several benefits. It captures relations between consecutive and nonconsecutive words of text documents. In addition, the labels of the collapsed typed dependency relations help to eliminate less important relations, i.e., relations involving prepositions. Also, words of text documents, regardless of their parents in the collapsed typed dependency parse trees, are distinguished in topic assignment. Furthermore, our experimental studies show that the proposed topic model significantly outperforms LDA and is also better than the Bigram topic model in terms of perplexity. We also show that our model achieves comparable results with other models in terms of stability, coherence, and accuracy. Besides, the results from our topic model have less ambiguity, given the fact the generated terms include pairs of words that are more descriptive than single words.

Moreover, we introduced a method to enforce topic similarity to conceptually

similar words. As a result, this algorithm led to more coherent topic distribution over words.

Topics from sample 1	Best aligned topics from sample 2	Best KL
Topic 1	Topic 14	0.834
Topic 2	Topic 20	1.630
Topic 3	Topic 13	0.835
Topic 4	Topic 3	0.730
Topic 5	Topic 11	0.454
Topic 6	Topic 18	0.951
Topic 7	Topic 19	0.450
Topic 8	Topic 18	0.760
Topic 9	Topic 15	0.420
Topic 10	Topic 13	0.939
Topic 11	Topic 5	0.526
Topic 12	Topic 17	0.439
Topic 13	Topic 12	0.953
Topic 14	Topic 7	1.053
Topic 15	Topic 6	1.013
Topic 16	Topic 14	1.139
Topic 17	Topic 5	1.041
Topic 18	Topic 9	1.172
Topic 19	Topic 10	1.026
Topic 20	Topic 17	1.226
Average		0.87955

Table 4.7: Topic stability across two different runs of the HPSG-based topic model on Reuters corpus.

Topics from sample 1	Best aligned topics from sample 2	Best KL
Topic 1	Topic 5	0.821
Topic 2	Topic 12	1.073
Topic 3	Topic 8	0.533
Topic 4	Topic 19	0.721
Topic 5	Topic 3	1.031
Topic 6	Topic 18	1.050
Topic 7	Topic 7	0.836
Topic 8	Topic 8	0.754
Topic 9	Topic 15	0.428
Topic 10	Topic 13	0.765
Topic 11	Topic 7	0.818
Topic 12	Topic 8	0.798
Topic 13	Topic 6	0.961
Topic 14	Topic 5	0.764
Topic 15	Topic 12	1.161
Topic 16	Topic 8	0.867
Topic 17	Topic 6	0.791
Topic 18	Topic 4	0.921
Topic 19	Topic 18	1.064
Topic 20	Topic 8	1.091
Average		0.8624

Table 4.8: Topic stability across two different runs of LDA on Reuters corpus.

Chapter 5

Topic Modeling using Term Importance

Topic models such as Latent Dirichlet Allocation (LDA) are powerful tools to uncover hidden thematic structures of documents. Typically, LDA uses symmetric Dirichlet priors, neglecting the influence of term importance in documents. In this chapter [33], we propose two generative topic models that incorporate term importance into latent topic variables by boosting the probability of important terms and consequently decreasing the probability of less important terms to better reflect the themes of documents. In essence, we assign weights to terms by employing corpus-level and document-level approaches. We incorporate term importance using a nonuniform base measure for an asymmetric prior over topic-term distributions in the LDA framework. This leads to better estimates for important terms that occur less frequently in documents. Our experimental studies show that the proposed models outperform LDA and the Bigram topic model in terms of perplexity and topic coherence. Ad-

ditionally, our topic models show better performance than LDA in text classification tasks.

5.1 Introduction

Topic modelling is a powerful statistical tool to uncover hidden thematic structures and multi-faceted summaries of documents or other discrete data. Most topic models, such as Latent Dirichlet Allocation (LDA) [16], consider documents to be a weighted mixture of topics, where each topic is a multinomial distribution over terms. The inferred topic model assigns a high probability to the topics of a corpus. In addition, the highest probable terms in each topic provide important terms that summarize the themes of the corpus [16, 40].

Typically, LDA-based topic models use fixed symmetric priors, and consider only frequencies of terms in the corpus to estimate posteriors of latent variables [2, 82]. This strategy makes sense from a point of view of computational efficiency [80], but it does not utilize additional information about how important a term is in the context of a corpus or with respect to some external knowledge to properly identify more relevant terms to describe a topic. As a result, some top-ranking terms in a topic may contain terms that are frequent but not important to the topic. For instance, in a topic about *"sport"*, it is reasonable for highly important terms such as *"athletics"*, *"arena"*, and *"track"* to have a high probability, but less important terms like *"announce"*, *"time"*, and *"year"* to have a low probability. However, if *"time"* and *"year"* frequently appear in the documents about sports, these terms may obtain a high probability in the distribution of the topic.

In this chapter, we propose two novel generative topic models, *topic model using corpus-level term importance (TMCTI)* and *topic model using document-level term importance (TMDTI)*. In our topic models, we consider an asymmetric Dirichlet prior over the topic-term distributions, constructed from additional information about term importance. In essence, we capture this additional information by adopting *corpus-level* and *document-level* term importance measures. Consequently, we obtain a topic model where terms can be a priori more or less probable in topics. We present experiments using several topic models on two datasets. Our experiments show that our topic models not only successfully incorporate term importance into LDA, but also achieve better estimates for terms that occur rarely in the corpus. In addition, our topic models improve perplexity and topic coherence of LDA and the Bigram topic model. Also, our topic models result in higher accuracy than LDA in text classification.

The structure of this chapter is as follows: In Section 5.2, we explain the methods we use to measure term importance. In Section 5.3, we discuss our proposed topic models using term importance. In Section 5.4, we demonstrate the effectiveness of our approach through experiments. Finally, Section 5.5 concludes the chapter.

5.2 Measuring term importance

Term importance has long been beneficial in a variety of applications in natural language processing and text mining [7, 47, 53]. We categorize the approaches we use to measure term importance into two groups: *corpus-level* and *document-level*. Note that it is possible to attribute term importance measures with natural linguistic features.

For example, all-capitalized, bolded, underlined, or italic terms sometimes offer important cues about term significance [6, 53]. Moreover, it is possible to employ other term importance measures, however, an investigation of these measures is beyond the scope of this thesis. Notice that the term importance scores obtained by following term importance measures are further normalized by scaling between zero and one.

5.2.1 Corpus-level term importance measures

Corpus-level term importance measures determine importance of a term across a corpus, as discussed below.

5.2.1.1 A Wikipedia-based term importance measure

Wikipedia-based measures have been proved to be beneficial in natural language processing applications [7, 37]. We adopt the approach proposed by Bendersky *et al.* [7] to compute term importance by using the statistics of an external data source. We use Wikipedia article titles[1], as our external resource. Due to the large volume and the high diversity of topics covered by Wikipedia, it is often assumed that important terms will appear in article titles in Wikipedia [7]. We calculate the importance of term t, $g(t)$, by counting the number of times term t occurs within a Wikipedia title, and normalize it by scaling between zero and one. We use Laplace smoothing to assign positive weights to all terms whether or not they are observed in Wikipedia titles.

[1]Available at http://dumps.wikimedia.org/enwiki/latest/

5.2.1.2 An idf-Wikipedia-based term importance measure

The Wikipedia-based approach is only dependent on Wikipedia. Below we use the inverse document frequency of term t, $idf(t)$, to smooth the Wikipedia-based score. The importance of term t, denoted by $I(t)$, is defined as:

$$I(t) = g(t) \times idf(t), \tag{5.1}$$

where $g(t)$ is defined in Section 5.2.1.1 and $idf(t) = log(M/df_t)$, where M is the total number of documents in the corpus and df_t is the number of documents containing term t [47]. As a result, rare terms across documents tend to gain a higher score and common terms like *"the"*, a lower score.

5.2.2 Document-level term importance measures

Document-level term importance measures determine the importance of a term in a document.

5.2.2.1 A tfidf-based term importance measure

Term frequency-inverse document frequency (tfidf) [47] is a statistical measure that increases proportionally to the frequency of a term in a document but lessens by the frequency of the term among documents in the corpus. The $tfidf$ score of a term t in document d, represented by *tfidf(t,d)*, is defined as

$$tfidf(t, d) = tf_{t,d} \times \log \frac{M}{df_t}, \tag{5.2}$$

where $tf_{t,d}$ measures the ratio of the number of times term t appears in document d to the total number of terms in document d, and M is the total number of documents in a corpus, and df_t is the number of documents containing term t.

5.2.2.2 A tfidf-Wikipedia-based term importance measure

The tfidf-based approach, explained in Section 5.2.2.1, is rigidly dependent on documents. It assigns scores to terms of documents based on a single data source. Alternatively, we adopt the approach proposed by Bendersky *et al.* [7] to compute term importance by combining the statistics of the underlying documents, i.e., $tfidf$, with the statistics of an external data source, i.e., Wikipedia, to achieve a more accurate score. We define $I(t, d)$, the importance of term t in document d, as:

$$I(t, d) = tfidf(t, d) \times g(t). \tag{5.3}$$

where $g(t)$ is a Wikipedia-based score defined earlier.

5.3 Proposed probabilistic topic models

We assume that a corpus consists of M documents denoted by $\{d_1, d_2, \cdots, d_M\}$. Each document d contains N_d words denoted by $\{w_{d,1}, w_{d,2}, \cdots, w_{d,N_d}\}$, where each word is the basic unit of discrete data belonging to a vocabulary of V terms. In addition, each term $w_{d,n} \in d$ is assigned a latent topic $z_{d,n} = k$. Each of these topics $k \in \{1, 2, \cdots, K\}$ is associated with a multinomial distribution $(\vec{\Phi}_k)$ over V terms. In addition, each document d is a multinomial distribution $(\vec{\Theta}_d)$ over K topics, where K is the number of topics in the corpus. We propose two probabilistic topic

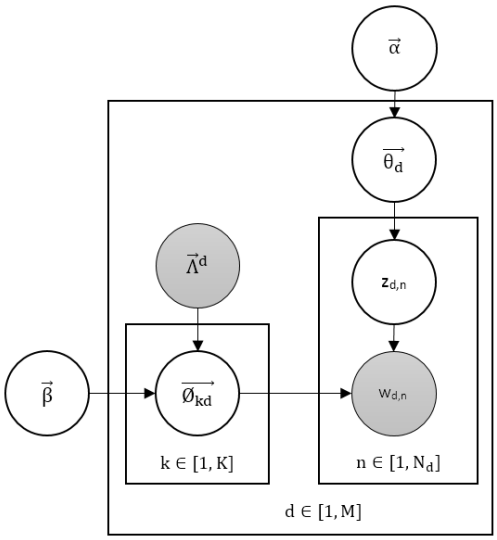

Figure 5.1: Graphical model representation of the topic model using corpus-level term importance measures (TM-CTI).

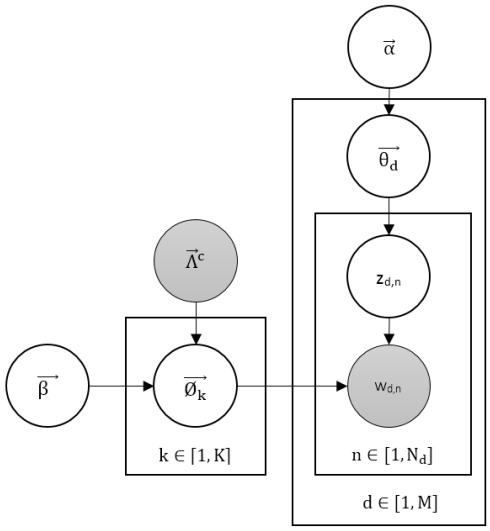

Figure 5.2: Graphical model representation of the topic model using document-level term importance measures (TMDTI).

models to incorporate additional information about term importance into the LDA framework to boost the probability of important terms and decrease the probability of less important terms in topics.

5.3.1 Topic model using corpus-level term importance (TMCTI)

Let $\vec{\Lambda}^c = \{\Lambda_1^c, \Lambda_2^c, \cdots, \Lambda_V^c\}$ represent the corpus-level importance scores of the V terms in the vocabulary, where $\Lambda_t^c \in \vec{\Lambda}^c$ denotes the importance score of term t, and $0 < \Lambda_t^c < 1$. TMCTI is designed based on the LDA framework. The generative procedure of TMCTI is formally stated in Algorithm 2, and graphically illustrated in Figure 5.1.

Algorithm 2 Generative process for topic model using corpus-level term importance (TMCTI).

1: **for** each topic $k \in [1, K]$ **do**
2: Generate $\vec{\Phi}_k \sim Dir(\vec{\beta} \otimes \vec{\Lambda}^c)$, where $\otimes$ is an element-wise multiplication.
3: **for** each document $d \in [1, M]$ **do**
4: Generate $\vec{\Theta}_d \sim Dir(\vec{\alpha})$
5: Generate document length $N_d \sim poisson(\xi)$
6: **for** each word $n \in [1, N_d]$ in document d **do**
7: Generate topic index $z_{d,n} \sim Mult(\vec{\Theta}_d)$
8: Generate word $w_{d,n} \sim Mult(\vec{\Phi}_{z_{d,n}})$

This algorithm is similar to LDA in generating each document d by drawing a

distribution over topics ($\vec{\Theta}_d$), generated from a Dirichlet distribution with a prior $\vec{\alpha}$. However, TMCTI differs from LDA in using an asymmetric Dirichlet prior $\vec{\beta} \otimes \vec{\Lambda}^c$, where β is a uniform symmetric vector determining how concentrated the probability mass of a sample from a Dirichlet distribution is likely to be and λ^c is the corpus level term importance vector $\vec{\Lambda}^c$, for topic-term distributions $\vec{\Phi}_k$. Since the terms are drawn from $\vec{\Phi}_k$, higher Λ_t^c values mean that term t is more likely in topic k. In this procedure, we first observe the corpus-level term importance vector $\vec{\Lambda}^c$. Then, the asymmetric prior, $\vec{\beta} \otimes \vec{\Lambda}^c$ is an element-wise multiplication[2] of $\vec{\beta}$ and $\vec{\Lambda}^c$ as shown in Line 2 of Algorithm 2. For example, suppose a corpus has three terms in the vocabulary with term importance scores given by $\vec{\Lambda}^c = \{0.1, 0.2, 0.3\}$. Also, assume $\vec{\beta} = \{0.1, 0.1, 0.1\}$, then $\vec{\beta} \otimes \vec{\Lambda}^c$ would be $\{0.01, 0.02, 0.03\}$.

TMCTI fulfils our requirement that the uniform symmetric Dirichlet prior $\vec{\beta}$ for $\vec{\Phi}_k$ is replaced with the nonuniform asymmetric prior $\vec{\beta} \otimes \vec{\Lambda}^c$ with base measure $\vec{\Lambda}^c$. Thus, more important terms have a higher chance to be generated. The dependency of $\vec{\Phi}_k$ on both $\vec{\beta}$ and $\vec{\Lambda}^c$ is indicated by directed edges from $\vec{\Lambda}^c$ and $\vec{\beta}$ to $\vec{\Phi}_k$ in the plate notation in Figure 5.1.

5.3.1.1 Parameter estimation

The only observed variables are the words in the corpus $\vec{W}$ and the corpus-level term importance vector $\vec{\Lambda}^c$. The corpus-level term importance vector is observed in the preprocessing step of our topic modelling procedure, using the corpus-level approaches explained in Section 5.2.1. We assume an asymmetric prior $\vec{\beta} \otimes \vec{\Lambda}^c$ for

[2]An element-wise multiplication, denoted by $\otimes$ in Algorithm 2, of two vectors is element by element multiplication of the vectors.

$\underline{\Phi}$, and a symmetric prior $\vec{\alpha}$ for $\underline{\Theta}$, where $\underline{\Phi} = \{\vec{\Phi}_k\}_{k=1}^{K}$ and $\underline{\Theta} = \{\vec{\Theta}_d\}_{d=1}^{M}$. These priors are conjugate to the multinomial distributions $\underline{\Phi}$ and $\underline{\Theta}$ [40, 82]. Hence, we can use collapsed Gibbs sampling [40, 34, 43] to obtain approximate estimates for $\underline{\Phi}$ and $\underline{\Theta}$. The complete likelihood of the model is factored as: $p(\vec{W}, \vec{z}|\vec{\alpha}, \vec{\beta} \otimes \vec{\Lambda}^c) = p(\vec{W}|\vec{z}, \vec{\beta} \otimes \vec{\Lambda}^c)p(\vec{z}|\vec{\alpha})$. The first probability is an average over $\underline{\Phi}$:

$$p(\vec{W}|\vec{z}, \vec{\beta} \otimes \vec{\Lambda}^c) = \int_{\underline{\Phi}} p(\vec{W}|\vec{z}, \underline{\Phi})p(\underline{\Phi}|\vec{\beta} \otimes \vec{\Lambda}^c)d\underline{\Phi}. \tag{5.4}$$

The first term in Equation 5.4 is obtained as:

$$p(\vec{W}|\vec{z}, \underline{\Phi}) = \prod_{k=1}^{K}\prod_{t=1}^{V}(\Phi_{k,t})^{n_k^t}, \tag{5.5}$$

where n_k^t is the total number of times topic k is assigned to term t. By expanding $p(\underline{\Phi}|\vec{\beta} \otimes \vec{\Lambda}^c)$ as a Dirichlet distribution, we obtain:

$$p(\underline{\Phi}|\vec{\beta} \otimes \vec{\Lambda}^c) = \prod_{k=1}^{K}\frac{1}{B(\vec{\beta} \otimes \vec{\Lambda}^c)}\prod_{t=1}^{V}(\Phi_{k,t})^{\beta_t \times \Lambda_t^c - 1}, \tag{5.6}$$

where $B(.)$ is the Beta function. Substituting the first and the second terms of Equation 5.4 with Equations 5.5 and 5.6, and using the Dirichlet integral[3] yields:

$$\begin{aligned} p(\vec{W}|\vec{z}, \vec{\beta} \otimes \vec{\Lambda}^c) &= \int_{\vec{\Phi}_k}\prod_{k=1}^{K}\frac{1}{B(\vec{\beta} \otimes \vec{\Lambda}^c)}\prod_{t=1}^{V}(\Phi_{k,t})^{n_k^t + \beta_t \times \Lambda_t^c - 1}d\vec{\Phi}_k \\ &= \prod_{k=1}^{K}\frac{1}{B(\vec{\beta} \otimes \vec{\Lambda}^c)}\int_{\vec{\Phi}_k}\prod_{t=1}^{V}(\Phi_{k,t})^{n_k^t + \beta_t \times \Lambda_t^c - 1}d\vec{\Phi}_k \\ &= \prod_{k=1}^{K}\frac{B(\vec{n}_k + \vec{\beta} \otimes \vec{\Lambda}^c)}{B(\vec{\beta} \otimes \vec{\Lambda}^c)}, \end{aligned} \tag{5.7}$$

[3] $B(\vec{\gamma}) = \int_{\vec{x}}\prod_{i=1}^{N}(x_i)^{\gamma_i - 1}d_{\vec{x}}^N$

where $\vec{n}_k = \{n_k^t\}_{t=1}^V$.

$p(\vec{z}|\vec{\alpha})$ remains analogous to LDA and is obtained by:

$$p(\vec{z}|\vec{\alpha}) = \prod_{d=1}^{M} \frac{B(\vec{n}_d + \vec{\alpha})}{B(\vec{\alpha})}, \tag{5.8}$$

where $\vec{n}_d = \{n_d^k\}_{k=1}^K$.

We can derive the full conditional distribution for term $W_i = t$ in document $d = l$ generated by topic $z_i = k$, given the corpus $\vec{W} = \{W_i = t, \vec{W}_{-i}\}$ and $\vec{z} = \{z_i = k, \vec{z}_{-i}\}$ where z_i denotes the topic assignment for the ith term $W_i \in \vec{W}$ and $\vec{z}_{-i}$ is topic assignments for the rest of the terms $\vec{W}_{-i} \subset \vec{W}$:

$$\begin{aligned} p(z_i = k|\vec{z}_{-i}, \vec{W}) &= \frac{p(\vec{W}, \vec{z})}{p(\vec{W}, \vec{z}_{-i})} = \frac{p(\vec{W}, \vec{z})}{p(\vec{W}_{-i}|\vec{z}_{-i})p(w_i)} \cdot \frac{p(\vec{z})}{p(\vec{z}_{-i})} \\ &= \frac{B(\vec{n}_k + \vec{\beta} \otimes \vec{\Lambda}^c)}{B(\vec{n}_{k,-i} + \vec{\beta} \otimes \vec{\Lambda}^c)} \cdot \frac{B(\vec{n}_d + \vec{\alpha})}{B(\vec{n}_{d,-l} + \vec{\alpha})} \\ &= \frac{n_{k,-i}^t + \beta_t \times \Lambda_t^c}{\sum_{t=1}^V (n_{k,-i}^t + \beta_t \times \Lambda_t^c)} \cdot \frac{n_{d,-l}^k + \alpha}{\sum_{k=1}^K (n_{d,-l}^k + \alpha)}, \end{aligned} \tag{5.9}$$

where $n_{k,-i}^t$ is the total number of times topic k is assigned to term t, excluding the current one, $n_{d,-l}^k$ is the total number of terms in document d assigned to topic k, excluding the current assignment.

Finally, we need to calculate the multinomial parameter sets $\underline{\Phi}$ and $\underline{\Theta}$. Note that $p(\vec{\Phi}_k|\vec{W}, \vec{\beta} \otimes \vec{\Lambda}^c) = Dirichlet(\vec{\Phi}_k|\vec{n}_k + \vec{\beta} \otimes \vec{\Lambda}^c)$, and $p(\vec{\Theta}_d|\vec{W}, \vec{\alpha}) = Dirichlet(\vec{\Theta}|\vec{n}_d + \vec{\alpha})$. Using the expectation of the Dirichlet distribution ($Dirichlet(\beta) = \beta_i / \sum_i \beta_i$) yields:

$$\Phi_{k,t} = \frac{n_k^t + \beta_t \times \Lambda_t^c}{\sum_{t=1}^V (n_k^t + \beta_t \times \Lambda_t^c)}, \tag{5.10}$$

$$\Theta_{d,k} = \frac{n_k^d + \alpha}{n_d^{(.)} + K\alpha}, \tag{5.11}$$

where n_k^d is the number of terms from document d assigned to topic k, and $n_d^{(.)}$ is the total number of terms in document d.

5.3.2 Topic model using document-level term importance (TMDTI)

In TMDTI, different from LDA [16] and TMCTI explained in Section 5.3.1, for each document d and each topic k, a new topic-term distribution $\vec{\Phi}_{kd}$ is drawn. In LDA and TMCTI, $\underline{\Phi}$ is a $V \times K$ array of term probabilities given topics, where V is the size of the vocabulary and K is the number of topics. In TMDTI, $\underline{\Phi}$ is a three dimensional $V \times K \times M$ array of term probabilities for each topic for each document, where M is the number of documents. That is, $\underline{\Phi} = \{\vec{\Phi}_{kd}\}$, where $k = 1, \cdots, K$, $d = 1, \cdots, M$, and $\vec{\Phi}_{kd}$ is the topic distribution over V terms for document d and topic k. Moreover, the Dirichlet prior $\vec{\beta}_k$ for $\vec{\Phi}_{kd}$ is replaced with a document-specific Dirichlet prior $\vec{\beta}_k \otimes \vec{\Lambda}^d$ that is a nonuniform asymmetric prior with a concentration parameter $\vec{\beta}_k$ and a nonuniform base measure $\vec{\Lambda}^d$. Since the terms are drawn from $\vec{\Phi}_{kd}$, higher Λ_t^d values mean that term t is more likely in document d in topic k.

Algorithm 3 represents the TMDTI generative probabilistic process. This process is graphically illustrated in Figure 5.2. Similar to TMCTI, an element-wise multiplication of $\vec{\beta}_k$ and $\vec{\Lambda}^d$, as shown in Line 5 of Algorithm 3, is used to compute the asymmetric prior $\vec{\beta}_k \otimes \vec{\Lambda}^d$. Note that TMDTI is similar to LDA and TMCTI in generating a document d by drawing a distribution over topics $\vec{\Theta}_d$, which is in turn drawn

from a Dirichlet distribution with a prior $\vec{\alpha}$.

Algorithm 3 Generative process for topic model using document-level term importance (TMDTI).

1: **for** each document $d \in [1, M]$ **do**
2: Generate $\vec{\Theta}_d \sim Dirichlet(\vec{\alpha})$
3: Generate document length $N_d \sim poisson(\xi)$
4: **for** each topic $k \in [1, K]$ **do**
5: Generate $\vec{\Phi}_{kd} \sim Dirichlet(\vec{\beta}_k \otimes \vec{\Lambda}^d)$.
6: **for** each word $n \in [1, N_d]$ in document d **do**
7: Generate topic index $z_{d,n} \sim Mult(\vec{\Theta}_d)$
8: Generate word $w_{d,n} \sim Mult(\vec{\Phi}_{z_{d,n}d})$

TMDTI also fulfils our requirement that the uniform symmetric Dirichlet prior $\vec{\beta}_k$ for $\vec{\Phi}_{kd}$ is replaced with a document-dependent nonuniform asymmetric prior $\vec{\beta}_k \otimes \vec{\Lambda}^d$. Thus, more important terms have a higher chance to be generated.

5.3.2.1 Parameter estimation

The only observed variables are the words in the corpus $\vec{W}$ and the document-level term importance matrix $\vec{\Lambda}^d$ of the corpus. $\vec{\Lambda}^d$ is computed in the preprocessing step of our topic modelling procedure, using the approaches explained in Section 5.2.2. We assume an asymmetric prior $\vec{\beta}_k \otimes \vec{\Lambda}^d$ for $\underline{\Phi}$, and a symmetric prior $\vec{\alpha}$ for $\underline{\Theta}$. Due to the conjugacy of these priors to the multinomial distributions $\underline{\Phi}$ and $\underline{\Theta}$ [34, 82], we can use Gibbs sampling procedure to estimate the latent variables $\underline{\Phi}$ and $\underline{\Theta}$. Similar derivation procedure, explained in Section 5.3.1, is used to obtain the full conditional

distribution for following Gibbs sampling equation:

$$p(z_i = k|\vec{z}_{-i}, \vec{W}) = \frac{n^t_{k,d,-i} + \beta^d_{kt} \times \Lambda^d_t}{\sum_{t=1}^{V}(n^t_{k,d,-i} + \beta^d_{kt} \times \Lambda^d_t)} \cdot \frac{n^k_{d,-l} + \alpha}{\sum_{k=1}^{K}(n^k_{d,-l} + \alpha)}, \tag{5.12}$$

where $n^t_{k,d,-i}$ is the total number of times term t in document d is assigned to topic k, excluding the current one. Finally, similar to the procedure, explained in Section 5.3.1, the conditional distribution for $\underline{\Phi}$ is:

$$\Phi_{k,d,t} = \frac{n^t_{kd} + \beta^d_{kt} \times \Lambda^d_t}{\sum_{t=1}^{V}(n^t_{kd} + \beta^d_{kt} \times \Lambda^d_t)}, \tag{5.13}$$

where n^t_{kd} denotes the number of times term t is assigned to topic k in document d. The distributions over topics ($\underline{\Theta}$) is similar to the ones in LDA and TMCTI given by Equation 5.11.

5.3.3 Integrating the HPSG-based topic model into topic model using term importance

In Chapter 4, we propose the HPSG-based topic model [32] that enriches text documents with collapsed typed dependency relations to effectively acquire syntactic and semantic dependencies between consecutive and nonconsecutive words of text documents. Thus, we assume that the corpus is represented by R unique collapsed dependency relations between words, denoted by $\vec{R} = \{r_1, r_2, \cdots, r_R\}$. These relations are instances of the 48 grammatical relations, described in Chapter 4, each of which consists of two words. In the HPSG-based topic model, we used symmetric Dirichlet priors, neglecting the influence of term importance in dependency relations and thus

in documents. In this section, we propose to incorporate relation importance into the HPSG-based topic model by boosting the probability of important relations and consequently decreasing the probability of less important relations to better reflect the themes of documents. We assign weights to relations by employing corpus-level and document-level approaches. We incorporate relation importance using a nonuniform base measure for an asymmetric prior over topic-relation distributions in the HPSG-based topic model.

5.3.3.1 The HPSG-based topic model using corpus-level relation importance

Similar to TMCTI, explained in Section 5.3.1, we assume that $\vec{\Lambda}^c = \{\Lambda^c_1, \Lambda^c_2, \cdots, \Lambda^c_R\}$ represents the corpus-level importance scores of R relations in the set of unique collapsed typed dependency relations, where $\Lambda^c_r \in \vec{\Lambda}^c$ denotes the importance score of relation r, extracted from the corpus, and $0 < \Lambda^c_r < 1$.

The generative process for this topic model is similar to Algorithm 2. Each document d is generated by drawing a distribution over topics ($\vec{\Theta}_d$), generted from a Dirichlet distribution with a prior $\vec{\alpha}$. However, different from TMCTI, the corpus is represented by typed dependency relations. Therefore, each topic k is generated by drawing a distribution over relations ($\vec{\Theta}_k$), generated from an asymmetric prior $\vec{\beta} \otimes \vec{\Lambda}^c$, obtained from an element-wise multiplication of the corpus-level relation importance vector $\vec{\Lambda}^c$ and $\vec{\beta}$. Thus, the likelihood of the model is defined as: $p(\vec{R}, \vec{z}|\vec{\alpha}, \vec{\beta} \otimes \vec{\Lambda}^c) = p(\vec{R}|\vec{z}, \vec{\beta} \otimes \vec{\Lambda}^c)p(\vec{z}|\vec{\alpha})$, where the first probablity is:

$$p(\vec{R}|\vec{z},\vec{\beta}\otimes\vec{\Lambda}^c)=\int_{\underline{\Phi}}p(\vec{R}|\vec{z},\underline{\Phi})p(\underline{\Phi}|\vec{\beta}\otimes\vec{\Lambda}^c)d\underline{\Phi}. \tag{5.14}$$

The first term in Equation 5.14 is obtained as:

$$p(\vec{R}|\vec{z},\underline{\Phi})=\prod_{k=1}^{K}\prod_{r=1}^{R}(\Phi_{k,r})^{n_k^r}, \tag{5.15}$$

where n_k^r is the total number of times topic k is assigned to relation r. By expanding $p(\underline{\Phi}|\vec{\beta}\otimes\vec{\Lambda}^c)$ as a Dirichlet distribution, we obtain:

$$p(\underline{\Phi}|\vec{\beta}\otimes\vec{\Lambda}^c)=\prod_{k=1}^{K}\frac{1}{B(\vec{\beta}\otimes\vec{\Lambda}^c)}\prod_{r=1}^{R}(\Phi_{k,r})^{\beta_r\times\Lambda_r^c-1}, \tag{5.16}$$

where $B(.)$ is the Beta function.

Following the procedure explained in Section 5.3.1, the conditional distribution for $\underline{\Phi}$ is:

$$\Phi_{k,r}=\frac{n_k^r+\beta_r\times\Lambda_r^c}{\sum_{r=1}^{R}(n_k^r+\beta_r\times\Lambda_r^c)}, \tag{5.17}$$

The distributions over topics ($\underline{\Theta}$) is similar to the ones in LDA and TMCTI given by Equation 5.11.

5.3.3.2 The HPSG-based topic model using document-level relation importance

Similar to TMDTI, for each document d and each topic k, a new topic-term distribution $\vec{\Phi}_{kd}$ is drawn. However, the Dirichlet prior $\vec{\beta}_k$ for $\vec{\Phi}_{kd}$ is replaced with a

document-specific Dirichlet prior $\vec{\beta}_k \otimes \vec{\Lambda}^d$ that is a nonuniform asymmetric prior with a concentration parameter $\vec{\beta}_k$ and a nonuniform base measure $\vec{\Lambda}^d$, where the vector $\vec{\Lambda}^d$ represents the document-level relation importance. Since the relations are drawn from $\vec{\Phi}_{kd}$, higher Λ^d_r values mean that relation r is more likely in document d in topic k.

Similar to the procedure explained in Section 5.3.2, the conditional distribution for $\underline{\Phi}$ is:

$$\Phi_{k,d,r} = \frac{n^r_{kd} + \beta^d_{kr} \times \Lambda^d_r}{\sum_{r=1}^{R}(n^r_{kd} + \beta^d_{kr} \times \Lambda^d_r)}, \tag{5.18}$$

where n^r_{kd} denotes the number of times term r is assigned to topic k in document d. The distributions over topics ($\underline{\Theta}$) is similar to the ones in LDA and TMCTI given by Equation 5.11.

5.3.4 Efficiency

Efficiency is a function of the number of iterations and the cost of each iteration of Gibbs sampling. Both LDA [16, 40] and the Bigram topic model [80] require $O(KV)$ for each iteration of Gibbs sampling[4] [72, 88], where V is the number of terms in the vocabulary and K is the number of topics. TMCTI and TMDTI require the same time complexity for each iteration of Gibbs sampling. However, both algorithms require a preprocessing step to compute their asymmetric priors using corpus-level and document-level approaches. Due to the use of hash indexing to store the scores of

[4]For clarity the time complexity of the multinomial random number generator $mult()$ is assumed to be $O(1)$.

LDA	TMCTI-Wiki	TMCTI-idfWiki	TMDTI-tfidf	TMDTI-tfidfWiki
million	**cash**	**bank**	million	**investment**
billion	million	**company**	**company**	million
japanese	**bank**	**cash**	billion	**company**
offer	share	million	share	state
share	billion	share	**stock**	official
said	**investment**	**stock**	plan	billion
financial	**business**	union	contract	president
bank	**stock**	**pay**	**cash**	share
cash	commercial	**money**	**business**	**stock**
workers	president	**financial**	bid	**bank**

Table 5.1: Ranked list of 10 terms of the most probable topic, generated from the Associated Press corpus by following topic models: LDA, TMCTI using Wikipedia-based measure, TMCTI using idf-Wikipedia-based measure, TMDTI using tfidf-based measure and TMDTI using tfidf-Wikipedia-based measure.

words in Wikipedia, TMCTI needs a single iteration, $O(V)$, through the vocabulary to compute the corpus-level term importance vector. TMDTI requires an iteration for each document to compute the document-level term importance vectors. Thus, the time complexity of the preprocessing step of TMDTI is $O(MV)$, where M is the number of documents. Note that using hashing techniques to compute $tfidf$ [73, 75] can enhance the time complexity of TMDTI to $O(M \log V)$.

5.4 Experiments

In our experiments we use the Associated Press corpus[5] that consists of 2,246 Associated Press documents, 33,872 terms, and 454,370 collapsed typed dependency relations. In addition, we use Reuters-21578 Distribution 1.0[6] that includes 21,578 documents. Due to the skew distribution of these documents [4, 31, 68], we only use a collection of documents belonging to the set of the 10 most populous classes. This collection contains 10,789 documents, 15,996 terms, and 793,345 collapsed typed dependency relations. Text cleaning is performed, which includes removal of most punctuation marks except embedded apostrophes and underscores. Then, corpus-level and document-level term importance scores are computed using approaches explained in Section 5.2.

We conduct experiments to compare the performance of following topic models: LDA [16], the Bigram topic model [80], WLDA [87], TMCTI explained in Section 5.3.1, TMDTI explained in Section 5.3.2, bigram TMCTI, bigram TMDTI, and the HPSG-based topic model using relation importance. For bigram methods, terms are consecutive word pairs, occurring in the corpus. Similarly, for the HPSG-based topic model, terms are consecutive or nonconsecutive word pairs, occurring in the corpus. Thus, statistics presented in Sections 5.2 and 5.3 are based on such terms.

In WLDA [87], terms are weighed by measuring their information content using corpus-level and document-level measures. Corpus-level term importance is computed using

[5]http://www.cs.princeton.edu/∼blei/lda-c

[6]http://www.research.att.com/∼lewis

$$I(t) = -\log_2 p(t), \tag{5.19}$$

where $p(t)$ is estimated from observed frequencies in the corpus and is computed as the number of occurrences of term t in the corpus, divided by the total number of term occurrences in the corpus. The objective of using this formula is to give high-probability terms such as "the" low weights [87]. LDA using this term-weighting function is denoted as Log-WLDA [87].

In [87] document-level term importance scores are computed using pointwise mutual information (PMI) between term t and document d, which is defined as follows:

$$PMI(t, d) = \log_2 \frac{p(t|d)}{p(t)}. \tag{5.20}$$

Note that in [87], the PMI score between t and d is defined as $-\log_2 \frac{p(t|d)}{p(t)}$. Here we remove the minus sign to be consistent with the formal PMI definition [26]. We compute $p(t|d)$ by dividing the number of times term t occurs in document d by the number of term occurrences in d, and $p(t)$ by dividing the the number times t occurs in the corpus by the total number of term occurrences in the corpus. LDA using this document-dependent term weighting function is denoted as PMI-WLDA [87].

The topic models are trained with 1000 iterations of Gibbs sampling to obtain samples from the posterior distribution over all possible assignments of terms to topics $\vec{z}$ at several choices of number of topics K. Initial values for the hyperparameters α and β applied to all our experiments are $\alpha = 50.0/K$ and $\beta = 0.01$. Note that these parameters are default parameters of most LDA-based topic models, expected to result in a fine-grained decomposition of the corpus into topics [40].

Table 5.1 illustrates ranked lists of the top 10 terms of the most probable topic generated by the aforementioned unigram topic models on the Associated Press corpus. Observing terms such as *"cash"*, *"bank"*, and *"investment"* helps to conclude that the top theme of the corpus is about *"finance"*. In this table, **boldface** indicates important (i.e., highly related) terms, whereas the unrelated terms are underlined. The terms that are not in boldface nor underlined are fairly related to the *"finance"* theme. Notice that our models (i.e., the 2nd to 5th columns) produce fewer unrelated terms and more important terms than the original LDA model. In addition, the important terms are positioned higher and unrelated terms are lower in our models than in LDA. These observations validate the effectiveness of incorporating term importance into LDA.

Below, we compare our models with other topic models in terms of *perplexity* and *topic coherence*. In addition, we evaluate the performance of the topic models in text classification tasks.

5.4.1 Perplexity

Perplexity is the most common criterion to evaluate the quality of topic models [47]. Perplexity measures the cross-entropy between the term distribution learned by the topic model and the distribution of terms in an unseen test document. Thus, a lower perplexity score indicates that the model is better in predicting distribution of the test document [16].

We evaluate perplexity as a function of different numbers of topics K, where $K = 20$, $K = 40$, $K = 60$, $K = 80$, and $K = 100$. Our experimental studies

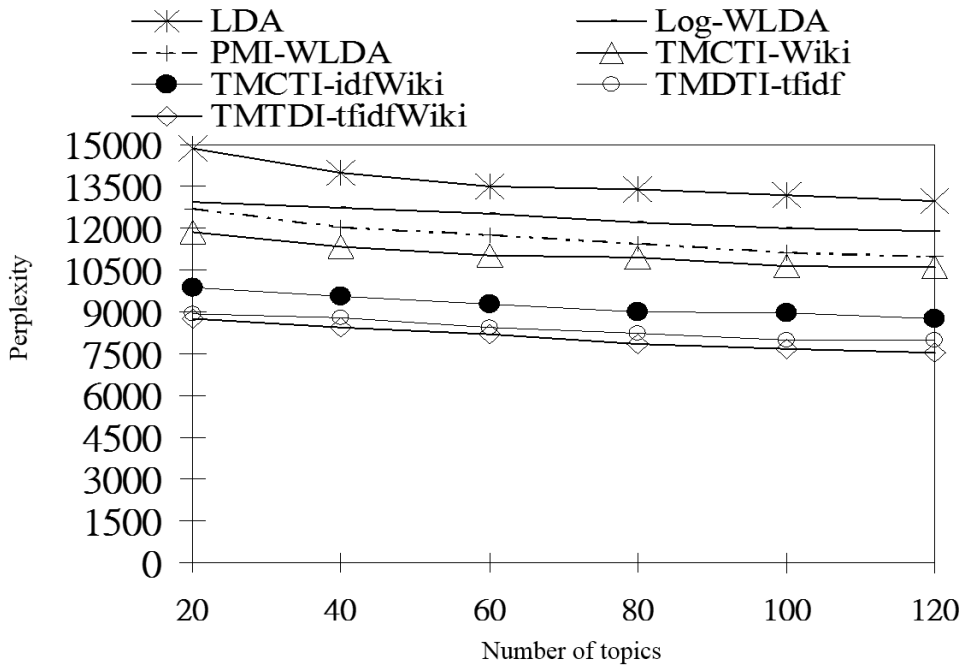

Figure 5.3: Perplexity as a function of number of topics, using LDA, Log-WLDA, PMI-WLDA, TMCTI Wikipedia-based, TMCTI idf-Wikipedia-based, TMDTI tfidf-based, and TMDTI tfidf-Wikipedia-based on the Association Press corpus.

show that increasing K values to more than 100 causes over-fitting that makes the perplexity of the new documents to explode. We train the topic models on 90% of the corpus to estimate the held out probability of remaining 10% of the corpus. We compute the perplexity of the held-out test set with respect to the topic model by

$$perplexity(D_{test}) = exp\left(-\frac{\sum_{d=1}^{Q}\log p(t_d)}{\sum_{d=1}^{Q}|t_d|}\right), \tag{5.21}$$

where D_{test} is the test corpus with Q documents, t_d denotes the set of terms in document $d \in D_{test}$, $|t_d|$ is the total number of terms in document d, and $p(t_d)$ is the probability estimate assigned to t_d by the topic model.

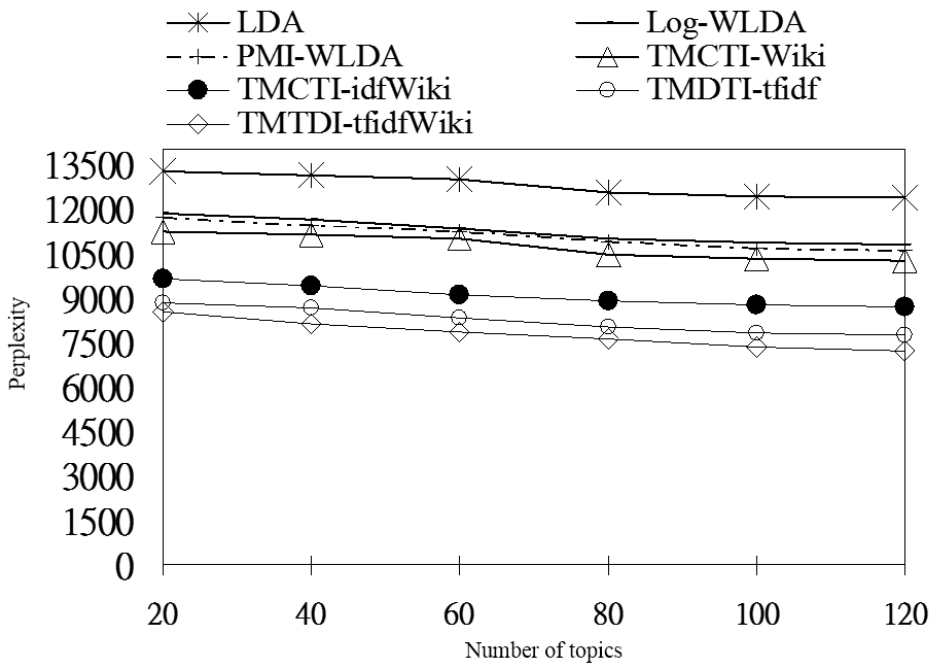

Figure 5.4: Perplexity as a function of number of topics, using LDA, Log-WLDA, PMI-WLDA, TMCTI Wikipedia-based, TMCTI idf-Wikipedia-based measure, TMDTI tfidf-based measure, and TMDTI tfidf-Wikipedia-based on the Reuters corpus.

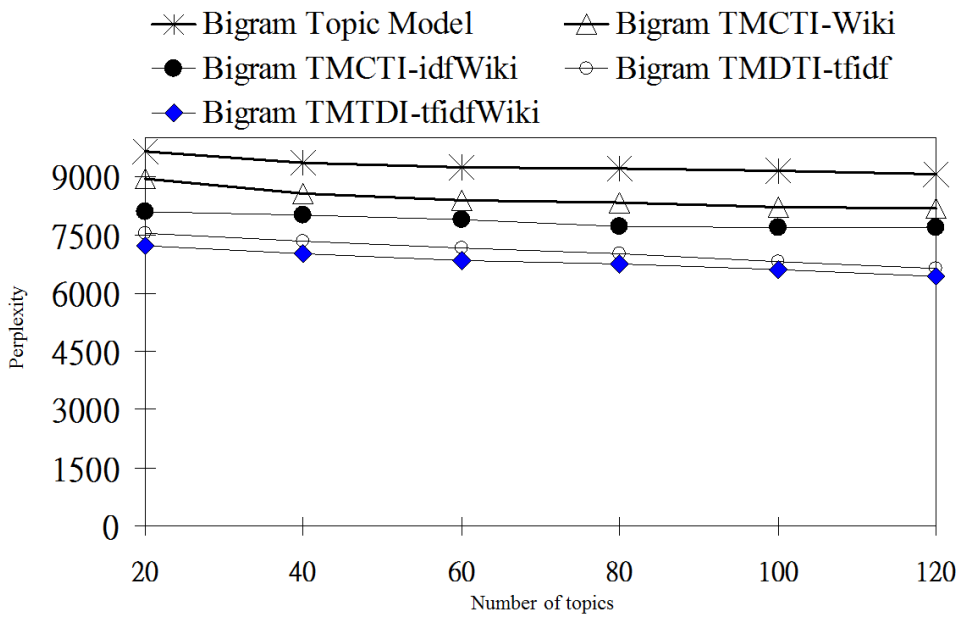

Figure 5.5: Perplexity as a function of number of topics, using Bigram topic model, Bigram TMCTI Wikipedia-based, Bigram TMCTI idf-Wikipedia-based, Bigram TMDTI tfidf-based, and Bigram TMDTI tfidf-Wikipedia-based on Association Press.

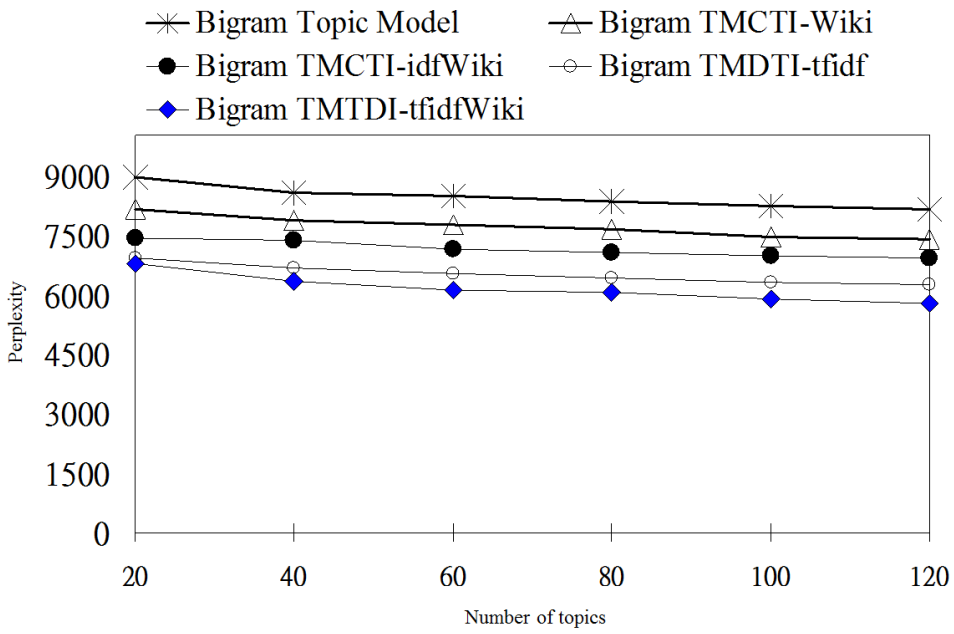

Figure 5.6: Perplexity as a function of number of topics, from Bigram topic model, Bigram TMCTI Wikipedia-based, Bigram TMCTI idf-Wikipedia-based, Bigram TMDTI tfidf-based, and Bigram TMDTI tfidf-Wikipedia-based on the Reuters corpus.

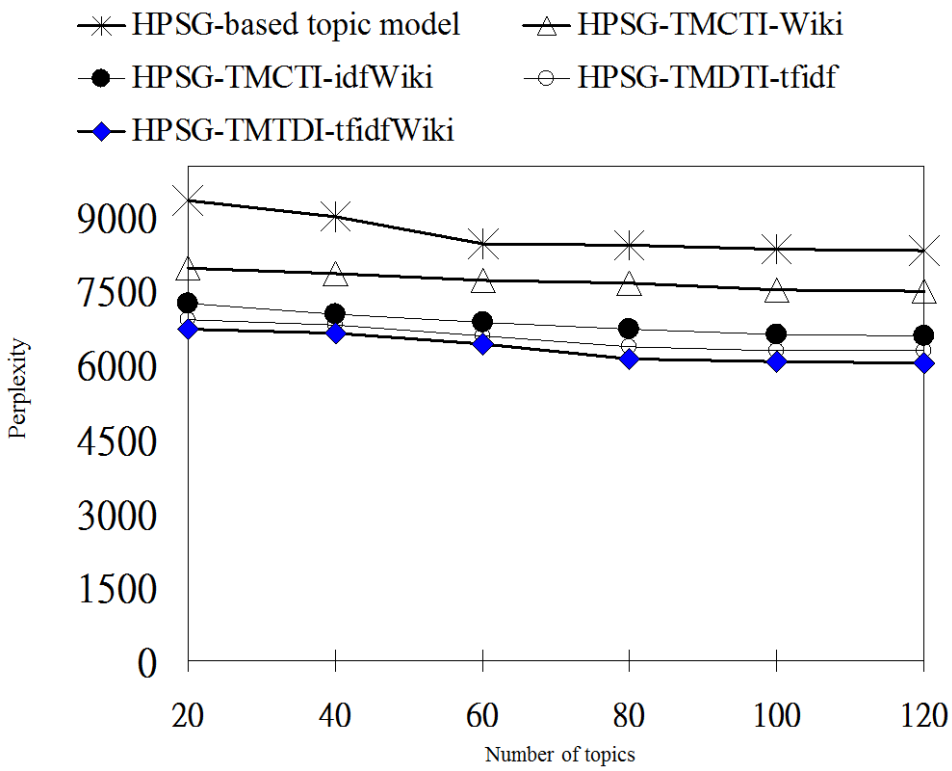

Figure 5.7: Perplexity as a function of number of topics, from the HPSG-based topic model, HPSG-TMCTI Wikipedia-based, HPSG-TMCTI idf-Wikipedia-based, HPSG-TMDTI tfidf-based, and HPSG-TMDTI tfidf-Wikipedia-based on the Associated Press corpus.

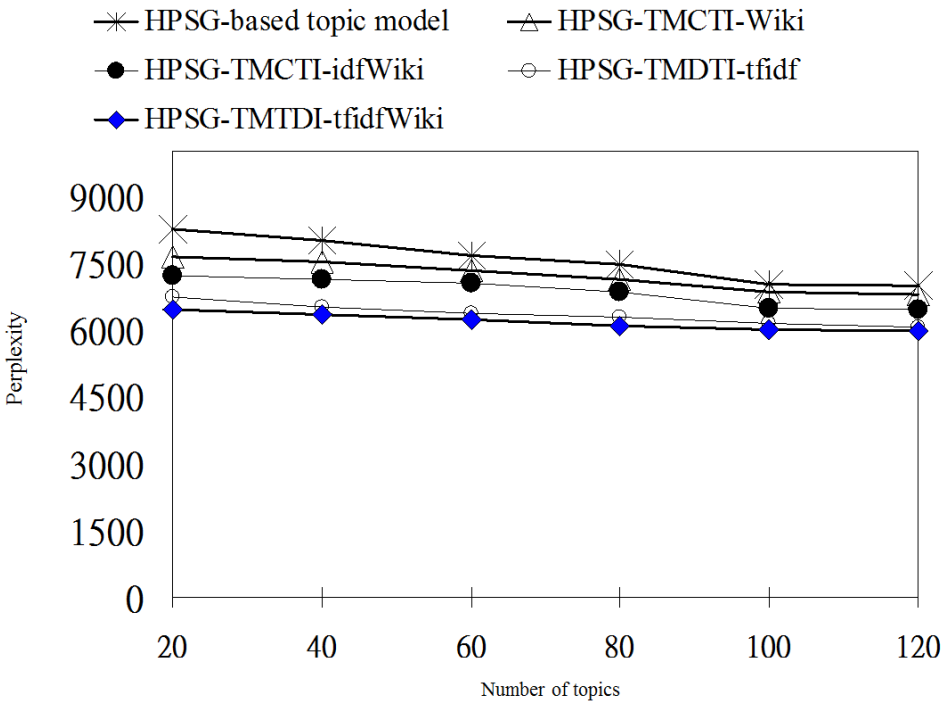

Figure 5.8: Perplexity as a function of number of topics, from the HPSG-based topic model, HPSG-TMCTI Wikipedia-based, HPSG-TMCTI idf-Wikipedia-based, HPSG-TMDTI tfidf-based, and HPSG-TMDTI tfidf-Wikipedia-based on the Reuters corpus.

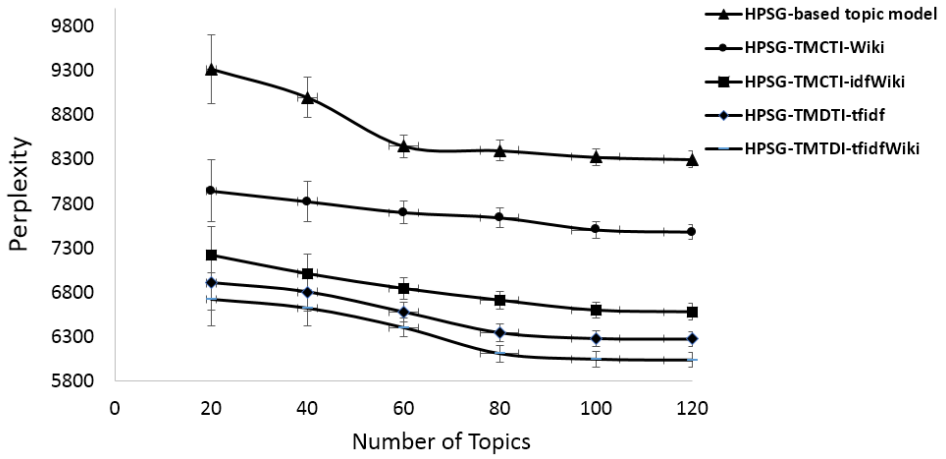

Figure 5.9: Perplexity as a function of number of topics and error bars, from the HPSG-based topic model, HPSG-TMCTI Wikipedia-based, HPSG-TMCTI idf-Wikipedia-based, HPSG-TMDTI tfidf-based, and HPSG-TMDTI tfidf-Wikipedia-based on the Associated Press corpus.

The results are illustrated in Figures 5.3, 5.4, 5.5, and 5.6. The x-axis shows the number of topics (K) used in each model; the y-axis shows the perplexity. These figures clearly indicate that the perplexity of our proposed topic models on both unigrams and bigrams improve over LDA. Among the four term importance measures, The topic model using tfidf-Wikipedia-based achieves the best performance in terms of perplexity, followed by tfidf-based, idf-Wikipedia-based, and solely Wikipedia-based measures. The results also indicate that topic models using document-level term importance measures perform better than the ones using corpus-level measures. Moreover, Figures 5.8 and 5.7 show the performance of the HPSG-based topic model using both corpus-level and document-level relation importance measures. Better estimates for held-out documents for our HPSG-based topic model using relation importance is due to augmenting topic models with both collapsed typed dependency relations and

term importance. We also assessed the quality of the predictions of HPSG-based topic model with standard deviations in 5-fold cross-validation. The results are illustrated in Figure 5.9.

5.4.2 Topic coherence

Topic coherence measures the integrity or coherence of top terms in a topic generated by a topic model. In other words, top n terms generated by topic k, denoted by $\vec{\Phi}_k = \{t_1, t_2, \cdots, t_n\}$, are coherent if they are semantically similar. We use the normalized pairwise mutual information (NPMI) [49] to calculate the average sum of semantic similarity scores between every pair of top 50 terms of the topics generated from the Associated Press corpus. Mathematically, the NPMI of top-n topic terms is computed by

$$NPMI(\mathbf{t}) = \sum_{j=2}^{n} \sum_{i=1}^{j-1} \frac{\log \frac{p(t_i,t_j)}{p(t_i)\cdot p(t_j)}}{-\log p(t_i,t_j)}, \tag{5.22}$$

where $p(x)$ is the probablity that term x appears in a corpus. Wikipedia[7] is used as our training corpus. Table 5.3 shows the average coherence scores of all the compared methods. Our proposed models lead to more coherent topics than the original LDA and the Bigram topic model. This coherence is due to the fact that we replaced the global constant prior for topic distributions over terms by a term importance prior. Thus, on the one hand, less important terms to a topic gain a lower probability and descend in the topic-term distributions, and on the other hand, important terms ascend in the topic-term distributions. As a result, more related terms to the topic

[7]http://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

Topic model	Coherence
LDA	0.512
TMCTI-Wiki	0.953
TMCTI-idfWiki	0.526
TMDTI-tfidf	0.519
TMDTI-tfidfWiki	0.834
Bigram topic model	0.507
Bigram TMCTI-Wiki	0.951
Bigram TMCTI-idfWiki	0.892
Bigram TMDTI-tfidf	0.743
Bigram TMDTI-tfidfWiki	0.835

Table 5.2: The average topic coherence of top 50 terms of 100 topics generated from the Associated Press corpus.

appear higher in the topic-term distributions leading to more coherent topics. We also observe that topic models that leverage Wikipedia produce more coherent topics than the ones that do not, with the solely Wikipedia-based measure achieves the highest coherence scores. This is due to the higher probabilities assigned to terms occurring in Wikipedia titles, which results in topics more likely being described by these title words.

5.4.3 Classification

Document classification is the task of assigning documents to class(es) [68]. A key issue in document classification is how to represents a document [54, 19]. The com-

Topic model	**Coherence**
LDA	0.413
TMCTI-Wiki	0.845
TMCTI-idfWiki	0.621
TMDTI-tfidf	0.476
TMDTI-tfidfWiki	0.857
Bigram topic model	0.391
Bigram TMCTI-Wiki	0.834
Bigram TMCTI-idfWiki	0.768
Bigram TMDTI-tfidf	0.713
Bigram TMDTI-tfidfWiki	0.825

Table 5.3: The average topic coherence of top 50 terms of 100 topics generated from the Reuters corpus.

mon approach to document representation is the bag-of-words representation, where a document is represented with a vector of the words that appear in it [19]. Often, the $tfidf$ values of words are used in a document vector in the bag-of-words representation so that common words across documents are less important than less frequent words. Alternatively, LDA-based approaches represent a document as a multinomial distribution over topics. This representation has been effective in text classification tasks [51]. We compare the use of our topic models for document representation against original LDA and $tfidf$-based bag-of-words methods in document classification tasks. We use the Reuters collection [4], that contains $7,770$ training and $3,019$ testing documents. The documents are multilabeled and can belong to one or more of the 10 classes. Results, shown in Table 5.4, are reported using the Naïve Bayes [68] classifier[8]. Note that the reported accuracy for the LDA-based approaches is the average accuracy that measures the percentage of correctly classified documents, obtained from experiments on different numbers of topics K, where $K = 20$, $K = 40$, $K = 60$, $K = 80$, and $K = 100$. Moreover, the table shows the standard deviation across 10-fold cross-validation runs for various topic models.

Empirical comparisons show that using topic models to represent documents improves the accuracy of text classification tasks. Moreover, incorporating term importance into topic models yields a higher accuracy than when using solely LDA-based topic models. Key to this improvement is incorporating term importance as a nonuniform base measure into the asymmetric prior over topic-term distributions. This leads to better estimates for less frequent important terms and consequently, better representation of the multinomial distribution over topics, and thus, better accuracy for

[8]Other classification algorithms yield similar results.

Document representation	**Accuracy**	**Standard deviation**
bag-of-words with tfidf	45.3%	1.28%
LDA	54.6%	0.94%
TMCTI-Wiki	55.7%	0.82%
TMCTI-idfWiki	58.3%	0.85%
TMDTI-tfidf	65.2%	1.546%
TMDTI-tfidfWiki	67.4%	0.78%
Bigram topic model	55.2%	1.75%
Bigram TMCTI-Wiki	55.4%	0.97%
Bigram TMCTI-idfWiki	56.2%	1.45%
Bigram TMDTI-tfidf	58.6%	1.06%
Bigram TMDTI-tfidfWiki	66.3%	1.45%

Table 5.4: Classification results for the Reuters corpus.

text classification.

5.5 Summary

We proposed two LDA-based topic models that do not consider a symmetric distribution prior over terms but rather adjust the prior by employing additional information about the importance of terms in a topic. The importance of terms in a topic is captured by corpus-level and document-level term importance scores. These scores are used as base measures for a nonuniform asymmetric Dirichlet distribution prior over terms. As a result, terms can be a priori more or less probable in a topic.

Our topic model has several benefits. The prior knowledge about term importance leads to a more robust topic model that boosts the probability of important terms. As a result, highly related terms to the central theme of the corpus are generated. In addition, our experimental studies show that our topic models significantly outperform LDA and the Bigram topic model in terms of perplexity and coherence. Moreover, incorporating importance vectors as a base measure for our Dirichlet priors yield a higher accuracy in classification tasks. We also found that topic models using document-level term importance measures perform better than corpus-level ones in perplexity and text classification, and measures leveraging Wikipedia produce more coherent topics. We recommend that the topic model using tfidf-Wikipedia-based measure is the best measure to use with our proposed LDA-based models. In addition, our method is an extra module that can be easily incorporated into other topic models.

Chapter 6

News Recommender System

People have always been confronting with a growing amount of data, which in turn demands more on their abilities to filter the content according to their preferences. Among the increasingly overwhelming amounts of webpages, documents, pictures, or videos, it is no longer intuitive to find what we really need. Furthermore, duplicate or several information sources are found covering the same topics. The users are sensitive to the recentness of information and their interests are also changing over time along with the content of the Web [65].

During the past two decades, the concepts of recommender systems have emerged to remedy the situation. The essence of recommender systems are highly associated with the extensive work in cognitive science, approximation theory, information retrieval, forecasting theories, and management science [1]. Recommender systems have many applications, such as product recommendations at Amazon.com [52], movies recommendations by MovieLens [59], and news recommendations [1].

In this chapter, we present an application of topic modeling to news recommender

systems. The reasons we employ topic models in news recommender systems are as follows. Firstly, topic models yield great insight about different themes of a newspaper article. Secondly, topic models capture probabilities of assigning different themes to newspaper articles. Thirdly, topic models provide a generative probabilistic model for the themes. As a consequence, topic models accurately assign probabilities to an unseen document. We focus specifically on the design and development of a news recommender system for *The Globe and Mail*[1]. The Globe and Mail offers most authoritative news in Canada, featuring national and international news. The goal is to design a news recommender system that eases reading and navigation through online newspapers. In essence, the recommender system acts as filters, delivering only news articles that can be considered relevant to a user.

There are in general three types of recommender systems: *Collaborative filtering-based*, *Content-based*, and *Hybrid-based*. Collaborative filtering-based recommender systems make recommendations based on the behavior of other users in the system. Intuitively, these systems assume that if users agree about the quality of some items, then, they will likely agree about other items [36]. For example, if a group of users have similar tastes to Mary, then, Mary is likely to like the things the group likes which she hasn't seen yet. However, in this approach the introduction of new users or new items can cause the *cold start* problem, as there will be insufficient data on these new entries for the collaborative filtering to draw any inferences for new users or items. Addressing the cold start problem can be important for a new user's engagement and is therefore of critical significance in trade applications. The content-based recommender systems make recommendations independent of other users, but

[1]http://www.theglobeandmail.com/

based on items a user likes [1]. This system only considers the properties of items, i.e. the content of news articles, and accordingly makes recommendations. For example, in a news recommender system, if Mary likes sports news, then, the content-based recommender system is likely to recommend articles about sports to her. Therefore, in this approach, introducing new users or items does not cause the cold start problem. Once a new user shows interest to an item, the system keeps recommending other items similar to the user's preferences. The hybrid recommender systems generate recommendations by combining the two aforementioned recommendation techniques. Given the fact that this recommender system contains collaborative filtering-based approaches, it suffers from the cold start problem.

Due to the textual nature of our news application domain and avoiding the cold start problem, we focus on content-based recommender systems. Most existing content-based news recommender systems are based on keywords that is they represent the content of news articles using a set of keywords neglecting the thematic structure of the articles. We apply topic models to discover hidden themes of the news articles, and we incorporate these themes into a content-based recommender system. Our experimental studies show that the proposed recommender system yields more accurate results than other counterparts.

The structure of this chapter is as follows. In Section 6.1, a general introduction of our application domain is explained. In Section 6.2, the related literature is reviewed. Section 6.3 presents our proposed content-based news recommender system. In Section 6.4, we demonstrate the effectiveness of our approach through experiments. Section 6.5 concludes the chapter.

6.1 Introduction

News recommender systems arise to efficiently handle the overwhelming number of news articles, simplify navigations, and retrieve relevant information. Formally, the recommendation problem can be formulated as follows: Let $\mathcal{U}$ be the collection of $|\mathcal{U}|$ users, represented by $\mathcal{U} = \{u_1, u_2, \cdots, u_{|\mathcal{U}|}\}$, and let $\mathcal{C} = \mathcal{D} \cup \mathcal{Q}$ represent all the news articles, where $\mathcal{D}$, denoted by $\mathcal{D} = \{d_1, d_2, \cdots, d_M\}$, is the collection of *read articles* that is all news articles that have been read by at least one user, and $\mathcal{Q}$, denoted by $\mathcal{Q} = \{q_1, q_2, \cdots, q_N\}$, is the collection of *non-read articles* that is all the latest articles published daily that have not yet been read and are to be recommended[2].

Let f be a utility function that measures the usefulness of a news article $c \in \mathcal{C}$ to a user $u_l \in \mathcal{U}$, i.e., $f : \mathcal{U} \times \mathcal{C} \rightarrow R$, where R is a totally ordered set (e.g., non-negative integers or real numbers within a certain range). Then, for each user $u_l \in \mathcal{U}$, we want to choose such news article $c^{'} \in \mathcal{C}$ that maximizes the user's utility. More formally:

$$\forall u_l \in \mathcal{U}, c^{'}_{u_l} = argmax_{c \in \mathcal{C}} f(u_l, c). \tag{6.1}$$

In recommender systems, the sets $\mathcal{U}$ and $\mathcal{C}$ are usually defined by several characteristics [1]. Similarly, in our work, each user $u_l \in \mathcal{U}$ is defined by a unique identifier, such as user ID. Each article in the collection $\mathcal{C}$ is defined by a unique article identifier and article content. In addition, we represent the utility of a news article by the amount of time a user spends on the article, which indicates the interestingness of the news article to the user. For example, user u_0 spent two minutes (out of five

[2]Note that our news recommender system is capable of personalizing the collection of non-read articles ($\mathcal{Q}$) for each user.

minutes[3]) on the news article *"d_0: SpaceX launches fifth official mission"*.

In our recommender system, the amount of time spent on the collection of non-read articles ($\mathcal{Q}$) is not available. Thus, the fundamental issue of our recommender system is that the utility function f is not defined on the whole $\mathcal{U} \times \mathcal{C}$ space, but only on $\mathcal{U} \times \mathcal{D}$ space. This means f needs to be extrapolated to the space $\mathcal{U} \times \mathcal{Q}$. Therefore, the goal of our news recommender system is to estimate the time each user would spend on the non-read news articles and issue appropriate recommendations based on these estimates.

In this chapter, we propose a content-based news recommender system by employing LDA-based approaches to measure the similarity between read news articles and non-read news articles. LDA-based approaches elicit a topic model from the collection of news articles. The topic model represents news articles as a multinomial distribution over topics, where each topic is a multinomial distribution over words. Then, given the time a user has spent on read news articles, and the topic models of the collection of news articles, a user's time spent toward non-read news articles is estimated.

6.2 Related Work

The main objective of a news recommender system is to estimate a utility function f that best predicts users' interests in the latest published articles. The estimates are obtained using different methods from machine learning, approximation theory, and

[3]In order to avoid idle time spent on a news article, we normalize the time by scaling between zero and five.

various heuristics [1].

All of the known recommender techniques have strengths and weaknesses. In this section we briefly survey the different recommender techniques, the data that they support, and the algorithms they employ [18, 21]. On this basis, the following three recommender techniques are distinguished:

- *Collaborative filtering-based recommender systems* predict the utility of items based on the behavior of other users in the system [1]. For example, in a news recommender application domain, in order to recommend news articles to a user x, the collaborative filtering-based recommender system tries to find similar users to user x, i.e., other users that have similar tastes in news (rate the same news similarly). Then, only the news articles that are most liked by similar users to user x will be recommended. The greatest strength of this approach is that it considers users' information, i.e. similar users' tastes. However, in the personalized recommender systems, the introduction of new users or new items can cause the *cold start* problem, as there will be insufficient data on these new entries for the collaborative filtering to draw any inferences for new users or items. In collaborative filtering-based recommender systems, a new item cannot be recommended until some users rate it, also referred to as the new item cold start problem. The system requires a substantial number of users to show interest to a new item before that item can be recommended [21]. Moreover, new users are unlikely given good recommendations because of the lack of their activities or interest history, the system is unable to find similar users to a new user. This problem is often referred to as the new user cold start

problem [17].

- *Content-based recommender systems* recommend items similar to items a user preferred in the past [1]. For example, a content-based news recommender system observes the collection of news articles a user prefers and reads frequently. Then, only the news articles that have a high degree of similarity to the user's read articles are recommended. The greatest strength of this approach is that it only considers the properties of an item, i.e. the content of news articles, and accordingly makes recommendations. Therefore, in this approach, once a new user is introduced to the system, as soon as they read their first article, the content-based recommender system starts by recommending articles similar to the read article. Thus, this approach does not cause the cold start problem mentioned in collaborative recommender systems. The weakness of this approach is that users are limited to being recommended news articles that are similar to their read history.

- *Hybrid recommender systems* generate recommendations by combining the above two recommendation techniques, thus, maximizing the benefits and minimizing the disadvantages of them [1]. For example, a hybrid recommendation system that combines content-based and collaborative recommendation systems considers both the content of news articles and a user's demographic information to issue recommendations. Given the fact that this approach contains collaborative recommender systems, it contains the disadvantages of such systems. Therefore, this approach also suffers from the cold start problem.

Due to the textual nature of our news recommendation domain and avoiding

the cold start problem, our proposed recommender system adopts a content-based approach that considers the content of news articles and accordingly issues recommendations.

6.3 The content-based recommender system

Our content-based recommender system employs probabilistic topic models to uncover the thematic similarity between news articles and a user's preferences. Then, news articles that have a high degree of similarity to the user's preferences are recommended.

We assume a collection of users is represented by $\mathcal{U} = \{u_0, u_1, \cdots, u_{|\mathcal{U}|}\}$. Let the corpus of news articles be $\mathcal{C} = \mathcal{D} \cup \mathcal{Q}$, where $\mathcal{D} = \{d_1, d_2, \cdots, d_M\}$ is the collection of read articles, and $\mathcal{Q} = \{q_1, q_2, \cdots, q_N\}$ is the collection of non-read articles. We define a read article $d_i \in \mathcal{D}$ as a tuple of textual content and a subset of readers. That is $d_i =< t_i, U_i >$, where t_i is the textual content, represented by a sequence of terms of the article and $U_i \subset \mathcal{U}$ is a subset of users associated with the article. Similarly, a non-read article $q_j \in \mathcal{Q}$ is defined by $q_j =< t_j, \emptyset >$, where the set of readers is empty.

Our task is to appropriately recommend non-read articles to users or alternatively to assign users to non-read articles. In other words, for each non-read article $q_j =< t_j, \emptyset >$, we plan to predict the most appropriate subset of users and replace it with the empty set ($\emptyset$).

The proposed content-based news recommender system consists of the following three steps.

6.3.1 Step 1: Building a topic model

In this step, we use LDA-based topic models to best reflect the thematic structure of news articles. We build a topic model from the collection of read articles ($\mathcal{D}$). Our topic model assumes that each news article $d_i \in \mathcal{D}$ has a multinomial distribution over K topics with parameters $\vec{\Theta}_{d_i}$. As a result of this step, we obtain $\underline{\Theta_{\mathcal{D}}}$ that is an $M \times K$ array of topic probabilities given read articles, where M is the total number of read articles and K is the total number of topics.

6.3.2 Step 2: Inference and learning

We use the topic model, built in Step 1, to infer the multinomial distribution of each non-read article ($q_j \in \mathcal{Q}$) over K topics with parameters $\vec{\Theta}_{q_j}$. As a result of this step, we obtain $\underline{\Theta_{\mathcal{Q}}}$ that is an $N \times K$ array of topic probabilities given non-read articles, where N is the total number of non-read articles and K is the total number of topics.

6.3.3 Step 3: Making recommendations

For each user $u_l \in \mathcal{U}$, we obtain their collection of read articles $D_{u_l} \subset \mathcal{D}$ and their respective topic vectors $\underline{\Theta_{D_{u_l}}}$. Given a collection of non-read articles $\mathcal{Q}$, and their topic vectors $\underline{\Theta_{\mathcal{Q}}}$, our proposed method outputs a ranked list $Q_y^{u_l} = \{q_0, q_1, \cdots, q_y\}$, where $q_r \in \mathcal{Q}$, of y non-read articles interesting to a user u_l.

The probability of article q_r being interesting to user u_l is computed for each $q_r \in \mathcal{Q}$ as

$$p(q_r|u_l, \mathcal{Q}, D_{u_l}) = \frac{InterestingnessScore(q_r, u_l, D_{u_l})}{\sum_{q_j \in \mathcal{Q}} InterestingnessScore(q_j, u_l, D_{u_l})}, \tag{6.2}$$

$$InterestingnessScore(q_r, u_l, D_{u_l}) = \sum_{d_i \in D_{u_l}} DocSim(q_r, d_i, D_{u_l}) \cdot timeSpent[u_l, d_i]. \tag{6.3}$$

$InterestingnessScore(q_r, u_l, D_{u_l})$ calculates how interesting article q_r is to user u_l. This score can be any real non-negative number. $DocSim(q_r, d_i, D_{u_l})$ measures the similarity between two articles, i.e. q_r and d_i, given a collection of read articles by user u_l (D_{u_l}) and returns a similarity measure ranging between $[0, 1]$, and $timeSpent[u_l, d_i]$ is the amount of time user u_l spends on article d_i.

We apply LDA-based approaches to compute the article similarity. We utilize two arrays $\vec{\Theta}_{q_r}$ and $\vec{\Theta}_{d_i}$, obtained from Steps 1 and 2, to determine the similarity between q_r and d_i. Arrays $\vec{\Theta_{q_r}}$ and $\vec{\Theta_{d_i}}$ represent the latent topic distribution of articles q_r and d_i. Thus, inspired from Chang *et al.* [24], we view each article as a topic-based vector and use *cosine-based* similarity measure to compute the similarity between a read and a non-read article. Note that our experimental studies show similar results for other similarity measure approaches, such as Manhattan distance. A comprehensive survey on similarity measures between vectors can be found at [22].

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. The more similar hence the more co-oriented the vectors, thus the cosine of the angle between them is closer to one. Cosine similarity measure is often used to compare documents for text mining, classification, and clustering purposes [22]. Equation 6.4 is used to calculate the similarity.

$$cosine - similarity(\vec{\Theta}_{q_r}, \vec{\Theta}_{d_i}) = \frac{\vec{\Theta}_{q_r} \cdot \vec{\Theta}_{d_i}}{|\vec{\Theta}_{q_r}| \times |\vec{\Theta}_{d_i}|}, \tag{6.4}$$

where "$\cdot$" denotes the inner product of two vectors, and $|\vec{x}|$ represents the size of the vector.

Finally, we return top y articles ranked by the $p(q_r|u_l, \mathcal{Q}, D_{u_l})$ probability.

6.4 Experiments

We conducted experiments on *The Globe and Mail* news article corpus. The Globe and Mail collection appeared on The Globe and Mail newswire during the period between January 2010 to March 2014. The articles were assembled and indexed with article IDs by personnel from The Globe and Mail. The Globe and Mail corpus contains $142,163,909$ news articles. Moreover, the collection contains $10,150$ subscribed users that have spent some time, i.e. any real non-negative number between one and five, on each article. In order to avoid idle time spent on a news article, we normalize the time by scaling between zero and five. The news articles are divided into $142,163,000$ read articles that are read by at least one reader and 909 non-read articles that are recently published.

We compare the performance of our proposed content-based recommender system against baseline recommendation systems that solely use bag-of-words tfidf representation of news articles. The following topic models are used in our experiments: LDA [16], the Bigram Topic Model [80], TMCTI and TMDTI, explained in Chapter 5. The topic models were trained with 1000 iterations of Gibbs sampling [39, 40] used in the MALLET [57]. Initial values for the hyperparameters α and β applied to all our

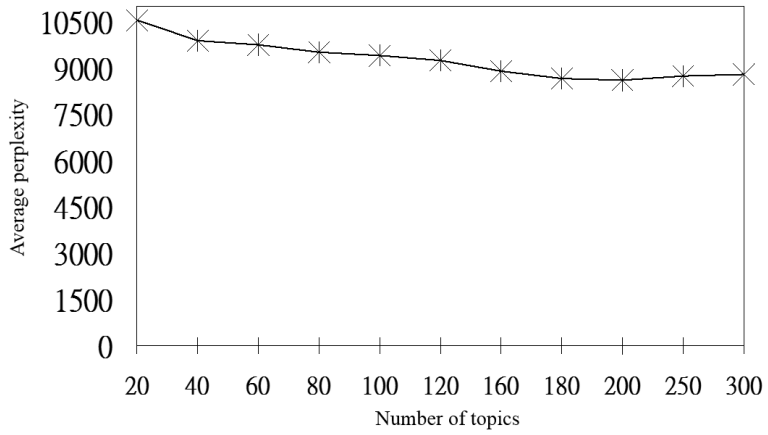

Figure 6.1: Average perplexity as a function of number of topics, using LDA, TMCTI Wikipedia-based, TMCTI idf-Wikipedia-based measure, TMDTI tfidf-based measure, and TMDTI tfidf-Wikipedia-based on The Globe and Mail corpus.

experiments are $\alpha = 50.0/K$ and $\beta = 0.01$. Note that these parameters are default parameters of most LDA-based topic models, expected to result in a fine-grained decomposition of the corpus into topics [40].

6.4.1 Number of topics

An open question in topic modeling is how to set the number of topics K. Several approaches exist, but ultimately, the appropriate number of topics must depend on both the corpus itself and user modeling goals [78, 40].

The optimum number of topics is expected to result in a fine-grained decomposition of the corpus into topics [40], where topic distributions over words are of minimum similarity. Furthermore, the optimum number of topics leads to a low cross-entropy between the term distribution learned by the topic model and the distribution of terms in an unseen test article. Thus, the optimum number of topics results in a

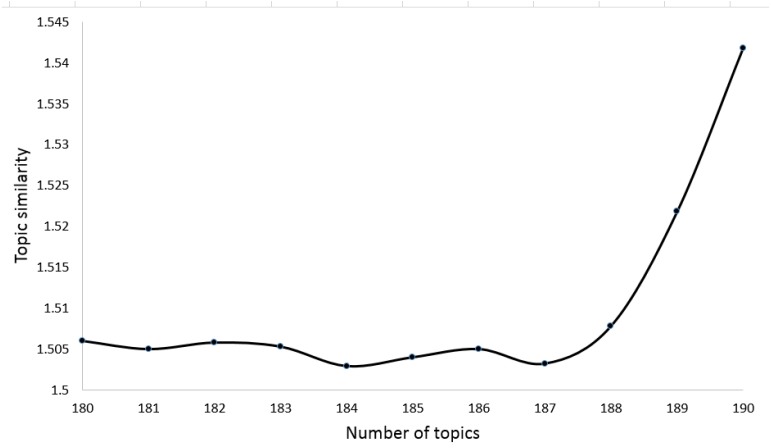

Figure 6.2: Similarity of topic distributions over words, as a function of number of topics, using LDA on The Globe and Mail corpus.

lower perplexity score indicating that the model is better in predicting distribution of the test article [16].

In our experiments, we learn topics for different values of K and choose the value which minimizes the perplexity score. The experiments are conducted using different topic models for different number of topics K, where $K = 20 \cdots K = 300$. Figure 6.1 illustrates the average perplexity as a function of number of K. In this figure, the values of $K \in [180 \cdots 190]$ achieve the best performance in terms of perplexity.

As mentioned earlier, a topic model generates K topics, where each topic is a distribution over V words, denoted by $\vec{\Phi}_k = \{w_1, w_2, \cdots, w_V\}$. Similarity between topics is the similarity of topic distributions over words across different topics. We calculate the normalized average sum of similarity scores between every pair of K topics ($K \in [180 \cdots 190]$), generated from The Globe and Mail corpus. As illustrated in Figure 6.2, $K = 187$ results in the most fine-grained decomposition of the corpus into topics with the minimum similarity between topic-word distributions.

6.4.2 Evaluation of the recommender system

In this section, we evaluate the performance of our proposed content-based news recommender system using the following metrics: *precision*, *recall*, and *F-measure*.

Precision, recall, and F-measure are well-known evaluation metrics in information retrieval literature [55]. For each user, we use the original set of read articles as the ground truth T_g. Assume that the set of recommended news articles are T_r, so that the correctly recommended articles are $T_g \cap T_r$. Precision, recall, and F-measure are defined as follows:

$$precision = \frac{|T_g \cap T_r|}{|T_r|}, \tag{6.5}$$

$$recall = \frac{|T_g \cap T_r|}{|T_g|}, \tag{6.6}$$

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}. \tag{6.7}$$

In our experiments, the number of recommended articles ranges from 1 to 30. Figures 6.3, 6.4, and 6.5 illustrate the precision, recall, and F-measure of the proposed recommender system as a function of number of recommended articles.

Empirical comparisons show that using topic models to represent articles improves the precision, recall, and F-measure of our proposed recommender system. Since the only difference between the comparisons is the article similarity function $DocSim(q_r, d_i, D_{u_l})$, which compares the similarity between a new non-read article q_r and a read article d_i, analyzing the differences between the two article similarity measures provides explanation about the performance difference.

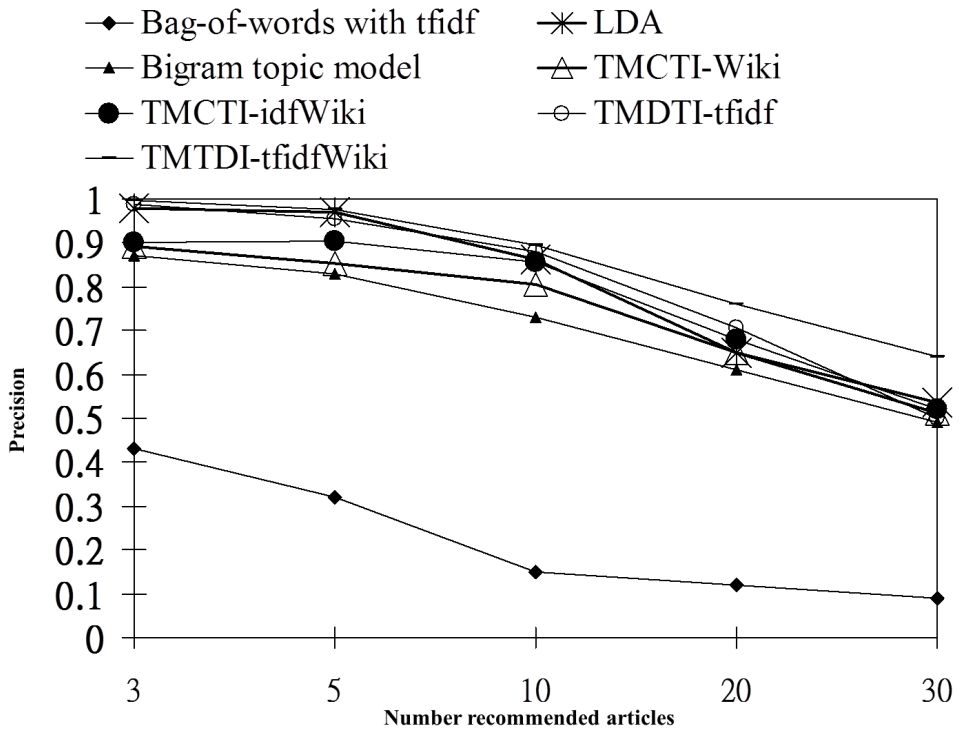

Figure 6.3: Precision of the proposed recommender system as a function of number of recommended articles, using the following article representation methods: bag-of-words with tfidf, LDA, the bigram topic model, TMCTI Wikipedia-based, TMCTI idf-Wikipedia-based measure, TMDTI tfidf-based measure, and TMDTI tfidf-Wikipedia-based on The Globe and Mail corpus.

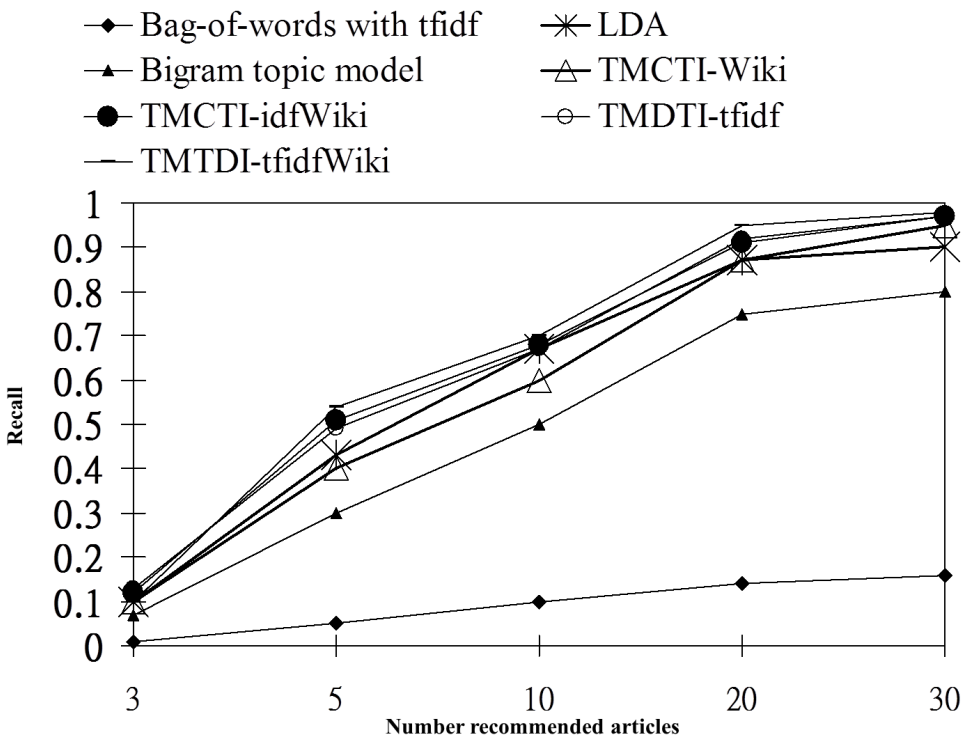

Figure 6.4: Recall of the proposed recommender system as a function of number of recommended articles, using the following article representation methods: bag-of-words with tfidf, LDA, the bigram topic model, TMCTI Wikipedia-based, TMCTI idf-Wikipedia-based measure, TMDTI tfidf-based measure, and TMDTI tfidf-Wikipedia-based on The Globe and Mail corpus.

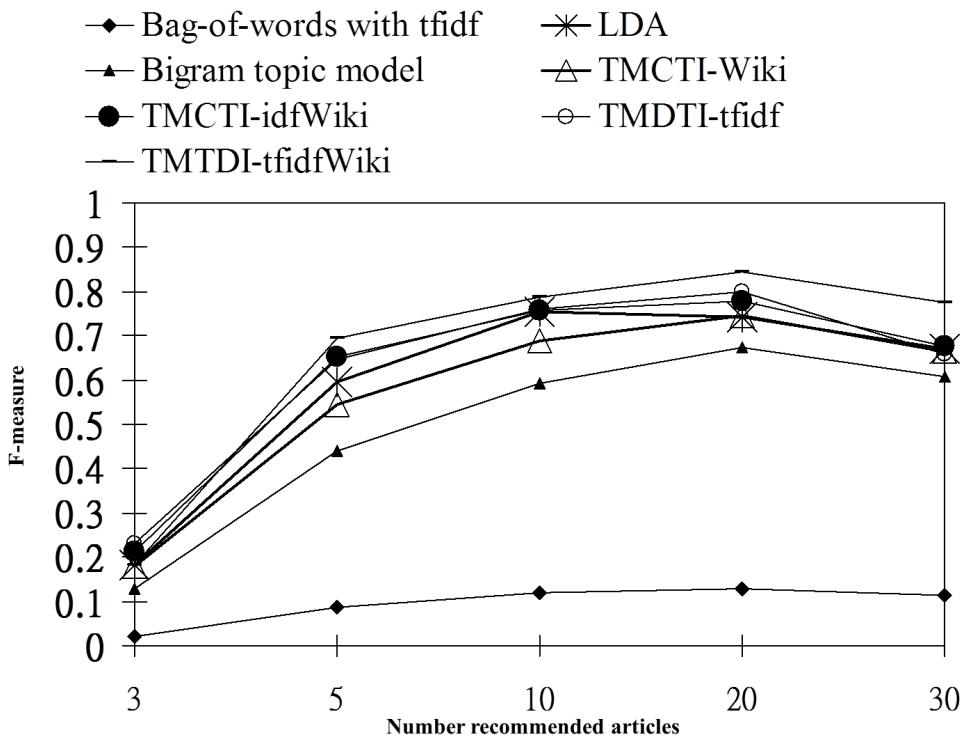

Figure 6.5: F-measure of the proposed recommender system as a function of number of recommended articles, using the following article representation methods: bag-of-words with tfidf, LDA, the bigram topic model, TMCTI Wikipedia-based, TMCTI idf-Wikipedia-based measure, TMDTI tfidf-based measure, and TMDTI tfidf-Wikipedia-based on The Globe and Mail corpus.

The bag-of-words with tfidf approach represents two articles by tfidf vectors. Then, the cosine similarity between these vectors are computed and used in the recommendation system. Generally speaking, the tfidf article similarity measures the quantity of term overlap, where each term has a different weight, in the two articles [79]. This approach ignores the thematic structures of articles to perform the similarity measure.

The LDA-based approaches first generate a set of topic vectors for the articles, each of which is represented by a distribution over terms. Terms in each topic are semantically coherent. Then, LDA-based recommender systems measure the cosine similarity between the topic vectors. Generally speaking, using LDA-based topic vectors quantifies the topic similarity between the two articles. Moreover, incorporating term importance into topic models yields a higher precision, recall, and F-measure than when using solely LDA-based topic vectors. Key to this improvement is incorporating term importance as a nonuniform base measure into the asymmetric prior over topic-term distributions. This leads to better estimates for less frequent important terms and consequently, more coherent representation of the multinomial distribution over topics, and thus, better quantifies the topic similarity between the two articles.

Hence we recommend using TMTDI-tfidfWiki topic model to represent articles for content-based news recommender systems.

6.5 Summary

This chapter presents a content-based recommender system for The Globe and Mail, a company that offers most authoritative news in Canada, featuring national and

international news. One of the important problems of The Globe and Mail newswire is the growing amount of articles, which in turn demands a system to automatically filter and deliver the content according to readers' preferences. Furthermore, in the current collaborative-based recommender system at The Globe and Mail, the introduction of new users or new news articles can cause the cold start problem, as there will be insufficient data on these new entries for the collaborative filtering to work accurately.

We propose to utilize the latent Dirichlet allocation (LDA) model to discover hidden themes of the news articles. We incorporate these themes into a content-based recommender system. Our experimental studies show that the proposed recommendation system yields better results than solely bag-of-words with tfidf presentation. Moreover, given the fact that our recommender system only considers the content of news articles to make recommendations, introducing a new user or a new news article does not cause the cold start problem.

Chapter 7

Conclusion and Future Work

Topic modelling is a powerful statistical tool to uncover hidden thematic structures and multi-faceted summaries of documents or other discrete data. Most topic models, such as Latent Dirichlet Allocation (LDA), consider documents to be a weighted mixture of topics, where each topic is a multinomial distribution over terms. The inferred topic model assigns a high probability to the topics of a corpus. In addition, the highest probable terms in each topic provide important terms that summarize the themes of the corpus.

The bag-of-words representation of text documents is of particular interest in most topic models. However, this representation does not contain information about the underlying structure of text documents. The goal of many topic modelling applications is to better discover the hidden thematic structure of a dataset, and this requirement is not always adequately addressed by the standard unsupervised machine learning setting. Incorporating additional knowledge about the dataset or statistics of an external data source into topic modelling applications allows us to explore or

better understand a dataset.

7.1 Our approaches

In this dissertation, we proposed three extended LDA models that incorporates syntactic and semantic structures of text documents into probabilistic topic models.

Our first proposed topic model, the HPSG-based topic model, enriches text documents with collapsed typed dependency relations to effectively acquire syntactic and semantic dependencies between consecutive and nonconsecutive words of text documents. This representation has several benefits. It captures relations between consecutive and nonconsecutive words of text documents. In addition, the labels of the collapsed typed dependency relations help to eliminate less important relations, i.e., relations involving prepositions. Furthermore, our experimental studies show that the proposed topic model significantly outperforms LDA and is also better than the Bigram Topic Model in terms of perplexity. We also show that our model achieves comparable results with other models in terms of stability, coherence, and accuracy. Besides, the results from our topic model have less ambiguity, given the fact the generated terms include pairs of words that are more descriptive than single words.

Our second and third proposed topic models do not use a symmetric distribution prior over terms but rather adjust the prior by employing additional information about the importance of terms in a topic. The importance of terms in a topic is captured by corpus-level (TMCTI) and document-level (TMDTI) term importance scores. These scores are used as base measures for a nonuniform asymmetric Dirichlet distribution prior over terms. As a result, terms can be a priori more or less probable

in a topic. Our topic models have several benefits. The prior knowledge about term importance leads to a more robust topic model that boosts the probability of important terms. As a result, highly related terms to the central theme of the corpus are generated. In addition, our method is an extra module that can be easily incorporated into other topic models. Furthermore, our experimental studies show that our topic models significantly outperform LDA and the Bigram Topic Model in terms of perplexity and coherence. We also found that topic models using document-level term importance measures perform better than corpus-level ones in perplexity, and measures leveraging Wikipedia produce more coherent topics. We recommend that the topic model using tfidf-Wikipedia-based measure is the best measure to use with our proposed extended LDA models.

Furthermore, we extend the HPSG-based topic model to include term importance. Typed dependency relations of text documents are extracted by employing syntax and semantic analysis. We further assign weights to those relations using the context of the corpus or an external data source. Then, these weights are incorporated into the HPSG-based topic model to increase the probability of important relations and to consequently decrease the probability of less important relations. Experimental studies show the effectiveness of our method.

Moreover, in this thesis, we introduced a method to enforce topic similarity to conceptually similar words. As a result, this algorithm led to more coherent topic distribution over words.

In addition, we applied our topic models in a content-based recommendation system for The Globe and Mail to ease reading and navigation through online newspaper articles. The proposed recommender system yields better results than the ones us-

ing bag-of-words methods for representing documents. Moreover, our recommender system does not suffer from the cold start problem.

7.2 Future directions

While the topic models presented in this thesis represent significant advances in probabilistic topic modelling, there are still many interesting opportunities for further improvement.

The inclusion of term importance is a powerful tool in topic modelling. While employing document level or corpus level term importance measures is very useful to estimate term importance, these measures do not take users' feedback into consideration. In order to make best predictions about term importance, it may be advantageous to leverage users' feedback.

The combination of collapsed typed dependency relations and topic modelling provides interesting directions for future work. The definition of syntactic typed dependency relations could allow the incorporation of sentiment similarity of terms of the corpus into typed dependency relations. We could eliminate relations that include terms that are not sentimentally related. This elimination could lead to more coherent topic word distributions.

Applying topic models in a content-based recommender system yields more accurate results than other recommender systems. However, our content-based recommender system must effectively evolve with its content. In our current system, the topic model needs to be generated offline. For instance, once non-read news articles enter the collection of read articles, the topic model needs to be updated to reflect

the themes of new articles. This offline generation of a topic model is a drawback, as it hinders the system's ability to evolve quickly. We could develop a real-time content-based recommender system, that leverages a stream of news articles and is capable of handling online LDA [44].

Bibliography

[1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transaction on Knowledge and Data Engineering*, 17(6):734–749, June 2005.

[2] D. Andrzejewski, X. Zhu, and M. Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceeding of the 26th Annual International Conference on Machin Learning*, ICML '09, pages 25–32, 2009.

[3] D. M. Andrzejewski. *Incorporating Domain Knowledge in Latent Topic Models*. PhD thesis, University of Wisconsin-Madison, USA, 2010.

[4] C. Apté, F. Damerau, and S. M. Weiss. Automated learning of decision rules for text categorization. *ACM Transaction Information System*, 12(3):233–251, July 1994.

[5] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 27–34, Arlington, Virginia, United States, 2009. AUAI Press.

[6] R. Balabantaray, D. Sahoo, B. Sahoo, and M. Swain. Text summarization using term weights. *International Journal of Computer Applications*, 38(1):10–14, 2012.

[7] M. Bendersky, D. Metzler, and W. B. Croft. Learning concept importance using a weighted dependence model. In *Proceeding of the Third ACM International. Conference on Web Search and Data Mining*, WSDM '10, pages 31–40, New York, NY, USA, 2010.

[8] I. Bhattacharya and L. Getoor. A latent dirichlet model for unsupervised entity resolution. In *SIAM International Conference on Data Mining*, 2006.

[9] B. Bigi. Using kullback-leibler distance for text categorization. In F. Sebastiani, editor, *Advances in Information Retrieval*, volume 2633 of *Lecture Notes in Computer Science*, pages 305–319. Springer Berlin Heidelberg, 2003.

[10] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[11] D. M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, Apr. 2012.

[12] D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *ACM*, 57(2):7:1–7:30, Feb. 2010.

[13] D. M. Blei and M. I. Jordan. Modeling annotated data. In *Proceedings of the 26th annual international ACM SIGIR Conference on Research and development*

in informaion retrieval, SIGIR '03, pages 127–134, New York, NY, USA, 2003. ACM.

[14] D. M. Blei and J. D. Lafferty. Correlated topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120. MIT Press, 2006.

[15] D. M. Blei and J. D. Mcauliffe. Supervised topic models. MIT Press, 2008.

[16] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[17] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal. A collaborative filtering approach to mitigate the new user cold start problem. *Knowledge-Based System*, 26:225–238, Feb. 2012.

[18] H. Borges and A. Lorena. A survey on recommender systems for news data. In E. Szczerbicki and N. Nguyen, editors, *Smart Information and Knowledge Management*, volume 260 of *Studies in Computational Intelligence*, pages 129–151. Springer Berlin Heidelberg, 2010.

[19] C. Boulis and M. Ostendorf. Text classification by augmenting the bag-of-words representation with redundancy compensated bigrams. In *Proceeding of the International Workshop in Feature Selection in Data Mining*, pages 9–16. Citeseer, 2005.

[20] J. L. Boyd-Graber and D. M. Blei. Syntactic topic models. *CoRR*, abs/1002.4665, 2010.

[21] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, Nov. 2002.

[22] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *International Journal of Mathematical Models and Methods in Applied Sciences*, 1(4):300–307, 2007.

[23] Y. Cha and J. Cho. Social-network analysis using topic models. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 565–574, New York, NY, USA, 2012. ACM.

[24] T.-M. Chang and W.-F. Hsiao. Lda-based personalized document recommendation. In *Proceedings of the PACIS*, 2013.

[25] C. Chemudugunta, A. Holloway, P. Smyth, and M. Steyvers. Modeling documents by combining semantic concepts with unsupervised statistical learning. In A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, editors, *The Semantic Web - ISWC 2008*, volume 5318 of *Lecture Notes in Computer Science*, pages 229–244. Springer Berlin Heidelberg, 2008.

[26] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, Mar. 1990.

[27] W. M. Darling and F. Song. Probabilistic topic and syntax modeling with part-of-speech lda. *CoRR*, abs/1303.2826, 2013.

[28] H. Daumé. Markov random topic fields. pages 293–296, 2009.

[29] M.-C. de Marnee and C. D. Manning. Stanford typed dependencies manual. 2012.

[30] M.-C. de Marneffe, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceeding International Conference on language resource and evaluation LREC*, pages 449–454, 2006.

[31] F. Debole and F. Sebastiani. An analysis of the relative hardness of reuters-21578 subsets: Research articles. *Journal of American Society of Information Science and Technology*, 56(6):584–596, Apr. 2005.

[32] E. Delpisheh and A. An. Topic modeling using collapsed typed dependency relations. In *Advances in Knowledge Discovery and Data Mining PAKDD'14*, pages 146–161. 2014.

[33] E. Delpisheh, A. An, and A. Agrawal. Topic modeling using term importance. 2015.

[34] G. Doyle and C. Elkan. Accounting for burstiness in topic models. In *Proceeding of the 26th Annual International. Conference on Machin Learning*, ICML '09, pages 281–288, 2009.

[35] V. Eidelman, J. Boyd-Graber, and P. Resnik. Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages

115–119, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[36] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. Collaborative filtering recommender systems. *Journal of Foundations and Trends in Human-Computer Interaction*, 4(2):81–173, 2011.

[37] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceeding of the 20th International. Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1606–1611, 2007.

[38] Y. Gao, Y. Xu, Y. Li, and B. Liu. A two-stage approach for generating topic models. In J. Pei, V. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, volume 7819 of *Lecture Notes in Computer Science*, pages 221–232. Springer Berlin Heidelberg, 2013.

[39] T. Griffiths. Gibbs sampling in the generative model of latent dirichlet allocation. *Standford University*, 518(11):1–3, 2002.

[40] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceeding of the National Academy of Sciences of the United States of America*, 101:5228–5235, 2004.

[41] T. L. Griffiths, M. Steyvers, D. M. Blei, and J. B. Tenenbaum. Integrating topics and syntax. In *In Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press, 2005.

[42] A. Gruber, M. Rosen-zvi, and Y. Weiss. Hidden topic markov models. In *Proceedings of Artificial Intelligence and Statistics*, 2007.

[43] G. Heinrich. Parameter estimation for text analysis. *Web: http://www. arbylon. net/publications/text-est. pdf*, 2005.

[44] M. D. Hoffman, D. M. Blei, and F. R. Bach. Online learning for latent dirichlet allocation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *NIPS*, pages 856–864. Curran Associates, Inc., 2010.

[45] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 50–57, New York, NY, USA, 1999. ACM.

[46] H. Jin, L. Zhang, and L. Du. Semantic title evaluation and recommendation based on topic models. In J. Pei, V. Tseng, L. Cao, H. Motoda, and G. Xu, editors, *Advances in Knowledge Discovery and Data Mining*, volume 7819 of *Lecture Notes in Computer Science*, pages 402–413. Springer Berlin Heidelberg, 2013.

[47] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2000.

[48] R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09, pages 61–68, New York, NY, USA, 2009. ACM.

[49] H. J. Lau, D. Newman, and T. Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539. Association for Computational Linguistics, 2014.

[50] R. D. Levine and W. D. Meurers. Head-driven phrase structure grammar: Linguistic approach, formal foundations, and computational realization. Elsevier, Oxford, 2006.

[51] K. Li, J. Xie, X. Sun, Y. Ma, and H. Bai. Multi-class text categorization based on LDA and SVM. *Procedia Engineering*, 15(0):1963 – 1967, 2011. CEIS 2011.

[52] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, Jan. 2003.

[53] X. Luo, H. Raghavan, V. Castelli, S. Maskey, and R. Florian. Finding what matters in questions. In *proceedings of NAACL-HLT*, 2013.

[54] R. E. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the dirichlet distribution. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 545–552, New York, NY, USA, 2005. ACM.

[55] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[56] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *Journal Artificial Intelligence Research*, 30(1):249–272, Oct. 2007.

[57] A. K. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[58] Q. Mei, X. Shen, and C. Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD international Conference on Knowledge discovery and data mining*, KDD '07, pages 490–499, New York, NY, USA, 2007. ACM.

[59] B. N. Miller, I. Albert, S. K. Lam, J. A. Konstan, and J. Riedl. Movielens unplugged: Experiences with an occasionally connected recommender system. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 263–266, New York, NY, USA, 2003. ACM.

[60] G. A. Miller. Wordnet: a lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.

[61] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD international Conference on Knowledge discovery and data mining*, KDD '07, pages 500–509, New York, NY, USA, 2007. ACM.

[62] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth Conference on Uncertainty in artificial intelligence*, UAI'02, pages 352–359, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[63] C. Musat, J. Velcin, M.-A. Rizoiu, and S. Trausan-Matu. Concept-based topic model improvement. In D. Ryzko, H. Rybinski, P. Gawrysiak, and

M. Kryszkiewicz, editors, *Emerging Intelligent Technologies in Industry*, volume 369 of *Studies in Computational Intelligence*, pages 133–142. Springer Berlin Heidelberg, 2011.

[64] C. Musat, J. Velcin, M.-A. Rizoiu, and S. Trausan-Matu. Improving topic models using conceptual data. 2011.

[65] D. Z. Mria Bielikov, Michal Kompan. Effective hierarchical vector-based news representation for personalized recommendation. *Computer Science and Information Systems*, (21):303–322, 2012.

[66] D. Newman, J. H. Lau, K. Grieser, and T. Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 100–108, 2010.

[67] K. Niemann and M. Wolpers. A new collaborative filtering approach for increasing the aggregate diversity of recommender systems. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 955–963, New York, NY, USA, 2013. ACM.

[68] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machin Learning*, 39(2-3):103–134, May 2000.

[69] R. Parimi and D. Caragea. Predicting friendship links in social networks using a topic modeling approach. In J. Huang, L. Cao, and J. Srivastava, editors,

Advances in Knowledge Discovery and Data Mining, volume 6635 of *Lecture Notes in Computer Science*, pages 75–86. Springer Berlin Heidelberg, 2011.

[70] J. Petterson, A. J. Smola, T. S. Caetano, W. L. Buntine, and S. M. Narayanamurthy. Word features for latent dirichlet allocation. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *NIPS*, pages 1921–1929. Curran Associates, Inc., 2010.

[71] C. Pollard and I. A. Sag. *Information-based syntax and semantics: Vol. 1: fundamentals*. Center for the Study of Language and Information, Stanford, CA, USA, 1988.

[72] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 569–577. ACM, 2008.

[73] J. Reed, Y. Jiao, T. Potok, B. Klump, M. Elmore, and A. Hurson. Tf-icf: A new term weighting scheme for clustering dynamic data streams. In *machine Learning and Applications, 2006. ICMLA '06. 5th International Conference on*, pages 258–263, Dec 2006.

[74] M. Rosen-Zvi, C. Chemudugunta, T. Griffiths, P. Smyth, and M. Steyvers. Learning author-topic models from text corpora. *ACM Transaction Information System*, 28(1):4:1–4:38, Jan. 2010.

[75] R. Salakhutdinov and G. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969 – 978, 2009. Special Section on Graphical Models and Information Retrieval.

[76] M. Steyvers and T. L. Griffiths. Rational analysis as a link between human memory and information retrieval. In N. Chater and M. Oaksford, editors, *The Probabilistic Mind: Prospects for Bayesian Cognitive Science*, pages 329–349. Oup Oxford, 2008.

[77] H. Su, J. Tang, and W. Hong. Learning to diversify expert finding with subtopics. In P.-N. Tan, S. Chawla, C. Ho, and J. Bailey, editors, *Advances in Knowledge Discovery and Data Mining*, volume 7301 of *Lecture Notes in Computer Science*, pages 330–341. Springer Berlin Heidelberg, 2012.

[78] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[79] S. Tuarob, L. C. Pouchard, and C. L. Giles. Automatic tag recommendation for metadata annotation using probabilistic topic modeling. In *Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '13, pages 239–248, New York, NY, USA, 2013. ACM.

[80] H. M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning ICML '06*, pages 977–984, New York, NY, USA, 2006. ACM.

[81] H. M. Wallach. *Structured Topic Models for Language*. PhD thesis, University of Cambridge, 2008.

[82] H. M. Wallach, D. Mimno, and A. McCallum. RethinkingRethinking LDA: Why Priors Matter. In *Proceedings of NIPS*, 2009.

[83] H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1105–1112, New York, NY, USA, 2009. ACM.

[84] X. Wang and E. Grimson. Spatial latent dirichlet allocation. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1577–1584. Curran Associates, Inc., 2008.

[85] X. Wang, A. McCallum, and X. Wei. Topical n-grams: Phrase and topic discovery, with an application to information retrieval. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 697–702, Washington, DC, USA, 2007. IEEE Computer Society.

[86] Y. Wang, P. Sabzmeydani, and G. Mori. Semi-latent dirichlet allocation: A hierarchical model for human action recognition. In *Proceedings of the 2Nd Conference on Human Motion: Understanding, Modeling, Capture and Animation*, pages 240–254, Berlin, Heidelberg, 2007. Springer-Verlag.

[87] A. T. Wilson and P. A. Chew. Term weighting schemes for latent dirichlet allocation. In *Human Language Technologies: The 2010 Annual Conference of*

the North American Chapter of the Association for Computational Linguistics, HLT '10, pages 465–473, 2010.

[88] H. Xiao and T. Stibor. Efficient collapsed gibbs sampling for latent dirichlet allocation. In *Asian Conference on Machine Learning (ACML)*, volume 13 of *JMLR W and CP*, 2010.

[89] D. Xiong, M. Zhang, and X. Wang. Topic-based coherence modeling for statistical machine translation. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 23(3):483–493, March 2015.

[90] X. Yi and J. Allan. A comparative study of utilizing topic models for information retrieval. In *Proceedings of the 31th European Conference on IR Research on Advances in Information Retrieval*, ECIR '09, pages 29–41, Berlin, Heidelberg, 2009. Springer-Verlag.

[91] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: A location-content-aware recommender system. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 221–229, New York, NY, USA, 2013. ACM.

[92] K. Yu, B. Zhang, H. Zhu, H. Cao, and J. Tian. Towards personalized context-aware recommendation by mining context logs through topic models. In P.-N. Tan, S. Chawla, C. Ho, and J. Bailey, editors, *Advances in Knowledge Discovery and Data Mining*, volume 7301 of *Lecture Notes in Computer Science*, pages 431–443. Springer Berlin Heidelberg, 2012.

[93] X. Zhu, D. Blei, and J. Lafferty. Taglda: Bringing document structure knowledge into topic models. Technical Report TR-1553, University of Wisconsin, 2006.

Appendix A

Typed Dependency Relations

The Stanford typed dependencies relations were designed to provide a simple description of the grammatical relationships between consecutive and nonconsective words of a sentence [29]. The current representation of the set of Stanford typed dependencies relations contains 48 grammatical relations, denoted by $rel(w_i, w_j)$, where rel represents a relation between w_i and w_j. The grammatical relations are defined in Table A.1, in alphabetical order according to the dependency's abbreviated name.

Grammatical Relations	**Definition**	**Example**
acomp(w_i,w_j)	Adjectival complement: w_j is an adjective that complements a verb w_i.	"She looks very beautiful." $acomp(looks, beautiful)$
advcl(w_i,w_j)	Adverbial clause modifier: w_j is a clause that modifies a verb or a clause w_i.	"The accident happened as the night was falling." $advcl(happened, falling)$
advmod(w_i,w_j)	Adverb modifier: w_j is a non-clausal adverb that modifies the meaning of word w_i.	"Genetically modified food" $advmod(modified, genetically)$
agent(w_i,w_j)	Agent: w_j is the complement of a passive verb w_i which is introduced by the preposition "by" and does the action.	"The man has been killed by the police." $agent(killed, police)$
amod(w_i,w_j)	Adjective Modifier: w_j is an adjective that changes the meaning of w_i.	"Sam eats red meat." $amod(meat, red)$

Continued on next page

Table A.1 – *Continued from previous page*

Grammatical Relations	**Definition**	**Example**
appos(w_i,w_j)	Appositional modifier: w_j is a noun immediately to the right of the first noun w_i that modifies or defines w_i.	"Sam, my brother, arrived." $appos(Sam, brother)$
aux(w_i,w_j)	Auxiliary: w_j is a modal auxiliary of a clause, where the main verb is w_i.	"Reagan has died." $aux(died, has)$
auxpass(w_i,w_j)	Passive auxiliary: w_j is a modal auxiliary of a passive clause, where the main verb is w_i.	"Kennedy has been killed." $auxpass(killed, been)$
cc(w_i,w_j)	Coordination: w_j is an element of a conjunct and the coordinating conjunction word w_i.	"Bill is big and honest." $cc(big, and)$
ccomp(w_i,w_j)	Clausal complement: w_j is a dependent clause with an internal subject which functions like an object of the verb or adjective w_i.	"He says that you like to swim." $ccomp(says, like)$

Continued on next page

Table A.1 – *Continued from previous page*

Grammatical Relations	**Definition**	**Example**
conj(w_i,w_j)	Conjunct: A relation between two elements w_i and w_j connected by a coordinating conjunction, such as "and" and "or".	"Bill is big and honest." $conj(big, honest)$
conj-negcc(w_i,w_j)	Negated coordination: A "but not", "instead of", "rather than", and "but rather" relationship between w_i and w_j.	"computers but not laptops." $conj - negcc(computers, laptops)$
cop(w_i,w_j)	Copula: A relation between the complement of a copular verb w_i and the copular verb w_j. Normally, copula is taken as a dependent of its complement.	"Bill is big." $cop(big, is)$
csubj(w_i,w_j)	Clausal subject: w_j is a clausal syntactic subject of a clause w_i.	"What she said makes sense." $csubj(makes, said)$
csubjpass(w_i,w_j)	Clausal passive subject: w_j is a clausal syntactic subject of a passive clause w_i.	"That she lied was suspected by everyone." $csubjpass(suspected, lied)$

Continued on next page

Table A.1 – *Continued from previous page*

Grammatical Relations	**Definition**	**Example**
det(w_i,w_j)	Determiner: w_j is a determiner of the head of a noun phrase w_i.	"The man is here." $det(man, The)$
discourse(w_i,w_j)	Discourse element: w_j is used for interjections and other discourse particles and elements (which are not clearly linked to the structure of the sentence, except in an expressive way).	"Iguazu is in Argentina uh-huh." $det(is, uh - huh)$
dobj(w_i,w_j)	Direct object: w_j is the noun phrase which is the (accusative) object of the verb w_i.	"She gave me a raise." $dobj(gave, raise)$
expl(w_i,w_j)	Expletive: w_j is an existential there for the verb w_i.	"There is a ghost in the room" $expl(is, There)$
goeswith(w_i,w_j)	Goes with: This relation links two parts of a word that are separated in text that is not well edited.	"They come here with out legal permission." $goeswith(with, out)$
iobj(w_i,w_j)	Indirect object: w_j is a noun phrase which is a (dative) object of a verb w_i.	"She gave me a raise." $iobj(gave, me)$

Continued on next page

Table A.1 – *Continued from previous page*

Grammatical Relations	**Definition**	**Example**
mark(w_i,w_j)	Marker: w_j is a word introducing a finite clause subordinate to another clause w_i.	"He says that you like to swim." $mark(that, swim)$
mwe(w_i,w_j)	Multi-word expression: This relation is used for certain multi-word idioms that behave like a single function word.	"He cried because of you." $mwe(of, because)$
neg(w_i,w_j)	Negation modifier: w_j modifies a word w_i.	"Bill is not a scientist." $neg(scientist, not)$
nn(w_i,w_j)	Noun compound modifier: w_i is any noun that serves to modify the head noun w_j.	"Oil price future" $nn(future, oil)$
npadvmod(w_i,w_j)	Noun phrase as adverbial modifier: w_j is a noun phrase used as an adverbial modifier of a phrase w_i.	"The director is 65 years old." $npadvmod(old, years)$
nsubj(w_i,w_j)	Nominal subject: w_j is a subject of a verb w_i.	"Clinton defeated Dole." $nsubj(defeated, Clinton)$

Continued on next page

Table A.1 – *Continued from previous page*

Grammatical Relations	**Definition**	**Example**
nsubjpass(w_i,w_j)	Passive nominal subject: w_j is a noun phrase which is the syntactic subject of a passive clause w_i.	"Dole was defeated by Clinton" *nsubjpass(defeated, Dole)*
num(w_i,w_j)	Numeric modifier: the noun w_j is any number phrase that modifies the meaning of the noun w_i with a quantity.	"Sam ate 3 sheep." $num(sheep, 3)$
number(w_i,w_j)	Element of compound number: w_j is a part of a number phrase or currency amount w_i.	"I have four thousand sheep." $number(thousand, four)$
parataxis(w_i,w_j)	Parataxis: w_j is the main verb of a clause and w_i is other sentential element(s), such as a sentential parenthetical, a clause after a ":" or a ";".	"The guy, John said, left early in the morning." $parataxis(left, said)$
pcomp(w_i,w_j)	Prepositional complement: w_j is the clause or prepositional phrase complement of a preposition w_i.	"They heard about you missing classes." $pcomp(about, missing)$

Continued on next page

Table A.1 – *Continued from previous page*

Grammatical Relations	**Definition**	**Example**
pobj(w_i,w_j)	Object of a preposition: w_j is the head of a noun phrase following the preposition w_i.	"I sat on the chair." $pobj(on, chair)$
poss(w_i,w_j)	Possession modifier: w_j is the possessive determiner of the head of the noun w_i.	"their offices" $poss(offices, their)$
possessive(w_i,w_j)	Possessive modifier: w_j is the possessive modifier of the head of the noun w_i and the genitive "'s".	"Bill's clothes" $possessive(John,'s)$
preconj(w_i,w_j)	Preconjunct: w_j is the head of a noun phrase that appears at the beginning of a conjunction w_i (and puts emphasis on w_i).	"Both the boys and the girls are here." $preconj(boys, both)$
predet(w_i,w_j)	Predeterminer: The relation between the head of a noun w_i and a word that precedes and modifies the meaning of the noun determiner w_j.	"All the boys are here." $predet(boys, all)$

Continued on next page

Table A.1 – *Continued from previous page*

Grammatical Relations	**Definition**	**Example**
prep(w_i,w_j)	Prepositional modifier: w_j is a prepositional phrase that modifies the meaning of a verb, adjective, noun, or even another prepositon w_i.	"I saw a cat in a hat." $prep(cat, in)$
prepc(w_i,w_j)	Prepositional clausal modifier: w_j is the prepositional clausal modifier of w_i.	"He purchased it without paying a premium." $prepc(purchased, paying)$
prt(w_i,w_j)	Phrasal verb particle: w_j is the particle for the verb w_i.	"They shut down the station." $prt(shut, down)$
quantmod(w_i,w_j)	Quantifier phrase modifier: w_j modifies the head of a quantifier phrase constituent w_i.	"About 200 people came to the party." $quantmod(200, About)$
rcmod(w_i,w_j)	Relative clause modifier: w_j ia a verb in a relative clause that changes the meaning of w_i.	"I saw the man you love." $rcmod(man, love)$
ref(w_i,w_j)	Referent: w_j is a relative clause that modifies the noun w_i.	"I saw the book which you bought" $ref(book, which)$

Continued on next page

Table A.1 – *Continued from previous page*

Grammatical Relations	**Definition**	**Example**
root(w_i,w_j)	Root: It points to the root of the sentence; and acts as the root of the tree.	"I love French fries." $root(root, love)$
tmod(w_i,w_j)	Temporal modifier: w_j is a noun phrase constituent that modifies the meaning of a constituent w_i.	"Last night, I swam in the pool." $tmod(swam, night)$
vmod(w_i,w_j)	Reduced non-finite verbal modifier: w_j is a participial or infinitive form of a verb heading a phrase w_i.	"I don't have anything to say to you" $vmod(anything, say)$
xcomp(w_i,w_j)	Open clausal complement: w_j is a predicative or clausal complement of a verb or an adjective w_i without its own subject.	"I am ready to leave." $xcomp(ready, leave)$
xsubj(w_i,w_j)	Controlling subject: The relation between the head of an open clausal complement w_j and the external subject of that clause.	"Tom likes to eat fish." $xsubj(eat, Tom)$

Continued on next page

Table A.1 – *Continued from previous page*

Grammatical Relations	**Definition**	**Example**

Table A.1: Grammatical relations used in typed dependency parse trees, defined in de Marneffe *et al.* [29, 30].