Topic Modeling Using Collapsed Typed Dependency Relations

Elnaz Delpisheh and Aijun An

Department of Electrical Engineering and Computer Science,
York University
Toronto, ON, Canada M3J 1P3
{elnaz,aan}@cse.yorku.ca

Abstract. 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 paper, we propose a novel 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, 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.

1 Introduction

A large amount of text corpora and discrete data demands more on improving people's ability to interpret and comprehend them. Previously, texts were collected and stored in large text repositories and retrieved by a set of keywords. Documents were seldom analyzed using their themes, because there were very few technologies to extract their thematic structures. During the past decade, *topic modeling* has emerged to remedy the situation. Topic modeling is a powerful statistical tool to uncover hidden thematic structures of documents, to facilitate document summarization and organization in a variety of applications in natural language processing, vision, social network analysis, and text mining [1–3]. Most topic models consider documents to be a weighted mixture of topics, where each topic is a multinomial distribution over words. An inferred topic model of a corpus assigns high probability to members of the corpus as well as to other similar documents [1, 2]. Text documents are the only observed data in most conventional topic models. However, more recent topic models extend previous models by incorporating extra information [4]. Extra information is obtained by enriching text representation to include information, such as authors of the documents [5], images associated with the text [6], style of writing and reviewers of the documents [7]. The aforementioned topic models represent documents as

V.S. Tseng et al. (Eds.): PAKDD 2014, Part II, LNAI 8444, pp. 146–161, 2014.

a bag-of-words, where the order of words, thus important linguistic structures of documents are neglected [1, 2].

In order to include richer linguistic structures of text documents, many methods were proposed to incorporate local word dependencies into topic models [8–12]. 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 *"There are countries that deny human basic civil rights."*, the term *"human rights"* contains two nonconsecutive words *"human"* and *"rights"* that are syntactically related. In order to capture sequential consecutive dependencies between words, the Bigram Topic Model [8] and Topical n-gram Model [9] extend word generation by conditioning on n previous words. However, the n-gram topic models do not capture relations between nonconsecutive words.

To remedy this problem, some recent methods integrate grammatical regularities of text documents into topic models. HMM-LDA [11] uses the states of a Hidden Markov Model to represent syntactic and semantic words. Then, 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 [12]. 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) [10] 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 [13]. 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 [10].

Moreover, text documents consist of words with possible conceptual similarities, called *synonyms*, defined in lexical resources like WordNet [14]. It is reasonable to expect the distribution of topics over synonymous words to be similar.

In this paper, 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.

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

In addition, we investigate the influence of enforcing similar topic distribution over conceptually similar words by generalizing words with their synonyms.

The structure of this paper is as follows: In Section 2, we discuss our proposed topic model incorporated with collapsed typed dependency relations. We also explain our method for generalizing words using synonyms. Section 3 introduces some criteria to evaluate topic models. Then, it demonstrates the effectiveness of our approach through experiments. Finally, Section 4 concludes the paper with some remarks on our future work.

2 Main Contributions

In this section, we first explain the collapsed typed dependency relations and how to find them from the HPSG parse trees. These relations are used in capturing consecutive and nonconsecutive dependencies between words of text documents. We then describe our topic model and how it embodies collapsed typed dependency relations. In addition, we propose a method to enforce similar topic distribution over synonymous words of text documents. Lastly, we explain the relationship between our contributions and other related work.

2.1 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 [15].

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. The collapsed typed dependency relations are extracted from typed dependency parse trees. The typed dependency parse tree of a sentence provides a tree representation of detailed grammatical relations between words in the sentence [16]. Words in the sentence are nodes of the tree and grammatical relations are the edge labels. The total number of grammatical relations that can be assigned by typed dependency parse trees is 48 [16]. Table 1 shows most common grammatical relations used in typed dependency parse trees. For more information on this set of relations, please see [13].

Typed dependency parse trees are constructed according to the *Head-Driven Phrase Structure Grammar* (HPSG). HPSG, developed by Pollard *et al.* [17], is a highly structured grammatical representation of text documents that effectively analyzes syntactic relations concerning multi-word constituents [15, 16]. 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 1[2] represents the HPSG-based parse

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

Table 1. Most common grammatical relations used in typed dependency parse trees, defined in de Marneffe *et al.* [13, 16]

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	Adjective Modifier: An adjective that changes the meaning of the noun.	"Sam eats red meat." amod(meat, red)
rcmod	Relative Clause Modifier: A relative clause that changes the meaning of the noun.	"I saw the man you love." rcmod(man, love)
nsubj	Nominal Subject: A word that is the subject of the clause.	"Clinton defeated Dole." nsubj (defeated, Clinton)
dobj	Direct Object: A word that is the direct object of the verb.	"They win the lottery." dobj (win, lottery)
expl	Expletive: This relation captures the existential there.	"There is a ghost in the room." expl(is, There)

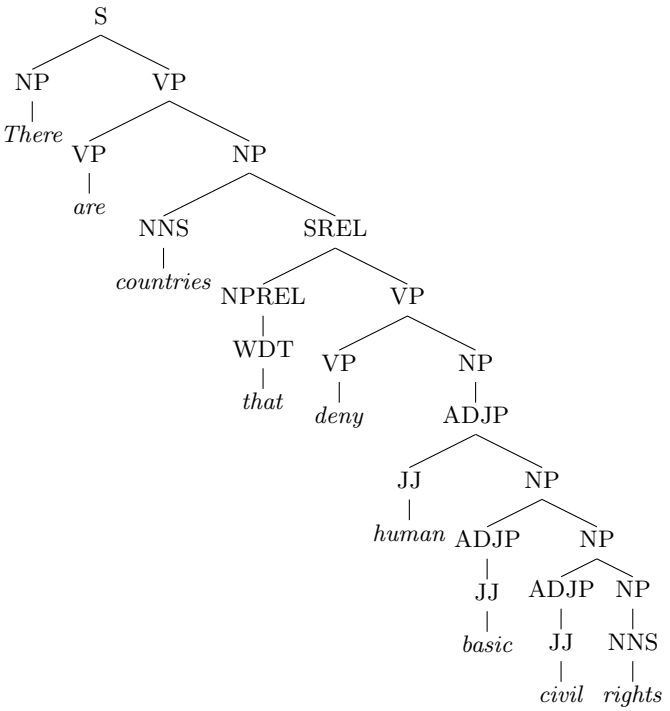

Fig. 1. The HPSG-based parse tree for the sentence *"There are countries that deny human basic civil rights."*. Abbreviations that are used in this tree are as follows: *S*: sentence; *VP*: verb phrase; *NNS*: plural noun; *SREL*: sentence relation; *NPREL*: noun phrase relation; *WDT*: wh-determiner; *ADJP*: adjective phrase; *JJ*: adjective.

tree of the sentence *"There are countries that deny human basic civil rights."*. In this tree, the leftmost branch, node *NP* represents the role of "noun phrase" for the leaf node *"There"*.

HPSG provides a high level syntactic representation of sentences in text documents [16]. However, we need to capture specific relations between every individual related pair of words. Thus, we need to elaborate HPSG to include additional labeled grammatical relations between words. This is achieved by constructing typed dependency parse trees from HPSG-based parse trees, using an algorithm described in [16]. This algorithm has two phases: dependency extraction and dependency typing. In the first phase, a sentence is parsed with a phrase structure grammar parser (HPSG). 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. Figure 2 shows the typed dependency parse tree constructed from Figure 1 for the sentence *"There are countries that deny human basic civil rights."*. As illustrated in this figure, nonconsecutive relations between words with gaps, i.e. *"human rights"* is captured under the *amod* relation. Typed dependency parse trees are constructed using the Stanford parser toolkit that has phrase structured grammars integrated in [13, 16][3].

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 2, is as follows: {*expl*(are, There), *nsubj*(are, countries), *nsubj*(deny, that), *rcmod*(countries, deny), *amod*(rights, human), *amod*(rights, basic), *amod*(rights, civil), dobj(deny, rights)}. 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. Each relation can have a discriminate topic.

The relations from typed dependency parse trees are further processed by collapsing relations involving prepositions and conjuncts to get direct dependencies between content words [16]. For instance, in the set of the aforementioned typed dependency relations, the relations involving the preposition *"that"* will be collapsed. Thus, relations *rcmod*(countries, deny) and *nsubj*(deny, that) will become *rcmod*(countries, deny) and *nsubj*(deny, countries). 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. In our work, we use the collapsed typed dependency relations to represent the corpus. Note that the order of words in the collapsed typed dependency relations matters.

2.2 Probabilistic Topic Model Using Collapsed Dependency Relations

We assume that corpus $\mathcal{D}$ consists of M documents denoted by $\mathcal{D} = \{d_1, d_2, \cdots, d_M\}$. Each document d_l contains n words denoted by $d_l = \{w_1, w_2, \cdots, w_n\}$.

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

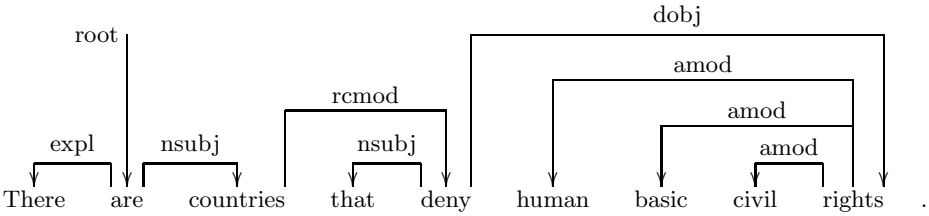

Fig. 2. The typed dependency parse tree of the sentence *"There are countries that deny human basic civil rights."*. See Table 1 for the explanation of each relation. As illustrated in this figure, the typed dependency parse tree effectively captures relations between nonconsecutive words, i.e., *dobj* relation between words *deny* and *right*.

Each document is represented by R collapsed dependency relations between words of the document, denoted by $\mathbf{R} = \{r_1, r_2, \cdots, r_R\}$. These relations are instances of the 48 grammatical relations described in Section 2.1, each of which consists of two words.

Our topic model assumes that each document d_l has a multinomial distribution over K topics with parameters $\Theta^{(d_l)}$. Thus, for a relation in document d_l, $P(z_l = j|\mathcal{D} = d_l) = \Theta_j^{(d_l)}$, where z_l denotes topic assignment to relation l. In our proposed model, the jth topic is represented by a multinomial distribution over R relations with parameters $\Phi^{(j)}$, thus $P(r_l|z_l = j) = \Phi_{r_l}^{(j)}$. Inspired from LDA [1, 2, 18], we provide a procedure to generate documents. In this procedure, each document d_l is generated by first drawing a distribution over topics ($\Theta^{(d_l)}$), generated from a Dirichlet distribution with parameter α. The relations in the document are then generated by drawing a topic j from this distribution and then drawing a relation from that topic according to a multinomial distribution with parameters ($\Phi_{r_l}^{(j)}$), generated from a Dirichlet distribution with parameter β.

Note that the only observed variables are the relations in the collection of documents. Document distribution over topics and topic distribution over relations are latent variables generated from Dirichlet distributions with parameters α and β, 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 [19].

We adopt Gibbs sampling algorithm proposed by Griffiths *et al.* [2, 18] to draw a topic from the conditional distribution iteratively. For each topic j the distribution is given by

$$P(z_l = j|\mathbf{z}_{-l}, \mathbf{R}) \propto P(r_l|z_l = j, \mathbf{z}_{-l}, \mathbf{R}_{-l}) P(z_l = j|\mathbf{z}_{-l}), \tag{1}$$

where $\mathbf{z}_{-l}$ and $\mathbf{R}_{-l}$ denote the $\mathbf{z}$ and $\mathbf{R}$ for all relations other than r_l. This expression is an instance of Bayes' rule with $P(r_l|z_l = j, \mathbf{z}_{-l}, \mathbf{R}_{-l})$ as the likelihood of

the data given a particular choice of z_l and $P(z_l = j|\mathbf{z}_{-l})$ as the prior on z_l. The likelihood is obtained by integrating over the parameters Φ, which results in

$$P(r_l|z_l = j, \mathbf{z}_{-l}, \mathbf{R}_{-l}) = \frac{n_{-l,j}^{(r_l)} + \beta}{n_{-l,j}^{(\cdot)} + R\beta}, \tag{2}$$

where $n_{-l,j}^{(\cdot)}$ is the total number of relations assigned to topic j, excluding the current one, and $n_{-l,j}^{(r_l)}$ is the total number of relation r_l assigned to topic j, excluding the current one.

Similarly, the prior is calculated by integrating over the parameters Θ:

$$P(z_l = j|\mathbf{z}_{-l}) = \frac{n_{-l,j}^{(d_l)} + \alpha}{n_{-l,\cdot}^{(d_l)} + K\alpha}, \tag{3}$$

where $n_{-l,j}^{(d_l)}$ is the total number of relations from document d_l assigned to topic j, excluding the current one, and $n_{-l,\cdot}^{(d_l)}$ is the total number of relations in document d_l, excluding the current one.

Then, the conditional distribution for the topic assignments is given by

$$P(z_l = j|\mathbf{z}_{-l}, \mathbf{R}) \propto \frac{n_{-l,j}^{(r_l)} + \beta}{n_{-l,j}^{(\cdot)} + R\beta} \frac{n_{-l,j}^{(d_l)} + \alpha}{n_{-l,\cdot}^{(d_l)} + K\alpha}. \tag{4}$$

2.3 Generalizing Words Using Synonyms

Text documents often contain words that are synonyms. Sets of synonyms can be obtained from lexical resources like WordNet [14]. In this work, we investigate the influence of generalizing words using a synonym on topic modeling.

Similar to LDA [1], we assume that a document is a multinomial distribution over K topics, where each topic is a multinomial distribution over N 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*.

2.4 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 [10] 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.* [10]. 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 labeled 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 [8] 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.* [20] employs WordNet to improve topic models by removing unrelated words from the simplified topic descriptions. Mei *et al.* [21] used WordNet to label each topic in a multinomial topic model. Newman *et al.* [22] uses WordNet to evaluate topic coherence. None of them uses synonyms to generalize words prior to building topic models.

3 Experiments

We conducted experiments on two text corpora to compare the performance of four following topic models: LDA [1], LDA on generalized words using synonyms, explained in Section 2.3, the Bigram Topic Model [8], and the HPSG Topic Model, explained in Section 2.2[4]. The first three topic models were trained with 1000 iterations of Gibbs sampling [2, 18] used in the MALLET [23]. However, the HPSG Topic Model was trained with 1000 iterations of Gibbs sampling. 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 the MALLET [23].

In our experiments we used Associated Press corpus[5] that consists of 2246 Associated Press articles, 33872 words, and 454370 collapsed typed dependency relations. In addition, we used Reuters-21578 Distribution 1.0[6] that includes 22 files. Each of the first 21 files contain 1000 documents, while the last file contains 578 documents. This corpus contains a total number of 43012 words and 793345 collapsed typed dependency relations.

Table 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 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 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,

[4] Section 2.4 provides a theoretical comparison between our proposed probabilistic topic model and STM [10]. Given the fact that the source codes of STM was not available prior to the submission of this paper, experimental comparisons with this method will be done in our future work.

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

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

Table 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 Topic Model from Reuters corpus

LDA	LDA on generalized words using synonyms	the Bigram Topic Model	the HPSG 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

as opposed to the Bigram Topic Model, terms generated by the HPSG 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) [1].
- Stable distribution of topics over words across samples [5].
- Coherent distribution of words learned by individual topics [22].
- Accurate distribution of topics over words.

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

3.1 Perplexity

Perplexity is the most common criterion to evaluate the quality of topic models [24]. Perplexity measures the cross-entropy between the word distribution learned by the topic model and the distribution of words in an unseen test document. Thus, lower perplexity score indicates that the model is better in predicting distribution of the test document [1, 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 Topic Model by

$$perplexity(\mathbf{D}_{test}) = exp\left(-\frac{\sum_{d=1}^{M} logP(\mathbf{R}_d)}{\sum_{d=1}^{M} |\mathbf{R}_d|}\right), \tag{5}$$

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

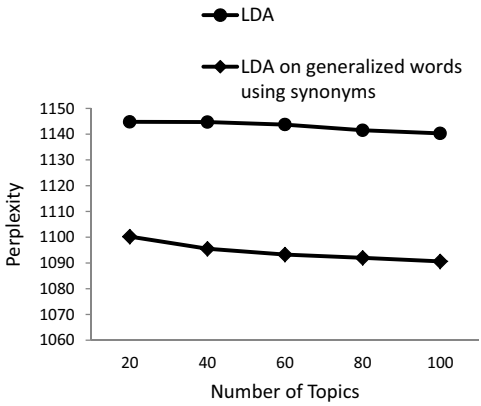

Fig. 3. Perplexity as a function of number of topics, using LDA, and LDA on generalized words using synonyms on Association Press corpus

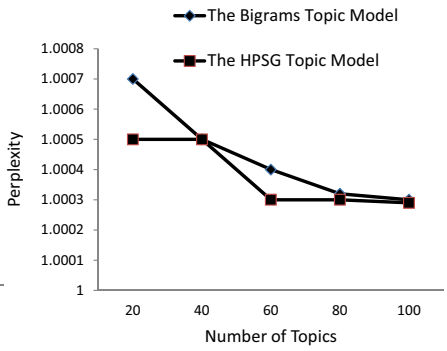

Fig. 4. Perplexity as a function of number of topics, using the Bigram Topic Model, and the HPSG Topic Model on Association Press corpus

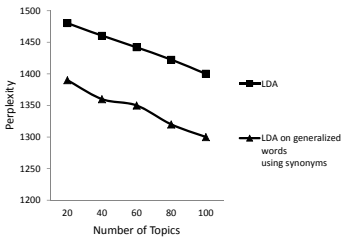

Fig. 5. Perplexity as a function of number of topics, using LDA, and LDA on generalized words using synonyms on Reuters corpus

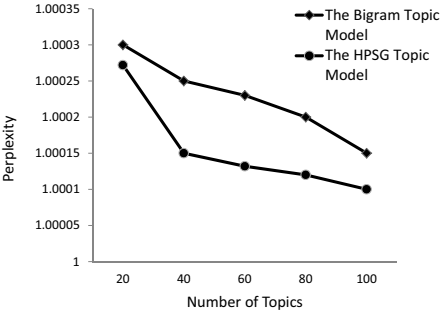

Fig. 6. Perplexity as a function of number of topics, using the Bigram Topic Model, and the HPSG Topic Model on Reuters corpus

typed dependency relations in document d, and $P(\mathbf{R}_d)$ is the probability estimate assigned to $\mathbf{R}_d$ by the HPSG topic model. The perplexity of $\mathbf{D}_{test}$ by other topic models, such as LDA, is defined similarly, except that $\mathbf{R}_d$ is replaced by $\mathbf{W}_d$, the set of words in the corpus.

The results are illustrated in Figures 3, 4, 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 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.

Table 3. Topic stability across two different runs of the HPSG Topic Model on Reuters corpus

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. Topic stability across two different runs of LDA 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

3.2 Stability

Stability is the similarity of topic distributions over words across different samples [5]. We follow the algorithm proposed by Rosen-Zvi *et al.* [5] 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 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 [26]. We compare the stability of topic distributions over relations across samples, generated by the HPSG Topic Model and LDA on the Reuters corpus. The results, illustrated in Tables 3 and 4, 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.

3.3 Topic Coherence

Topic coherence measures the integrity or coherence of a set of words generated by a topic model. Words generated by topic T, denoted by $T = \{w_1, w_2, \cdots, w_n\}$,

Table 5. The average topic coherence of top 50 words of 20 topics generated from Reuters corpus

Topic model	Coherence
LDA	41.35
LDA on generalized words using synonyms	41.68
the Bigram Topic Model	39.18
the HPSG Topic Model	39.79

Table 6. The average accuracy of topic distribution over words from a subset of topic-labeled Reuters corpus

Topic model	Accuracy
LDA	0.225
LDA on generalized words using synonyms	0.220
the Bigram Topic Model	0.221
the HPSG Topic Model	0.223

are coherent if they are semantically similar. In order to calculate the topic coherence score, we adopted the method proposed by Newman *et al.* [22]. We calculate the semantic similarity scores between every pair of words in a topic using the Lesk algorithm [27][7]. Then, we compute their arithmetic means. 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 Topic Model on Reuters corpus. The results are shown in Table 5. The HPSG Topic Model generates slightly more coherent topic distributions over words than The Bigram Topic Model. The HPSG Topic Model performs comparable to LDA in topic coherence. However, 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.

3.4 Accuracy

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

We assume that the test corpus $\mathcal{T}$ consists of M documents $\mathcal{T} = \{d_1, d_2, \cdots, d_M\}$. 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 n words, denoted by $T = \{w_1, w_2, \cdots, w_n\}$. The accuracy score of the topic model is calculated by computing $Accuracy = \frac{\sum_{i=1}^{M} min_{j=1,\cdots,K} d(T_j, L)}{M}$, where $d(T_j, L)$ denotes the semantic similarity between two sets of T_j and L. This semantic similarity is measured using the Lesk algorithm, explained in Section 3.3.

We compared the accuracy of LDA, LDA on generalized words using synonyms, the Bigram Topic Model, and the HPSG Topic Model on a subset of

[7] The Lesk algorithm uses dictionary definitions of two words in a pair and counts the number of words that are shared between two definitions. The more overlapping the definitions are, the more related the words are. The Lesk toolkit is available at `http://text-similarity.sourceforge.net`

Reuters corpus that contains topic-labeled documents. As illustrated in Table 6, these algorithms are comparable in terms of accuracy. However, LDA is slightly better.

4 Conclusions

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.

In the future, we will extend our topic model to effectively capture more dependencies between words in sentence or document levels. In addition, we will investigate the influence of the order of words in the collapsed typed dependency relations.

Acknowledgement. This project is funded in part by the Center for Information Visualization and Data Driven Design (CIV/DDD) established by the Ontario research fund.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
2. Griffiths, T.L., Steyvers, M.: Finding scientific topics. Proceedings of the National Academy of Sciences of the United States of America 101, 5228–5235 (2004)
3. Su, H., Tang, J., Hong, W.: Learning to diversify expert finding with subtopics. In: Tan, P.-N., Chawla, S., Ho, C.K., Bailey, J. (eds.) PAKDD 2012, Part I. LNCS, vol. 7301, pp. 330–341. Springer, Heidelberg (2012)
4. Andrzejewski, D.M.: Incorporating Domain Knowledge in Latent Topic Models. PhD thesis, University of Wisconsin-Madison, USA (2010)
5. Rosen-Zvi, M., Chemudugunta, C., Griffiths, T., Smyth, P., Steyvers, M.: Learning author-topic models from text corpora. ACM Trans. Inf. Syst. 28, 4:1–4:38 (2010)

6. Blei, D.M., Jordan, M.I.: Modeling annotated data. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, SIGIR 2003, pp. 127–134. ACM, New York (2003)
7. Mimno, D., McCallum, A.: Expertise modeling for matching papers with reviewers. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007, pp. 500–509. ACM, New York (2007)
8. Wallach, H.M.: Topic modeling: Beyond bag-of-words. In: NIPS 2005 Workshop on Bayesian Methods for Natural Language Processing (2005)
9. Wang, X., McCallum, A., Wei, X.: 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 2007, pp. 697–702. IEEE Computer Society, Washington, DC (2007)
10. Boyd-Graber, J.L., Blei, D.M.: Syntactic topic models. CoRR abs/1002.4665 (2010)
11. Griffiths, T.L., Steyvers, M., Blei, D.M., Tenenbaum, J.B.: Integrating topics and syntax. In: Advances in Neural Information Processing Systems 17, pp. 537–544. MIT Press (2005)
12. Gruber, A., Rosen-zvi, M., Weiss, Y.: Hidden topic markov models. In: Proceedings of Artificial Intelligence and Statistics (2007)
13. de Marnee, M.C., Manning, C.D.: Stanford typed dependencies manual (2012)
14. Miller, G.A.: Wordnet: a lexical database for english. Commun. ACM 38, 39–41 (1995)
15. Levine, R.D., Meurers, W.D.: Head-driven phrase structure grammar: Linguistic approach, formal foundations, and computational realization. Elsevier, Oxford (2006)
16. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: Proc. Intl. Conf. on Language Resources and Evaluation LREC, pp. 449–454 (2006)
17. Pollard, C., Sag, I.A.: Information-based syntax and semantics: Vol. 1: Fundamentals, Stanford, CA, USA. Center for the Study of Language and Information (1988)
18. Griffiths, T.: Gibbs sampling in the generative model of latent dirichlet allocation. Standford University 518, 1–3 (2002)
19. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus (2006)
20. Musat, C., Velcin, J., Rizoiu, M.-A., Trausan-Matu, S.: Concept-based topic model improvement. In: Ryżko, D., Rybiński, H., Gawrysiak, P., Kryszkiewicz, M. (eds.) Emerging Intelligent Technologies in Industry. SCI, vol. 369, pp. 133–142. Springer, Heidelberg (2011)
21. Mei, Q., Shen, X., Zhai, C.: Automatic labeling of multinomial topic models. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007, pp. 490–499. ACM, New York (2007)
22. Newman, D., Lau, J.H., Grieser, K., Baldwin, T.: 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 2010, pp. 100–108. Association for Computational Linguistics, Stroudsburg (2010)
23. McCallum, A.K.: Mallet: A machine learning for language toolkit (2002), `http://mallet.cs.umass.edu`
24. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 1st edn. Prentice Hall PTR, Upper Saddle River (2000)

25. Chemudugunta, C., Holloway, A., Smyth, P., Steyvers, M.: Modeling documents by combining semantic concepts with unsupervised statistical learning. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 229–244. Springer, Heidelberg (2008)
26. Bigi, B.: Using kullback-leibler distance for text categorization. In: Sebastiani, F. (ed.) ECIR 2003. LNCS, vol. 2633, pp. 305–319. Springer, Heidelberg (2003)
27. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 136–145. Springer, Heidelberg (2002)