Latent Topic Embedding

Di Jiang
Baidu, Inc., China
jiangdi@baidu.com

Lei Shi
Baidu, Inc., China
shilei06@baidu.com

Rongzhong Lian
Baidu, Inc., China
lianrongzhong@baidu.com

Hua Wu
Baidu, Inc., China
wu_hua@baidu.com

Abstract

Topic modeling and word embedding are two important techniques for deriving latent semantics from data. General-purpose topic models typically work in coarse granularity by capturing word co-occurrence at the document/sentence level. In contrast, word embedding models usually work in fine granularity by modeling word co-occurrence within small sliding windows. With the aim of deriving latent semantics by capturing word co-occurrence information at different levels of granularity, we propose a novel model named *Latent Topic Embedding* (LTE), which seamlessly integrates topic generation and embedding learning in one unified framework. We further propose an efficient Monte Carlo EM algorithm to estimate the parameters of interest. By retaining the individual advantages of topic modeling and word embedding, LTE results in better latent topics and word embedding. Experimental results verify the superiority of LTE over the state-of-the-arts in real-life applications.

1 Introduction

Topic modeling and word embedding are gaining significant momentum in the field of text mining. General-purpose topic models such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003) and Sentence LDA (Jo and Oh, 2011) usually utilize word co-occurrences at the document/sentence level to compose the ”topics”, which capture the latent semantics between words. These models are plagued by the simplistic bag-of-words assumption, which ignores the valuable sub-sequence information between words. Some recent endeavors introduced n-gram information into topic models (Wallach, 2006), however, the size of vocabulary is significantly enlarged and these techniques are hardly feasible for real-life applications. Therefore, the technique of topic modeling needs a remedy for solving the word sequence problem with fairly low cost. Word embedding models such as Word2Vec (Mikolov et al., 2013a) map words into distributed representations. Word embedding models primarily focus on the word co-occurrences within small sliding windows, which enable word embedding to capture (at least partially) the information of word sequences. One key problem of the existing word embedding models is that they are typically short-sighted and are not aware of the themes of the document.

With their differences, the core of topic modeling and word embedding is based upon the assumption that the words co-occurring frequently should have semantic commonality. In light of their individual advantages and drawbacks, we see that the two techniques are essentially complimentary and can be integrated to enhance each other. In this paper, we propose the *Latent Topic Embedding* (LTE) to seamlessly integrate topic modeling and word embedding in one framework. In the generative process of LTE, we assume that the observed words in document can be generated through two channels: one is through the Multinomial distribution and the other is based upon topic embeddings as well as word embeddings. In this way, the embedding information influences the result of topic modeling while the topic information affects the training of word embeddings in return. LTE enables topic modeling to utilize word sequence information and it equips word embeddings with the document-level vision. We propose a Monte Carlo

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers,
pages 2689–2698, Osaka, Japan, December 11-17 2016.

EM algorithm to efficiently infer the parameters of interest in LTE. Extensive experiments on real-life applications verify the superiority of LTE over several strong baselines.

The rest of this paper is organized as follows. In Section 2, we review the related work. In Section 3, we discuss the technical details of LTE. In Section 4, we illustrate how to conduct parameter inference for LTE. In Section 5, we present the experimental results. Finally, we conclude this paper in Section 6.

2 Related Work

The present work is related to previous research on topic modeling and word embedding. Latent Dirichlet allocation (LDA) is a generative probabilistic model for collections of discrete data such as text corpora (Blei et al., 2003)(Griffiths and Steyvers, 2004). LDA and its variants has been widely employed in many texting mining scenarios (Wang and McCallum, 2006)(Krestel et al., 2009)(Xu et al., 2009)(Jiang et al., 2013) and demonstrated promising performance. It is worth mentioning that some work such as the bigram topic model (Wallach, 2006) aims to alleviate the negative effect of bag-of-words assumption in LDA. However, considerable computational cost is involved since the bigram model creates a multinomial distribution for each pair of the topics and the words, the amount of which is usually voluminous. While topic modeling received intensive research in the field of Bayesian network research, word embedding received much attention in the field of neural network. Word embedding (Bengio et al., 2003) is proposed to fight the curse of dimensionality by learning a distributed representation for words which allows each training sentence to inform the model about an exponential number of semantically neighboring sentences. Mikolov presented several extensions of Skip-gram that improve both the quality of the vectors and the training speed (Mikolov et al., 2013b) (Mikolov et al., 2013a). Paragraph vector that is an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts was proposed in (Le and Mikolov, 2014). Word embedding is adapted for incorporating contextual information in learning vector-space representations of situated language (Bamman et al., 2014). A more relevant work is (Liu et al., 2015), which inputs the result of topic modeling into word embedding models to learn the topical word embedding. The major difference between this work and ours is that they did not aim to integrate topic modeling and word embedding and yet only utilizes the result of topic modeling as the input of word embedding models. Recently, (Nguyen et al., 2015) extended two Dirichlet multinomial topic models by incorporating word embeddings to improve the word-topic mapping. (Li et al., 2016) proposed a generative model that replaces the Multinomial word generation assumption of LDA with embedding based assumption.

Although topic modeling and word embedding receive intensive attention in recent years, to the best of our knowledge, there is no previous endeavor on integrating them together as a joint learning task to enhance each other. LTE paves the way for collectively modeling of word co-occurrence information at different granularity levels while retaining the topic modeling result as well as the word embedding result.

3 Generative Process of Latent Topic Embedding

Latent Topic Embedding (LTE) views each document as a bag of sentences and each sentence is composed of words. The generative process of LTE is formally depicted in Algorithm 1. For each topic k, the corresponding multinomial topic-word distribution ϕ_k is drawn from $Dirichlet(\beta)$. When generating a document, a multinomial document-topic distribution θ_d is drawn from $Dirichlet(\alpha)$. For each sentence s in the document, we draw a latent topic z_{ds} based on the document-topic distribution. For each token in the document, we drawn an indicator i from Bernoulli(τ). If i is 0, the word w is generated according to topic-word distribution $\phi_{z_{ds}}$. If i is 1, the word w is generated according to topic-word distribution $P(w|z_{ds}, C_w, M)$, which is defined as follows:

$$P(w|z_{ds}, C_w, M) = P(v_w|\mathbf{x}_w) = \frac{e^{\mathbf{x}_w \cdot v_w}}{\sum_{w'} e^{\mathbf{x}_w \cdot v_{w'}}}. \tag{1}$$

In Eq. (1), C_w stands for the sliding window for w. Specifically, C_w contains several words that precedes w. where $M = \{\mathbf{v_w}, \mathbf{v_z}\}$ stands for the word embedding and the topic embedding, $\mathbf{x}_w$ is the result

Algorithm 1: Generative Process

```
for each topic $k \in (1, 2, ..., K)$ do
    draw a word distribution $\phi_k \sim$ Dirichlet ($\beta$);
end
for each document $d$ do
    draw a topic distribution $\theta_d \sim$ Dirichlet ($\alpha$);
    for each sentence $s$ in $d$ do
        draw a topic $z_{ds} \sim$ Multinomial ($\theta_d$)
        for each token in $s$ do
            draw an indicator $i \sim$ Bernoulli($\tau$)
            if $i = 0$ then
                generate word $w \sim$ Multinomial ($\phi_{z_{ds}}$)
            end
            else
                generate word $w \sim P(w|z_{ds}, C_w, M)$
            end
        end
    end
end
```

of element-wise addition of the word embeddings of C_w and the topic embedding indexed by z_{ds} (i.e., $\mathbf{x}_w = \mathbf{v}_{c_w} \oplus v_{z_{ds}}$) and v_w is the embedding of w. The parameters of interest are ϕ, θ and M.

4 Training LTE

In Section 4.1, we describe how to sample the latent topics for sentences. In Section 4.2, we discuss how to optimize the vectors via stochastic gradient descent. The parameter inference algorithm is formally presented in Section 4.3.

4.1 Sampling Latent Variables

By translating the generative process of LTE into joint distribution, we aim to maximize the likelihood of the observed words $\mathbf{w}$: $P(\mathbf{w}|\alpha, \beta, \tau, M)$. Ideally, we would compute optimal M by maximizing $P(\mathbf{w}|\alpha, \beta, \tau, M)$ directly. However, evaluating this likelihood is intractable and what can be computed is the complete likelihood $p(\mathbf{w}, \mathbf{i}, \mathbf{z}|\alpha, \beta, \tau, M)$:

$$
\begin{aligned}
P(\mathbf{w}, \mathbf{i}, \mathbf{z}|\alpha, \beta, \tau, M) &= P(\mathbf{z}|\alpha)P(\mathbf{i}|\tau)P(\mathbf{w}|\mathbf{z}, \mathbf{i}, \beta, M) \\
= \Big(\frac{\Gamma(\sum_{z=1}^{T} \alpha_z)}{\prod_{z=1}^{T} \Gamma(\alpha_z)}\Big)^D \prod_{d=1}^{D} \frac{\prod_{z=1}^{T} \Gamma(m_{dz} + \alpha_z)}{\Gamma(\sum_{z=1}^{T}(m_{dz} + \alpha_z))} &\Big(\frac{\Gamma(\sum_{v=1}^{V} \beta_v)}{\prod_{v=1}^{V} \Gamma(\beta_v)}\Big)^T \prod_{z=1}^{T} \frac{\prod_{v=1}^{V} \Gamma(n_{zv} + \beta_v)}{\Gamma(\sum_{v=1}^{V}(n_{zv} + \beta_v))} \\
&\prod_{d=1}^{D} \prod_{s \in d} \prod_{w \in s} P(v_w|\mathbf{x}_w)^{\mathcal{I}(i_w=1)} \times (1-\tau)^A \tau^B,
\end{aligned}
\tag{2}
$$

where m_{dz} is the number of sentences that are assigned to topic z in document d. n_{zv} is the number of times that v is assigned to topic z through Multinominal distribution and $\Gamma(\cdot)$ indicates Gamma function, A is the number of 0 that are generated by the Bernoulli distribution and B is the number of 1 that are generated by the Bernoulli distribution. By applying Bayes rule, the full conditional of assigning topic k to z_{ds} is obtained as follows:

$$
\begin{aligned}
P(z_{ds} = k, \mathbf{i}_{ds}|\mathbf{w}, \mathbf{z}_{-ds}, \mathbf{i}_{-ds}, \alpha, \beta, \tau, M) &= (1-\tau)^{A_s} \tau^{B_s} \\
\frac{m_{dk} + \alpha_k}{\sum_{k'=1}^{K}(m_{dk'} + \alpha_{k'})} \frac{\Gamma(\sum_{w=1}^{W}(n_{kw} + \beta_w))}{\Gamma(\sum_{w=1}^{W}(n_{kw} + \beta_w + N_{iw}))} \prod_{w \in \mathbf{W} \& i_w=0} &\frac{\Gamma(n_{kw} + \beta_w + N_{iw})}{\Gamma(n_{kw} + \beta_w)} \prod_{w \in s \& i_w=1} P(v_w|\mathbf{x}_w)
\end{aligned}
\tag{3}
$$

The possible combinations of $\mathbf{i}_{ds}$ is exponential in the length of the sentence s. Similar to (Nguyen et al., 2015), we conduct approximation of the above equation and integrate out $\mathbf{i}_{ds}$,

$$P(z_{ds} = k|\mathbf{w}, \mathbf{z}_{-ds}, \mathbf{i}_{-ds}, \alpha, \beta, \tau, M) \approx \frac{m_{dk} + \alpha_k}{\sum_{k'=1}^{K}(m_{dk'} + \alpha_{k'})} \prod_{w \in s} \left((1-\tau)\frac{n_{kw} + \beta_w}{\sum_{w=1}^{W}(n_{kw} + \beta_w)} + \tau P(v_w|\mathbf{x}_w)\right) \tag{4}$$

Exactly calculating $P(v_w|\mathbf{x}_w)$ is computational infeasible, since the normalization term involves all the words in the vocabulary. Thus, we utilize noise contrastive estimation (NCE) to approximate it. The advantage of NCE is that it allows us to fit models that are not explicitly normalized making the training time effectively independent of the vocabulary size. Thus, we will be able to drop the normalization factor from the above equation, and simply use $e^{\mathbf{x}_w \cdot v_w}$ in place of $P(\mathbf{x}_w|v_w)$. Similar to the method described in (Mnih and Teh, 2012)(Dyer, 2014), we fixing the normalized constants in $P(\mathbf{x}_w|v_w)$ to 1, then we obtain the following approximation:

$$P(z_{ds} = k|\mathbf{w}, \mathbf{z}_{-ds}, \mathbf{i}_{-ds}, \alpha, \beta, \tau, M) \propto (m_{dk} + \alpha_k) \prod_{w \in s} \left((1-\tau)\frac{n_{kw} + \beta_w}{\sum_{w=1}^{W}(n_{kw} + \beta_w)} + \tau e^{\mathbf{x}_w \cdot v_w}\right) \tag{5}$$

For each word w in sentence s, its latent indicator i_w is sampled as follows:

$$P(i_w = 0|z_{ds} = k) \propto (1-\tau)\frac{n_{kw} + \beta_w}{\sum_{w=1}^{W}(n_{kw} + \beta_w)} \tag{6}$$

$$P(i_w = 1|z_{ds} = k) \propto \tau e^{\mathbf{x}_w \cdot v_w} \tag{7}$$

The above sampling process repeats for a predefined number of iterations. It is worth mentioning that there are works about scaling up Gibbs sampling or make it more efficient. Since the topic of designing better Gibbs sampling algorithms is beyond the scope of this paper, interested readers may refer to (Newman et al., 2009) and (Wang et al., 2009) for more detailed information.

4.2 Embedding Optimization

Now we convert the joint likelihood in Eq. (2) to its logarithm form, which is defined as follows:

$$\begin{aligned}\mathcal{L}(\mathbf{w}, \mathbf{i}, \mathbf{z}; \alpha, \beta, \tau, M) = D \log\left(\frac{\Gamma(\sum_{z=1}^{T}\alpha_z)}{\prod_{z=1}^{T}\Gamma(\alpha_z)}\right) + \sum_d \sum_z \log(\Gamma(m_{dz} + \alpha_z)) - \\ \sum_d \log(\Gamma(\sum_z (m_{dz} + \alpha_z))) T \log\left(\frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\right) + \sum_z \sum_v \log(\Gamma(n_{zv} + \beta_v)) - \\ \sum_z \log(\Gamma(\sum_v (n_{zv} + \beta_v))) + \sum_d \sum_{s \in d} \sum_{w \in s \& i_w = 1} \log P(v_w|\mathbf{x}_w) + A \log(1-\tau) + B \log \tau.\end{aligned} \tag{8}$$

Eq. (8) is a separable function. Each hyperparameter can be independently maximized. The hyperparameters α, β and τ can be straightforwardly optimized by Newton-Raphson algorithm like (Blei et al., 2003). As we usually utilize fixed α and β, the focus now is to illustrate how to optimize the vectors in M through maximizing $\sum_d \sum_{s \in d} \sum_{w \in s \& i_w = 1} \log P(v_w|\mathbf{x}_w)$whose corresponding NCE log-likelihood is as follows:

$$\begin{aligned}\sum_d \sum_{s \in d} \sum_{u \in w \cup NEG(w)} \Big\{ l_u^{c_w} \cdot \log[\sigma(\mathbf{x}_w \cdot v_u - \log(\frac{|NEG|}{|V|}))] + \\ [1 - l_u^{c_w}] \cdot \log[1 - \sigma(\mathbf{x}_w \cdot v_u - \log(\frac{|NEG|}{|V|}))] \Big\},\end{aligned} \tag{9}$$

where $\sigma(\cdot)$ to denote the sigmoid function, |NEG| is the number of negative samples for each word and $|V|$ is the size of vocabulary. We use stochastic gradient descent to optimize the embedding, the update formula for v_u in C_w is as follows:

$$v_u := v_u + \eta \sum_{u' \in w \cup NEG(w)} \Big[l_{u'}^{c_w} - \sigma(\mathbf{x}_w \cdot v_{u'} - \log(\frac{|NEG|}{|V|})) \Big] \cdot v_{u'}, \tag{10}$$

where $NEG(w)$ stands for the negative samples of w. The update formula for the topic embedding v_z is as follows:

$$v_z := v_z + \eta \sum_{u' \in w \cup NEG(w)} \Big[l^{c_w}_{u'} - \sigma(\mathbf{x}_w \cdot v_{u'} - \log(\frac{|NEG|}{|V|})) \Big] \cdot v_{u'}. \tag{11}$$

4.3 Monte Carlo EM

Based on the above discussion, we now formally present the parameter inference of LTE in Algorithm 2. After applying this algorithm, we obtain the quantities of interest such as Θ, Φ the topic embeddings and the word embeddings. Note that LTE covers both the outputs of topic model and the output of word embedding. Theoretically, it can be applied in any scenario where topic modeling or word embedding is previously utilized. In the experiments, we will show that retaining both the outputs of topic model and word embedding is critical for comprehensively capturing different kinds of latent semantics in text.

Algorithm 2: Monte Carlo EM

```
repeat
    run Gibbs sampling according to Eq. (5)(6)(7) ;
    optimize the corresponding parameters according to Eq. (10) and (11);
until a predefined number of iterations;
```

5 Experiments

In this section, we evaluate the performance of LTE. Unless otherwise stated, the experimental results are obtained when the size of the embedding is set to 20 and the size of the sliding window is 5. Similar insights are obtained when varying the two parameters and we skip them due to space limitation. In Section 5.1, we present some topic examples. In Section 5.2, we show the result of perplexity evaluation. In Section 5.3, we evaluate the the performance LTE through a task of topical word extraction.

Table 1: LTE Topic Examples (The number in brackets is the frequency of the word)

	Multinomial Perspective	Embedding Perspective
Topic1	Taiwan(2332), China(30904), issue(19080), unity(2165), relationship(6052), principle(2256), Taiwan independence(20), people(4172), mainland(1125), peace(699)	party(2), legislator(21), two states theory(1), tamper(42), attentively(1), Frank Hsieh(2), beautify(141), Taiwan independence(20), Tsai Ing-wen(12)
Topic2	space(4507), satellite(244), technology(9673), system(7348), country(10619), international(5937), research(6571), data(3845), utilize(4035), earth(1170)	battery(484), spacecraft(10), sun(1686), antenna(156), circuit(259), airship(121), optics(97), transducer(221), physics(120), satellite(244)
Topic3	children(2950), woman(1053), violence(490), committee(936), behavior(4218), family(3918), society(10239), government(6141), measure (1734), right(1565)	drugster(12), antenatal(35), cancer(676), diarrhea(310), teenager(617), girl(2160), sexual abuse(4), patriarch(2431), nonage(63), Zhu Lin(5)
Topic4	central government(1838), conference(2004), work(18347) people(4172), the Communist Party of China(436), National People's Congress(408), member(3986), today(8322), State Department(970), committee member(493)	Hebei province(993), vice-governor(40), Shenyang city(657), Public security bureau(384), deputy mayor(118), deputy secretary(106), deputy director general(191), Hupei(585), accept bribes(125)

5.1 LTE Topics

An informal but important measure of the success of the proposed model is the plausibility of the discovered search topics (Doyle and Elkan, 2009). Hence, we can qualitatively evaluate LTE through viewing its latent topics. An import feature of LTE is that it discovers latent topics from two perspectives. The first perspective is based on the Θ parameter, which corresponds to the Multinomial distributions over the vocabulary. The second perspective is based on the topic embedding and word embedding, i.e., the

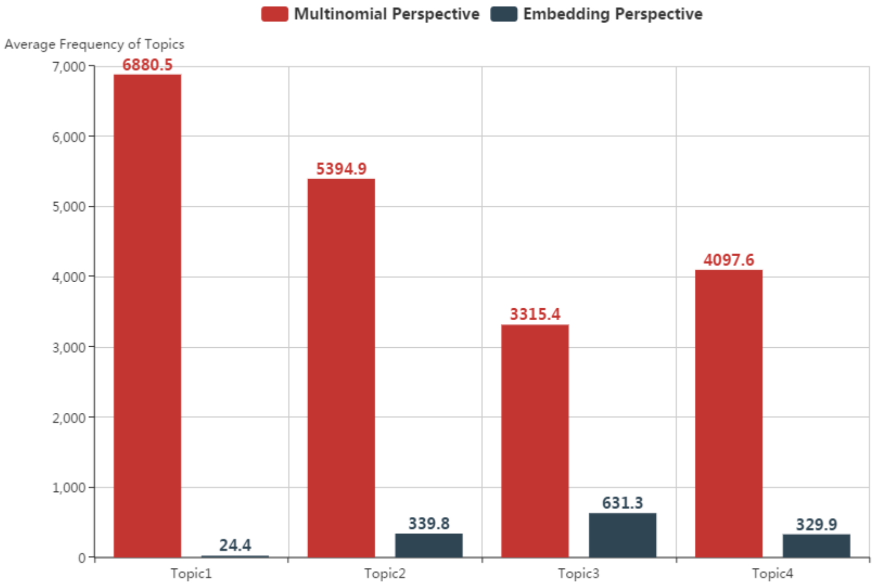


Figure 1: Average Word Frequency of the Two Perspectives

words whose embeddings have the highest cosine similarity with the topic embedding can be considered as the content of this topic. We utilize Web page dataset for the experiment. Some topic examples are presented in Table 1[1]. We observe that the words are semantically coherent in both of the two perspectives. For example, Topic 1 is about political issues between mainland China and Taiwan, Topic 2 is related to space technology, Topic 3 discusses the well-being of women and children and Topic 4 contains words about the political system of China. For each topic, the words from the two perspectives are semantically relevant and complimentary to each other.

An important insight is obtained from analyzing the frequencies of words in topics. The average word frequencies of the two perspectives are presented in Figure 1. We can see that word frequency of the second perspective is significantly smaller than that of the first perspective. For example, in Topic 1, the average word frequency of the first perspective is 6880.5 while that of the second perspective is only 24.4. This phenomenon shed light on an big advantage of LTE in text mining: bridging the semantic relevance between words with different frequencies. LTE overcomes the inherent problem of topic models that the topics are usually dominated by words of high frequency. By using the topic and word embeddings, we can effectively discover the semantics of words of relatively low frequency.

5.2 Perplexity Evaluation

We proceed to quantitatively compare LTE with LDA and the state-of-the-arts (i.e., Topical Word Embedding (TWE-1) (Liu et al., 2015) and Latent Feature-Dirichlet Multinomial Mixture (LFDMM) (Nguyen et al., 2015)) in terms of perplexity, which is a standard measure of evaluating the generalization performance of a probabilistic model (Rosen-Zvi et al., 2004). A lower perplexity indicates better generalization performance. A holdout dataset containing about ten thousand Web pages are utilized for perplexity evaluation. The result of perplexity comparison is presented in Figure 2. Since TWE-1 reuses the result of LDA, they have exactly the same performance in terms of perplexity. When varying the number of topics from 10 to 100, LTE always achieves the lowest perplexity, showing that generative process of LTE is a reasonable assumption for the data. An important observation is that LTE significantly outperforms LFDMM, showing that adding the sentence assumption and jointly utilizing word embedding and topic embedding to generate words result in better fit for the latent data structure of natural language documents. Perplexity is an indicator of the quality of the Multinomial topics. We observe that jointly

[1] The original Chinese words are translated into English to enhance readability.

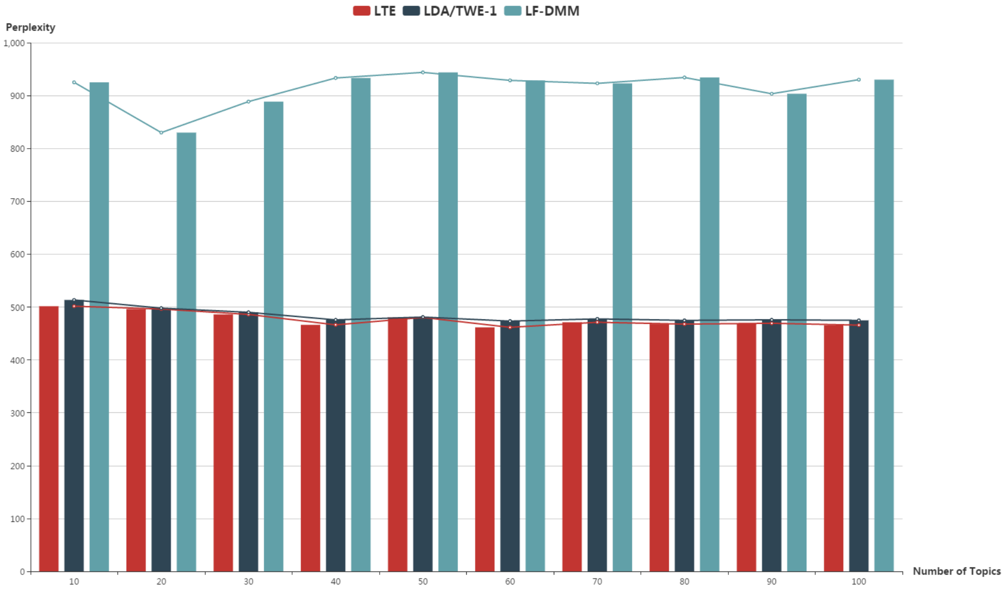


Figure 2: Perplexity on Holdout Data

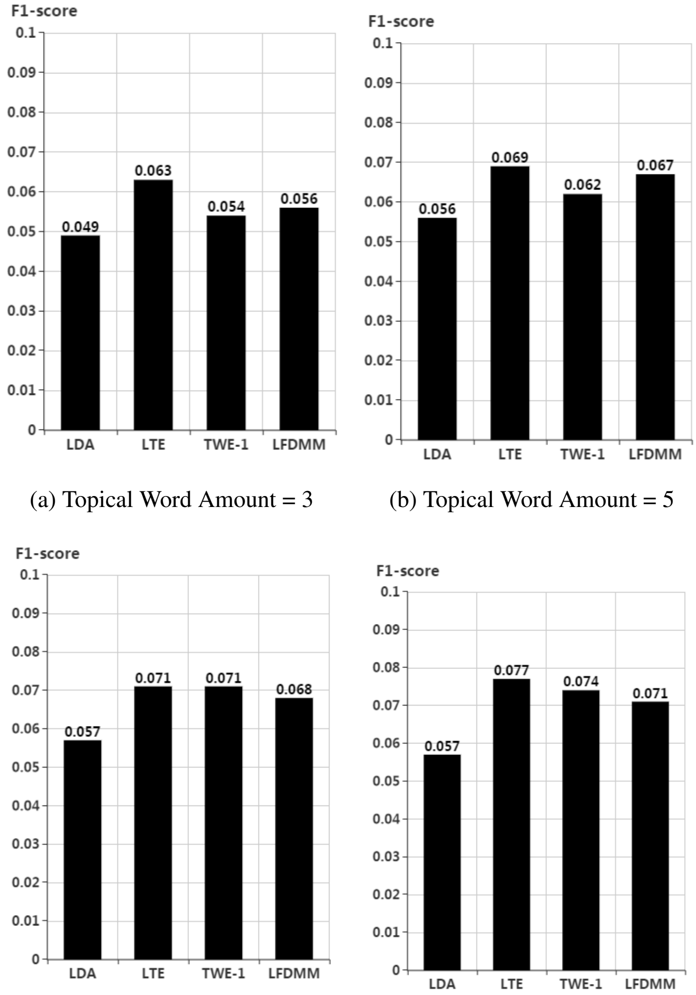
training of multinomial topics and embeddings does not harm the quality of the Multinomial topics. Rather, the joint training paradigm of LTE slightly improves the quality of the Multinomial topics. This observation verifies our assumption that the collectively utilizing co-occurrence information of different granularity has the potential of improving the performance of topic models.

5.3 Topical Word Extraction

We now evaluate the performance of LTE in the scenario of topical word extraction, which is critical for natural language understanding in modern search engines. Given a document, the goal of topical word extraction is to find some words that are highly relevant to the document theme. Conventionally, LDA plays an important role in topical word extraction (Zhao et al., 2011)(Pasquier, 2010). The existing methods based LDA are usually plagued by the weakness of capturing the semantics of words with low frequency. In this section, we study whether the embeddings generated by LTE are able to alleviate this problem. Ten thousands Web pages are utilized for this evaluation and the ground truth (i.e., the words that are highly relevant to the document theme) is manually prepared by human experts.

To derive the topical words for a document d, we first calculate the score of each word w in d and the score reflect the relevance between w and the themes of d. Then we sort all the words according to their scores and select the top-k words as the topical words of d. For TWE-1, LFDMM and LTE, the score of a word w is calculated based on embeddings by $score(w) = \sum_z P(z|d)\cos(v_w, v_z)$, where cos is the cosine similarity between two embeddings. As for LDA, we rely on the multinomial topics and calculate the score by $score(w) = \sum_z P(z|d)P(w|z)$. We compare the performance of these models in terms of $F1$ score, which is the harmonic mean of precision and recall.

The experimental result is shown in Figure 3. The models under-study tend to have higher $F1$ scores when the number of topical words increases. We observe that LDA always demonstrates the worst performance. The reason is that LDA is prone to select the frequent words and risks missing some words highly relevant to the document theme. In contrast, embedding information is less sensitive to the effect of word frequency. Therefore, TWE-1, LFDMM and LTE demonstrate better performance than LDA when the number of topical words varies from 3 to 10. LTE always demonstrates the highest $F1$ score. Comparing to TWE-1 and LFDMM which either reuse the output of LDA or Word2Vec, LTE jointly trains the Multinomial parameters and the embeddings, which are complimentary to each other and is effective to result in better topic modeling results and embeddings.



(a) Topical Word Amount = 3

(b) Topical Word Amount = 5

(c) Topical Word Amount = 7

(d) Topical Word Amount = 10

Figure 3: Topical Word Extraction

6 Conclusion

In this paper, we propose LTE to seamlessly integrate topic model and word embedding into one joint learning framework. We discuss a Monte Carlo EM algorithm for learning the parameter of LTE. LTE does not only output topic-related distributions but also generates distributed representation for words and latent topics. By applying LTE, we obtain coherent latent topics and the embedding generated by LTE are effective for identifying topical words of documents. Extensive experiments verify our assumption that topic modeling and word embedding are potentially complimentary for each other. While LTE is a specific model for off-the-shelf usage, the technique discussed in this paper can be easily transfer to many other scenarios where integrating other topic modeling and word embedding techniques are needed.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments and suggestions. This work is supported by National Basic Research Program of China (973 program No. 2014CB340505) .

References

David Bamman, Chris Dyer, and A. Noah Smith. 2014. Distributed representations of geographically situated language. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 828–834. Association for Computational Linguistics.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Gabriel Doyle and Charles Elkan. 2009. Accounting for burstiness in topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 281–288. ACM.

Chris Dyer. 2014. Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235.

Di Jiang, Kenneth Wai-Ting Leung, Wilfred Ng, and Hao Li. 2013. Beyond click graph: Topic modeling for search engine query log analysis. In *International Conference on Database Systems for Advanced Applications*, pages 209–223. Springer.

Yohan Jo and Alice H Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 815–824. ACM.

Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. 2009. Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems*, pages 61–68. ACM.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *ICML*.

Shaohua Li, Tat-Seng Chua, Jun Zhu, and Chunyan Miao. 2016. Generative topic embedding: a continuous representation of documents. In *Proceedings of The 54th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Andriy Mnih and Yee Whye Teh. 2012. A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2009. Distributed algorithms for topic models. *The Journal of Machine Learning Research*, 10:1801–1828.

Dat Quoc Nguyen, Richard Billingsley, Lan Du, and Mark Johnson. 2015. Improving topic models with latent feature word representations. *Transactions of the Association for Computational Linguistics*, 3:299–313.

Claude Pasquier. 2010. Task 5: Single document keyphrase extraction using sentence clustering and latent dirichlet allocation. In *Proceedings of the 5th international workshop on semantic evaluation*, pages 154–157. Association for Computational Linguistics.

Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. 2004. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press.

Hanna M Wallach. 2006. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM.

Xuerui Wang and Andrew McCallum. 2006. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM.

Yi Wang, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y Chang. 2009. Plda: Parallel latent dirichlet allocation for large-scale applications. In *Algorithmic Aspects in Information and Management*, pages 301–314. Springer.

Gu Xu, Shuang-Hong Yang, and Hang Li. 2009. Named entity mining from click-through data using weakly supervised latent dirichlet allocation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1365–1374. ACM.

Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, Ee-Peng Lim, and Xiaoming Li. 2011. Topical keyphrase extraction from twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 379–388. Association for Computational Linguistics.