Abstract

Latent Dirichlet Allocation (LDA) mining thematic structure of documents plays an important role in natural language processing and machine learning areas. However, the probability distribution from LDA only describes the statistical relationship of occurrences in the corpus and usually in practice, probability is not the best choice for feature representations. Recently, embedding methods have been proposed to represent words and documents by learning essential concepts and representations, such as Word2Vec and Doc2Vec. The embedded representations have shown more effectiveness than LDA-style representations in many tasks. In this paper, we propose the Topic2Vec approach which can learn topic representations in the same semantic vector space with words, as an alternative to probability. The experimental results show that Topic2Vec achieves interesting and meaningful results.

1 Introduction

Modeling text (words, topics and documents) is a key problem in natural language processing (NLP) and information retrieval (IR). The goal is to find short and essential descriptions which enable efficient processing of large systems and benefit basic tasks such as classification, clustering, summarization and estimation of similarity or relevance.

During the past decades, various models and solutions are proposed, such as Bag-of-Words (BOW) (Harris, 1954), TF-IDF (Salton and McGill, 1983), Latent Semantic Analysis (LSA) (Landauer et al., 1998) and Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 1999). But the best-known model is Latent Dirichlet Allocation (LDA) (Blei et al., 2003) which describes the hierarchical relationships between words, topics and documents. In LDA, documents are represented as probability distributions over latent topics where each topic is characterized by a distribution over words. However, the probability distribution generated from LDA prefers to describe the statistical relationship of occurrences rather than real semantic information embedded in words, topics and documents. Also LDA will assign high probabilities to high frequency words and those words with low probabilities are hard to be chosen as representatives of topics. But in practice, low probability words sometimes distinguish topics better. For example, LDA will assign higher probability and choose “food” as representative other than “cheeseburger”, “drug” other than “aricept” and “technology” other than “smartphone”.

Recently, distributed representations with neural probabilistic language models (NPLMs) (Bengio et al., 2003) were proposed to represent words and documents as low-dimensional vectors in one semantic space, and achieved significant results in many NLP and ML tasks (Collobert and Weston, 2008; Mnih and Hinton, 2009; Mikolov et al., 2013a; Mnih and Kavukcuoglu, 2013; Huang et al., 2012; Le and Mikolov, 2014). In particular, Word2Vec proposed by Mikolov et al. (2013a) could automatically learn concepts and semantic-syntactic relationships between words like vec(“Berlin”) - vec(“Germany”) = vec(“Paris”) - vec(“France”). Doc2Vec (Para2Vec) proposed by Le and Mikolov (2014) achieves state-of-the-art performance on sentiment analysis. Naturally, in this paper, we want to answer the question that, what will happen if we embed topics in the semantic vector space?

Following the ideas of previously proposed models for words and documents, we propose the model Topic2Vec as shown in Fig. 1. Based on the Word2Vec, we incorporates topics into the NPLM framework for learning distributed representations.
of topics in the same semantic space with words. Furthermore, words and topics naturally can estimate similarity and relevance with each other such as using cosine function rather than using probability.

In the experiments, we evaluate two different topic representations including embedding of Topic2Vec and probability of LDA in two aspects: listed examples and t-SNE 2D embedding of nearest words for each topic. The experimental results show that our Topic2Vec achieves distinctive and meaningful results compared to LDA.

2 Related Models

2.1 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) (Blei et al., 2003) is a probabilistic generative model that assumes each document is a mixture of latent topics, where each topic is a probability distribution over all words in vocabulary. Briefly, LDA generates a sequence of words as follows:

- For each of the $N$ word $w_n$ in document $d$:
  - Sample a topic $z_n \sim \text{Multinomial} (\theta_d)$
  - Sample a word $w_n \sim \text{Multinomial} (\phi_{z_n})$.

By Gibbs Sampling estimation, we obtain document-topic probability matrix $\Theta$ and topic-word probability matrix $\Phi$. For a new document of arbitrary length, we can infer its involved latent topics and meanwhile we will assign a topic label for each word in the document.

2.2 Word2Vec

Inspired by Neural Probabilistic Language Model (NPLM) (Bengio et al., 2003), Mikolov et al. (2013a) proposed Word2Vec including CBOW and Skip-gram for computing continuous vector representations of words from large data sets.

When training, given a word sequence $D = \{w_1, \ldots, w_M\}$, the learning objective functions are defined to maximize the following log-likelihoods, based on CBOW and Skip-gram, respectively.

$$L_{\text{CBOW}}(D) = \frac{1}{M} \sum_{i=1}^{M} \log p(w_i|w_{\text{ctx}}), \quad (1a)$$

$$L_{\text{Skip-gram}}(D) = \frac{1}{M} \sum_{i=1}^{M} \sum_{-k \leq c \leq k, c \neq 0} \log p(w_{i+c}|w_i), \quad (1b)$$

Here, in Equation (1a), $w_{\text{ctx}}$ indicates the context of the current word $w_i$. In Equation (1b), $k$ is the window size of context. For any variables $w_j$ and $w_i$, the conditional probability $p(w_j|w_i)$ is calculated using softmax function as follows,

$$p(w_j|w_i) = \frac{\exp(w_j \cdot w_i)}{\sum_{w \in W} \exp(w \cdot w_i)}, \quad (2)$$

where $w$, $w_i$ and $w_j$ are respectively the word representations of word $w$, $w_i$ and $w_j$, $W$ is the word vocabulary.

3 Topic2Vec

Inspired by word2vec, we incorporate topics and words into the NPLM. We propose Topic2Vec as shown in Fig. 1 for learning distributed topic representations together with word representations. Topic2Vec is also separated in CBOW and Skip-gram situations. For instance, given a word sequence $(w_{t-2}, w_{t-1}, w_t, w_{t+1}, w_{t+2})$, in which $w_t$ is the current word assigned with topic $z_t$ by LDA. The CBOW predicts the word $w_t$ and topic $z_t$ based on the surrounding words $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$, while the Skip-gram predicts surrounding words $(w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2})$ given current $w_t$ and $z_t$.

When training, given a word-topic sequence of a document $D = \{w_1 : z_1, \ldots, w_M : z_M\}$, where $z_i$ is the word $w_i$’s topic inferred from LDA, the learning objective functions can be defined to maximize the following log-likelihoods, based on CBOW and Skip-gram, respectively.

$$L_{\text{CBOW}}(D) = \frac{1}{M} \sum_{i=1}^{M} \left( \log p(w_i|w_{\text{ctx}}) + \log p(z_i|w_{\text{ctx}}) \right),$$
The complexity of our Topic2Vec is linear with size of vocabulary. The algorithm are used to optimize our model. By the way, we use Negative Sampling without Hierarchical Softmax. Stochastic gradient descent (SGD) and back-propagation algorithm are used to optimize our model. By the way, the simplicity and efficient solution, we just follow the optimization scheme that used in Word2Vec 

\[ \mathcal{L}_{\text{Skip-gram}}(D) = \frac{1}{M} \sum_{i=1}^{M} \sum_{-k \leq c \leq k, c \neq 0} \log p(w_{i+c} | w_i) + \log p(w_{i+c} | z_i). \] 

Topic2Vec aims at learning topic representations along with word representations. Considering the simplicity and efficient solution, we just follow the optimization scheme that used in Word2Vec (Mikolov et al., 2013a). To approximately maximize the probability of the softmax, we use Negative Sampling without Hierarchical Softmax (Mikolov et al., 2013b). Stochastic gradient descent (SGD) and back-propagation algorithm are used to optimize our model. By the way, complexity of our Topic2Vec is linear with size of dataset, same with Word2Vec.

**Figure 2:** Nearest words and topics for each topic. Words are listed with conditional probabilities in LDA while words and topics are listed with calculated cosine similarity in Topic2Vec.

\[ \text{Topic}_6 \quad \text{Topic}_19 \quad \text{Topic}_27 \quad \text{Topic}_47 \]

| word | prob. | word | prob. | word | prob. | word | prob. |
|------|-------|------|-------|------|-------|------|-------|
| food | 0.027 | drug | 0.031 | medical | 0.033 | dog | 0.011 |
| restaurant | 0.008 | eats | 0.019 | hospital | 0.014 | puke | 0.009 |
| eat | 0.008 | cancer | 0.019 | care | 0.019 | tree | 0.009 |
| more | 0.005 | study | 0.011 | patients | 0.018 | dogs | 0.009 |
| chicken | 0.005 | patients | 0.011 | doctors | 0.016 | plants | 0.008 |
| cooking | 0.005 | treatment | 0.009 | health | 0.013 | trees | 0.008 |
| eating | 0.005 | bed | 0.009 | doctor | 0.009 | animal | 0.007 |
| one | 0.005 | heart | 0.008 | patient | 0.009 | plant | 0.007 |
| good | 0.005 | risk | 0.008 | surgery | 0.008 | animals | 0.006 |
| foods | 0.005 | more | 0.007 | center | 0.008 | zoo | 0.006 |

- **LDA**
- **Topic2Vec**
- **Figure 2**: Nearest words and topics for each topic. Words are listed with conditional probabilities in LDA while words and topics are listed with calculated cosine similarity in Topic2Vec.

**4 Experiments**

**4.1 Dataset**

We use the English Gigaword Fifth Edition as our training data for learning fundamental word and topic representations. We randomly extract part of documents and construct our training set described as follows: we chose 100,000 documents, where each consists of more than 1,000 characters from subfolder itw_eng (Los Angeles Times) containing 411,032 documents. Besides, we eliminate those words that occur less than 5 times and the stop words. In the end, training set contains about 42 million words and the vocabulary size is 102,644.

**4.2 Evaluation Methods**

In experiments, we run Topic2Vec in Skip-gram and learn topic representations together with word representations. And then we evaluate topic repre-
sentations via comparing Topic2Vec with LDA in two aspects: (1) we select most related topics or words conditioned on selected topics and (2) we embed these related words or topics in 2D space using t-SNE (Maaten et al., 2008). During the process, we cluster words into topics as follows:

- LDA: each topic is a probability distribution over words. We select the top $N = 10$ words with highest conditional probability.
- Topic2Vec: topics and words are equally represented as the low-dimensional vectors, we can immediately calculate the cosine similarity between words and topics. For each topic, we select higher similarity words.

4.3 Analysis of Results

Fig. 2 shows top 10 nearest words from LDA and Topic2Vec for eight typically selected topics, respectively. We now give more detailed analysis to understand the difference between them. As shown in Fig. 2, in Topic 19, LDA returns the words like “drug”, “drugs”, “cancer” and “patients”, while Topic2Vec returns “aricept”, “memantine”, “enbrel” and “gabapentin”. In Topic 27, LDA returns the words of “medical”, “hospital”, “care”, “patients” and “doctors”, while Topic2Vec returns “neonatal”, “anesthesiologists”, “anesthesia” and “comatose”. We only know that Topic 19 and Topic 27 share the same topic about “patients” or “medical”, but we can’t get their further difference from the results of LDA. But from the result of Topic2Vec, we can easily discover that Topic 19 focuses on a more specific topic about drugs (“aricept”, “memantine”, “enbrel” and “gabapentin”), while Topic 27 focuses on another specific topic about treatment (“anesthesiologists”, “anesthesia” and “comatose”), they are absolutely different. Obviously, Topic2Vec presents more distinguished results between two similar topics.

Fig. 3 shows the 2D embedding of the corresponding related words for each topic by using t-SNE. Obviously, Topic2Vec produces a better grouping and separation of the words in different topics. In contrast, LDA does not produce a well separated embedding, and words in different topics tend to mix together.

In summary, for each topic, words selected by Topic2Vec are more typical and representative compared to those returned by LDA. Eventually, Topic2Vec can better distinguish different topics.

5 Conclusions and Future Work

In this paper, via integrating NPLM, Word2Vec and LDA, we propose the Topic2Vec which successfully embeds latent topics in the same semantic vector space with words. In principle, our purpose clearly aims at learning new fashion topic representation by Topic2Vec. From the observation of experiments, Topic2Vec presents more distinguished results than LDA and we have the conclusion that Topic2Vec can model topics better. But now, we just qualitatively evaluate the performance of Topic2Vec and LDA, we will quantitatively do more detailed analysis about their difference in the future. Besides, we have to run LDA firstly to assign a topic for each word in the corpus before Topic2Vec. We also will explore new independent topic models which can mine thematic structure of documents as LDA and learn inherent representations and model topics better as
Topic2Vec, simultaneously.

Acknowledgments

References

[Bengio et al.2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent and Christian Jauvin. 2003. A neural probabilistic language model. Journal of Machine Learning Research, 3:1137-1155.

[Collobert and Weston2008] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, ICML ’08, pages 160-167, New York, NY, USA. ACM.

[Landauer et al.1998] Thomas K. Landauer, Peter W. Foltz, and Darrell Laham. 1998. An introduction to latent semantic analysis. Discourse processes 25.2-3 (1998): 259-284.

[Blei et al.2003] David M. Blei, Andrew Y. Ng and Michael I. Jordan. 2003. Latent dirichlet allocation. Journal of Machine Learning Research, 3:993-1022.

[Mikolov et al.2013a] Tomas Mikolov, Kai Chen, Gerg Corrado and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781.

[Mikolov et al.2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26, pages 3111-3119.

[Mikolov et al.2013c] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013c. Linguistic regularities in continuous space word representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 746-751, Atlanta, Georgia, June. Association for Computational Linguistics.

[Le and Mikolov2014] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In Proceedings of the 31th International Conference on Machine Learning, Beijing, China.

[Socher et al.2011] Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng and Christopher D. Manning. 2011a. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In Advances in Neural Information Processing Systems 24.

[Socher et al.2011] Richard Socher, Cliff C. Lin, Andrew Y. Ng and Christopher D. Manning. 2011b. Parsing natural scenes and natural language with recursive neural network. In Proceedings of the 26th International Conference on Machine Learning.

[Huang et al.2012] Eric H. Huang, Richard Socher, Christopher D. Manning and Andrew Y. Ng. 2012 Improving word representations via global context and multiple word prototypes. In Annual Meeting of the Association for Computational Linguistics.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In Conference on Empirical Methods in Natural Language Processing.

[Harris1954] Zellig S. Harris. 1954. Distributional structure. Word.

[Mass et al.2011] Andrew L. Mass, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics.

[Sebastiani2002] Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. ACM Computing Survey, 34(1): 1-47.

[Yogatama et al.2014] Dani Yogatama, Manaal Faruqui, Chris Dyer and Noah A. Smith. 2014. Learning word embeddings efficiently with noise-contrastive estimation. In Advances in Neural Information Processing Systems 26 (NIPS 2013).

[Mnih and Kavukcuoglu2013] Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In Advances in Neural Information Processing Systems 26, pages 1081-1088.

[Mnih and Hinton2009] Andriy Mnih and Geoffrey E. Hinton. 2009. A scalable hierarchical distributed language model. In Advances in Neural Information Processing Systems, pages 1081-1088.

[Morin and Bengio2005] Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In Proceedings of the International Workshop on Artificial Intelligence and Statistics, pages 246-252.

[Turnier et al.2010] Joseph Turian, Lev Ratinov and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 384-394. Association for Computational Linguistics.

[Turney and Pantel2010] Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research.

[Gildea and Hofmann1999] Daniel Gildea and Thomas Hofmann. 1999. Topic-based language models using EM. In Proceedings of EUROSPEECH, pages 2167-2170.
