Automatic Labelling of Topic Models using Word Vectors and Letter Trigram Vectors

Abstract. The native representation of LDA-style topics is a multinomial distributions over words, but automatic labelling of such topics has been shown to help readers interpret the topics better. We propose a novel framework for topic labelling using word vectors and letter trigram vectors. We generate labels automatically and propose automatic and human evaluations of our method. First, we use a chunk parser to generate candidate labels, then map topics and candidate labels to word vectors and letter trigram vectors in order to find which candidate label are more semantically related with that topic. A label can be found by calculating the similarity between a topic and its candidate label vectors. Experiments on three data sets show that the method is effective.

Keywords. Topic Labelling. Word Vectors. Letter Trigram Vectors.

1 Introduction

Topic models have been widely used in tasks like information retrieval, text summarization, word sense discrimination and sentiment analysis. Popular topic models include Mixture of unigrams [1], Probabilistic Latent Semantic Indexing [2] and Latent Dirichlet Allocation (LDA) [3].

Topics in topic models are usually represented as word distributions, e.g. via the top-10 words of highest probability in a given topic. For example, the multinomial word distribution *'feed contaminated farms company eggs animal food dioxin authorities german'* is a topic extracted from a collection of news articles. The model gives high probabilities to those words like *feed*, *contaminated*, and *farms*. This topic refers to an animal food contamination incident. Our research aims to generate topic labels to make LDA topics more readily interpretable.

A good topic label has to satisfy the following requirements: (1) it should capture the meaning of a topic; (2) it should be easy for people to understand. There are many ways to represent a topic, such as a list of words, a single word or phrase, or a sentence or paragraph. A word can be too general in meaning while a sentence or a paragraph can be too detailed to capture a topic. In this research, we select phrases to represent topics.

Our method consists of three steps. First, we generate candidate topic labels, then map topics and candidate labels to vectors in a vector space. Finally by calculating and comparing the similarity between a topic and its candidate label vectors, we can find a topic label for each topic.

Our contributions in this work are: (1) the proposal of a method for generating and scoring labels for topics; and (2) the proposal of a method using two word vector models and a letter trigram vector model for topic labelling. In experiments over three corpora, we demonstrate the effectiveness of our method.

2 Related work

Topics are usually represented by their top-N words. For example, Blei [3] simply use words ranked by their marginal probabilities p(w|z) in an LDA topic model. Lau use features including PMI, WordNet-derived similarities and Wikipedia features to re-rank the words in a topic, and select the top three words as their topic label [4]. A word representation can be hard for humans to understand. Some other methods use human annotation [5, 6], with obvious disadvantages: on the one hand the result is influenced by subjective factors, and on the other hand, it is not an automatic method and is hard to replicate.

Some use feature-based methods to extract phrases as topic labels. Lau [7] proposed a method that is based on: (1) querying Wikipedia using the top-N topic words, and extracting chunks from the titles of those articles; (2) using RACO to select candidate labels from title chunks; and (3) ranking candidate labels according to features like PMI and the student's t test, and selecting the top-rank label as the final result.

Blei and Lafferty [8] used multi-word expressions to visualize topics, by first training an LDA topic model and annotating each word in corpus with its most likely topic, then running hypothesis testing over the annotated corpus to identify words in the left or right of word or phrase with a given topic. The hypothesis testing is run recursively. Topics are then represented with multi-word expression.

Recent work has applied summarization methods to generate topic labels [9]. Cano et al. proposed a novel method for topic labelling that runs summarization algorithms over documents relating to a topic. Four summarization algorithms are tested: Sum basic, Hybrid TFIDF, Maximal marginal relevance and TextRank. The method shows that summarization algorithms which are independent of the external corpus can be applied to generate good topic labels.

Vector based methods have also been applied to the topic labelling task. Mei [10] developed a metric to measure the "semantic distance" between a phrase and a topic model. The method represents phrase labels as word distributions, and approaches the labelling problem as an optimization problem that minimize the distance between the topic word distribution and label word distribution.

Our technique is inspired by the vector based method [10] and work in learning vector representations of words using neural networks [11, 12, 13]. The basic intuition is that a topic and a label are semantically related in a semantic vector space [14].

3 Methodology

3.1 Preliminary

Distributional vectors can be used to model word meaning to capture correlations between words [14]. In order to capture correlations between a topic and a label, we map LDA topics and candidate labels to a vector space, and calculate the similarity between pairs of topic vectors and candidate label vectors. A candidate label which has the highest similarity is chosen as the label for that topic.

Table 1. Symbol description.

Symbol	Description
z	A topic
T	Number of topic in corpus
ϕ_z	word distribution of topic z
w	A word
d	A document
θ_d	topic distribution of d
l	A label
Lz	A set of candidate labels of topic z
S	A Letter trigram set
D	A Document set
V	Vocabulary Size
Sim	Similarity measure
y	A vector
GS	Gold standard

The framework of our method is shown in Figure 1. Note that $l_1 \ldots l_n$ represent candidate labels of topic *z* and *Sim* represents the similarity between two vectors. The symbols used in this paper are detailed in Table 1.

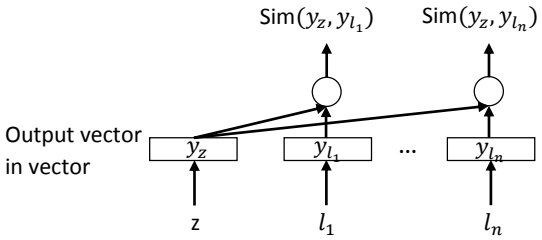

Fig. 1. Outline of our method.

We apply three kinds of vectors: letter trigram vectors from [15], and two word vectors: CBOW (continuous bag-of-words model) and Skip-gram [13]. A letter trigram vector is able to capture morphological variants of the same word to points that are close to each other in a letter trigram vector space. CBOW and Skip-gram are two deep neural vector models. They are able to capture latent features from a corpus.

3.2 Candidate Label Extraction

We first identify topic-related document sets according to a topic summarization method [9]. The predominant topic of a document d can be calculated by

$$z_d = argmax\ \theta_d(z) \tag{1}$$

Given a topic z, the set of documents D_z whose predominant topic is z is then simply the set of documents that have z as their predominant topic. For each topic z, we then use OpenNLP[1] to extract chunks that contain at least two LDA top-10 words from D_z, as candidate labels l_z.

3.3 Vector Generation

CBOW Vector.

CBOW generates continuous distributed representations of words from their context of use. The model builds a log-linear classifier with bi-directional context words as input, where the training criterion is to correctly classify the current (middle) word. It captures the latent document features and has been shown to perform well over shallow syntactic relation classification tasks [13]. The framework of CBOW model is illustrated in Figure 2.

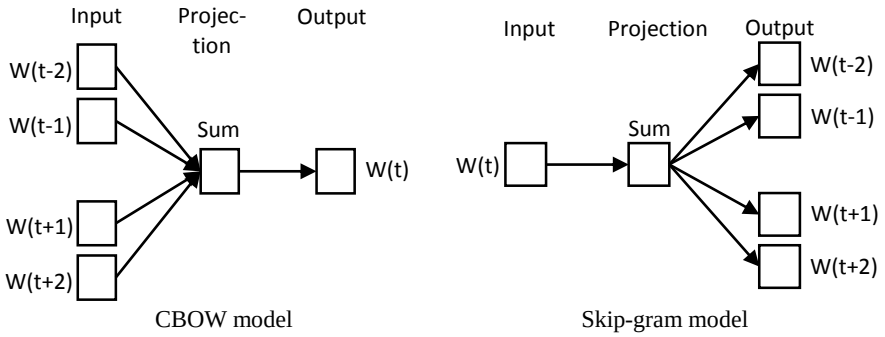

Fig. 2. CBOW and Skip-gram model.

Recent research has extended the model to go beyond the word level to capture phrase- or sentence-level representations [16, 17, 18, 13]. We simply apply the weighted additive method [16] to generate a phrase vector. Word vectors of candidate label and LDA topic are generated as follows:

$$y_l^{cbow} = \sum_{w_j \in l} y_{w_j}^{cbow} \tag{2}$$

$$y_z^{cbow} = \sum_{w_j \in z} y_{w_j}^{cbow} * \phi_z(w_j) \tag{3}$$

where $y_{w_j}^{cbow}$ is the word vector of CBOW.

Skip-gram Vector.

[1] http://opennlp.apache.org/

The Skip-gram model [13] is similar to CBOW, but instead of predicting the current word based on bidirectional context, it uses each word as an input to a log-linear classifier with a continuous projection layer, and predicts the bidirectional context. The framework of the Skip-gram model is illustrated in Figure 2.

The Skip-gram model can capture latent document features and has been shown to perform well over semantic relation classification tasks [13].

Phrase vectors are, once again, generated using the weighted additive method [16]. Skip-gram vectors of candidate labels and LDA topics are generated as follows:

$$y_l^{skip} = \sum_{w_j \in l} y_{w_j}^{skip} \tag{4}$$

$$y_z^{skip} = \sum_{w_j \in z} y_{w_j}^{skip} * \phi_z(w_j) \tag{5}$$

where $y_{w_j}^{skip}$ is the word vector of Skip-gram model.

Letter Trigram Vectors.

Based on the method of [15], we use letter trigram vectors to represent a topic and its candidate labels. Each dimension in a letter trigram vector represents a letter trigram (e.g. abc, acd). We generate a letter trigram set S_l for each phrase l. A letter trigram set is defined as the set of letter trigrams from the phrase. For example, the letter trigram set of the phrase stock market is {'^st', 'sto', 'toc', 'ock', 'ck ', ' ma', 'mar', 'ark', 'rke', 'ket', 'et$'}. For each dimension i in letter trigram vector of phrase l, we assign a weight based on the following:

$$y_l^{letter\ trigram}(i) = \begin{cases} \frac{occurrence_i}{\sum occurrence_j}, if\ letter\ trigram\ i\ is\ in\ S_l \\ 0, otherwise \end{cases} \tag{6}$$

where occurrence is the number of letter trigrams i and j in S_l.

Using a similar method, we generate the letter trigram set S_z of the top-10 LDA words, for each letter trigram i in the top-10 words, and assign a weight as follows:

$$y_z^{letter\ trigram}(i) = \begin{cases} \sum_{w_j\ that\ contains\ i} \phi_z(w_j), if\ letter\ trigram\ i\ is\ in\ S_z \\ 0, otherwise \end{cases} \tag{7}$$

where w_j is the topic word that contains letter trigram i. We further normalize the letter trigram vector as follows:

$$y_z^{letter\ trigram\ normalized}(i) = \frac{y_z^{letter\ trigram}(i)}{\sum_j y_z^{letter\ trigram}(j)} \tag{8}$$

3.4 Topic Label Selection

After generating vectors for candidate labels and LDA topics, we then calculate the similarity between them. The candidate label with the highest similarity with a topic is selected as the topic label according to the following formula:

$$l_z = argmax_{l \in L_z} Sim(y_l, y_z) \tag{9}$$

where

$$Sim(y_l, y_z) = \text{cosine}(y_l, y_z) = \frac{{y_l}^T y_z}{\|y_l\| \|y_z\|} \tag{10}$$

4 Experiments

4.1 Dataset & Gold standard

Table 2. Dataset.

Corpus	#Document	#Topic	#Topic eliminated
News-30	3743	30	2
News-40	3743	40	6
News-50	3743	50	11
Twitter-30	35815	30	19
Twitter-40	35815	40	30
Twitter-50	35815	50	40
NIPS-30	2075	30	5
NIPS-40	2075	40	8
NIPS-50	2075	50	20

We use three corpora in our experiments: (1) News, (2) Twitter, and (3) NIPS. The News and Twitter corpora are described in [9]. News was collected between November 2010 and January 2011 from traditional News media (BBC, CNN, and New York Times) and comprises 3744 articles across three categories (War, Education and Law). The Twitter corpus was collected between November 2010 and January 2011 and contains 35815 tweets in three categories (War, Education and Law). The NIPS corpus is a collection of NIPS abstracts from 2001 to 2010. The News corpus contains about 0.61M words, the Twitter corpus contains about 0.37M words, the NIPS corpus contains about 0.44M words.

The LDA training parameter α is set to 50/T and β is set to 0.01. We test the effect of the topic labelling method when T (the number of topics) is set to 30, 40 and 50 for each corpus. We use an entropy-based method to filter meaningless topics. The entropy of topic z can be calculated by the following formula.

$$\text{Entropy(z)} = -\sum_{i=1}^{V} \phi_z(w_i) \log(\phi_z(w_i)) \tag{11}$$

In the News and Twitter corpora, those topics whose entropy is higher than 0.9 are eliminated, and in the NIPS corpus, topics whose entropy is higher than 1.4 are eliminated; these thresholds were set based on manual analysis of a handful of topics for each document collection. For the Twitter corpus, we further filter topics which lack a

meaningful gold standard label. The method for generating gold standard labels will be illustrated later in this section. Table 2 shows detailed information of our dataset.

Yang [19] indicates that gold standard labels from human beings suffer from inconsistency. The inter-annotator F-measure of human annotators is 70-80%. Therefore we develop an automatic method to generate gold standard labels to evaluate the proposed method: for each topic z, we extract chunks from titles in D_z, assign a weight to each chunk according to the word frequency in that chunk, and select the chunk that has the highest weight as the label (GS) for that topic. Our underlying motivation in this is that each headline is the main focus of a document. A phrase from a title can be a good representation of a document. Therefore a phrase from a title can be a good label for that topic.

Note that the News and NIPS corpora have titles for each document, while the Twitter corpus has not title information. The gold standard for the News and NIPS corpora were thus generated automatically, while for the Twitter corpus — which was collected over the same period of time as the News corpus — we apply the following method [9]: first calculate the cosine similarity between each pair of Twitter and News topic word distributions; then for each Twitter topic i, select the News topic j that has the highest cosine similarity with i and where the similarity score is greater than a threshold (0.3 in this paper). The label (GS) of News topic j is then regarded as the gold standard (GS) of Twitter topic i.

4.2 Evaluation Metrics

We evaluate our results automatically and via human evaluation.

Automatic Evaluation Method.

Because of synonym and polysemy, we can't compare the generated label directly with the GS automatically. Rather, we propose to generate an evaluation score as follows:

$$score_z = \frac{\sum_{w \in GS} max_{w' \in l} Lin'(w,w') + \sum_{w' \in l} max_{w \in GS} Lin'(w,w')}{\#words(GS) + \#words(l)} \tag{12}$$

$$\text{Lin}'(\text{w}, \text{w}') = \begin{cases} 1, if\ stem(w) = stem(w') \\ Lin(w, w') \end{cases} \tag{13}$$

Where Lin is the word similarity based on WordNet2 [20]. GS and l represent the gold standard and the label generated for topic z. The Porter stemmer3 is used to stem all words. The score is used to measure the semantic similarity between a generated label and a GS.

Human Evaluation Method.

[2] http://nlp.shef.ac.uk/result/software.html

[3] http://tartarus.org/~martin/PorterStemmer/

We also had six human annotators manually score the extracted labels. Each annotator is presented with the top-10 LDA words that describe a topic, the gold standard label, and a series of extracted labels using the methods described in Section 3. They then score each extracted label as follows: 3 for a very good label; 2 for a reasonable label, which does not completely capture the topic; 1 for a label semantically related to the topic, but which is not a good topic label; and 0 for a label which is completely inappropriate and unrelated to the topic.

4.3 Baseline Methods

LDA Top 1 . Simply select the top-1 topic word as the topic label.

First Order. This method is proposed by [10]. It generates a word vector of candidate labels according to PMI values in the original corpus. In this paper, the first-order vector is used in our vector based method shown in Figure 1.

4.4 Experimental Results

The word2vec[4] toolbox was used to train the CBOW and Skip-gram models. The window size was set to 5 for both models. We experimented with word vectors of varying dimensions; the results are shown in Figure 3, based on automatic evaluation. When the number of dimensions is 100, the result is the best on average, and this is the size we use for both CBOW and Skip-gram throughout our experiments. The dimension of the letter trigram vector is 18252.

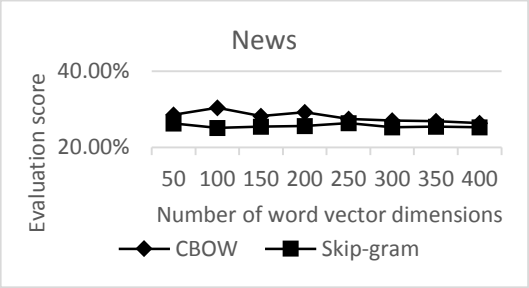

Fig. 3. Automatic evaluation results for word vectors over the News corpus for different dimension sizes.

Figure 4 shows the automatic evaluation results for topic labelling with different numbers of topics. We can see that the result varies with the number of topics. When the topic number T is 50, the score for the News and NIPS corpora is the highest; and when the topic number is 40, the score for the Twitter corpus is highest.

[4] https://github.com/NLPchina/Word2VEC_java

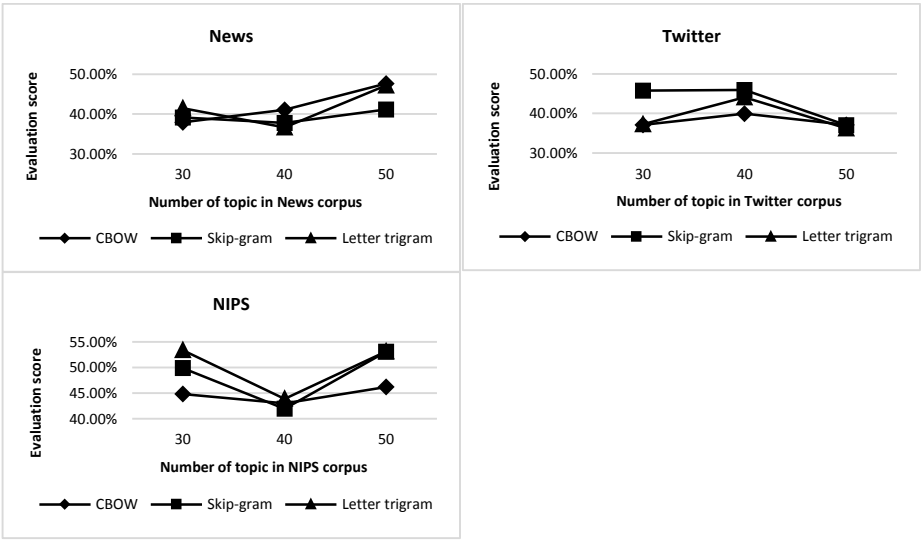

Fig. 4. Automatic evaluation results.

Table 3 shows the result between our methods and baseline methods in News-50, Twitter-40 and NIPS-50 corpora.

Table 3. Evaluation scores of three kinds of vectors vs. baselines.

	Automatic evaluation scores			**Human evaluation scores**		
Method	**News-50 corpus score**	**Twitter-40 corpus score**	**NIPS-50 corpus score**	**News-50 corpus score**	**Twitter-40 corpus score**	**NIPS-50 corpus score**
CBOW	**47.64**%	39.94%	46.12%	**1.94**	1.10	2.10
Skip-gram	41.15%	**45.90%**	53.05%	1.85	**1.60**	2.07
Letter tri-gram	47.17%	44.08%	**53.14%**	1.86	1.50	**2.13**
LDA top 1	33.73%	32.65%	41.62%	1.04	1.00	1.02
First order	42.77%	41.90%	42.74%	1.60	**1.60**	1.70

Based on the experimental results in Table 3, we summarize as follows:

1. Most methods perform better over NIPS than News and Twitter. The reason is that we use NIPS abstracts (and not full papers) to train the LDA topic model. Abstracts are more closely related to the paper titles. This means that automatically-generated gold standard labels are more likely to score well for NIPS.
2. The Skip-gram model performs much better than CBOW over Twitter and NIPS, while over News, CBOW is better than Skip-gram. The reason is that Skip-gram model outperforms CBOW of the same dimension in data sparse corpus like NIPS and Twitter. Mikolov [13] shows that the Skip-gram model performs better over

semantic tasks while CBOW performs better over shallow syntactic tasks, based on which we assumed that the Skip-gram model should be better for topic labelling. However, our experiments indicate that results are also dependent on the genre of the corpus: News topics usually refer to concrete information like the people and the action of a certain event; NIPS topics, on the other hand usually refer to scientific concepts, while Twitter topics are more comments on certain events, and informal and brief.

3. The letter trigram vectors perform surprisingly consistently over the three corpora in Table 3. Letter trigrams simply capture character features of a word, and the method is therefore not dependent on the genre of a corpus. Compared with the first order vector, letter trigram vectors have reduced dimensionality, and are able to capture morphological variations of the same word to points that are close to each other in the letter trigram space.

The three methods proposed in this paper are all better than LDA top-1 word baseline. The reason might be that in this method, we compare the top-1 word with a phrase (GS). Our methods are also better than the first-order baseline in most cases. Our result shows that trigram vectors are more suitable for topic labelling over different types of corpus. The Skip-gram model is better than CBOW for the Twitter and NIPS corpora, while the CBOW model is more suitable for the News corpus.

Table 3 also shows the results for human evaluation. We summarize the results as follows:

1. Similar to the automatic evaluation results, the score over NIPS is higher than the other two corpora. The score for News is higher than the score for Twitter. Under human evaluation, labels generated using vector-based methods are on average reasonable labels for NIPS, and nearly reasonable labels for News. Even for a corpus without title information like Twitter, it can extract related topic labels.
2. Human evaluation achieves very similar results to our automatic evaluation. It shows that our automatic evaluation method is effective, and can potentially save manual labor for future work on topic label evaluation.

4.5 Effectiveness of Topic Labelling Method

Effectiveness of Label Results.

To show the effectiveness of our method, some sample topic labels from News, Twitter and NIPS are shown in the Tables 4. Full results over the three corpora can be accessed via internet[5]. We can see from the table that extracted labels are meaningful and representative for the topics. For example, the first topic in News talks about heavy snow which has seriously influenced transportation, the extracted labels like heavy snow and winds represent topic meaning well.

[5] http://lt-lab.sjtu.edu.cn/wordpress/wp-content/uploads/2014/05/topic%20label%20result.zip

5 Conclusion

We propose a method that incorporates word vector and letter trigram vector models to topic labelling. Experiments over three corpora indicate that all three kinds of vectors are better than two baseline methods. Based on the results for automatic and human evaluation, labels extracted using the three vector methods have reasonable utility. We also demonstrated that our automatic method of generating and evaluating labels is effective. The results of word vector models vary across the different corpora, while the letter trigram model is less influenced by the genre of the corpus. Skip-gram outperforms CBOW of the same dimension in data sparse corpus like NIPS and Twitter. The limitation of word vectors is that the quality of a topic label relies on the quality of the word vector representation, which in turn is influenced by the corpus size. In the future, we plan to do more experiments on different types of corpus in order to test the three vector-based models. Letter trigram vectors do not need training, and are more suitable for different types of corpus. We also plan to do more experiments on different types of vector representations, and on vector combination.

Table 4. Label examples for News, Twitter and NIPS topics.

	News topic	Twitter topic	NIPS topic
LDA top 10 words	Snow weather service heavy airport closed storm power county north	Prison guilty murder htt trial rights iran ex human jail	Motion human model visual attention range tracking body target task
GS	Ice and snow hit	Amanda knox murder appeal	Human motion perception
CBOW	Heavy snow and winds	Convicted ex	Motion and camera motion
Skip-gram	Storm closed	Murder trial	Human visual motion perception
Letter trigram	Weather service	Prison sentence	Human visual motion perception
First order	Derry airport closed	Human rights violation	Motion estimation

Reference

1. Gimpel K. Modeling topics[J]. Inform. Retrieval, 2006, 5: 1-23.
2. Hofmann T. Probabilistic latent semantic indexing[C]//Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999: 50-57.
3. Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. the Journal of machine Learning research, 2003, 3: 993-1022.
4. Lau J H, Newman D, Karimi S, et al. Best topic word selection for topic labelling[C]//Proceedings of the 23rd International Conference on Computational Linguistics: Posters. Association for Computational Linguistics, 2010: 605-613.
5. X. Wang and A. McCallum. 2006. Topics over time: A non-Markov continuous-time model of topical trends. In KDD, pages 424–433.
6. Q. Mei, C. Liu, H. Su, and C. Zhai. 2006. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In WWW 2006, pages 533–542.

7. Lau J H, Grieser K, Newman D, et al. Automatic labelling of topic models[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011: 1536-1545.
8. Blei D M, Lafferty J D. Visualizing topics with multi-word expressions[J]. arXiv preprint arXiv:0907.1013, 2009.
9. Cano Basave A E, He Y, Xu R. Automatic labelling of topic models learned from Twitter by summarisation[C]. Association for Computational Linguistics (ACL), 2014.
10. Mei Q, Shen X, Zhai C X. Automatic labelling of multinomial topic models[C]//Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007: 490-499.
11. Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model[J]. The Journal of Machine Learning Research, 2003, 3: 1137-1155.
12. Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning[C]//Proceedings of the 25th international conference on Machine learning. ACM, 2008: 160-167.
13. Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in Neural Information Processing Systems. 2013: 3111-3119.
14. Blunsom P, Grefenstette E, Hermann K M. New Directions in Vector Space Models of Meaning[C]//Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014.
15. Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data[C]//Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. ACM, 2013: 2333-2338.
16. Mitchell J, Lapata M. Composition in distributional models of semantics[J]. Cognitive science, 2010, 34(8): 1388-1429.
17. Socher R, Huval B, Manning C D, et al. Semantic compositionality through recursive matrix-vector spaces[C]//Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, 2012: 1201-1211.
18. Yessenalina A, Cardie C. Compositional matrix-space models for sentiment analysis[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011: 172-182.
19. Jing Yang, Weiran Xu, Songbo Tan. Task and Data Designing of Sentiment Sentence Analysis Evaluation in COAE2014[J]. Journal of Shanxi University (Natural Science Edit), 2015, 1: 003.
20. Lin D. An information-theoretic definition of similarity[C]//ICML. 1998, 98: 296-304.
21. Nikolaos Aletras and Mark Stevenson, "Measuring the Similarity between Automatically Generated Topics" Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 22–27,Gothenburg, Sweden, April 26-30 2014.
22. Nikolaos Aletras and Mark Stevenson, "Labelling Topics using Unsupervised Graph-based Methods" Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 631–636, Baltimore, Maryland, USA, June 23-25 2014.
23. Griffiths T L, Steyvers M. Finding scientific topics[J]. Proceedings of the National Academy of Sciences, 2004, 101(suppl 1): 5228-5235.