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

Wanqiu Kou,♠ Fang Li,♠ and Timothy Baldwin♡

♠ Department of Computer Science and Engineering, Shanghai Jiaotong University
♡ Department of Computing and Information Systems, The University of Melbourne
autumn2012@qq.com, fli@sjtu.edu.cn, tb@ldwin.net

Abstract. The native representation of LDA-style topics is a multinomial distributions over words, which can be time-consuming to interpret directly. As an alternative representation, automatic labelling has been shown to help readers interpret the topics more efficiently. 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 is more semantically related to that topic. A label can be found by calculating the similarity between a topic and its candidate label vectors. Experiments on three common datasets show that not only the labelling method, but also out approach to automatic evaluation is effective.

Key words: Topic Labelling, Word Vectors, Letter Trigram Vectors

1 Introduction

Topic models have been widely used in tasks like information retrieval [1], text summarization [2], word sense induction [3] and sentiment analysis [4]. Popular topic models include mixture of unigrams [5], probabilistic latent semantic indexing [6], and latent Dirichlet allocation (LDA) [7].

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; and (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, an image, or a sentence or paragraph [8]. A word can be too general

Kou, Wanqiu, Li Fang and Timothy Baldwin (to appear) Automatic Labelling of Topic Models using Word Vectors and Letter Trigram Vectors, in Proceedings of the Eleventh Asian Information Retrieval Societies Conference (AIRS 2015), Brisbane, Australia.

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 pre-existing corpora, we demonstrate the effectiveness of our methods.

2 Related work

Topics are usually represented by their top-N words. For example, Blei et al. [7] simply use words ranked by their marginal probabilities $p(w|z)$ in an LDA topic model. Lau et al. [9] 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. A single word can often be inadequate to capture the subtleties of a topic. Some other methods use human annotation [10, 11], 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 to use as topic labels. Lau et al. [12] 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 [13] 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-ranked label as the final result.

Blei and Lafferty [14] used multiword 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 multiword expressions.

Recent work has applied summarization methods to generate topic labels. Cano et al. [15] 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 et al. [16] 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

Symbol	Description
z	A topic
T	The number of topics
ϕ_z	The word distribution of topic z
w	A word
d	A document
θ_d	The topic distribution of d
l	A topic label
L_z	A set of candidate labels for topic z
S	A letter trigram set
D	A document set
V	Vocabulary size
Sim	A word similarity measure
y_w	A vector representation of word w
GS	A gold standard label

Table 1: An overview of the variables used to describe our models

that minimize the distance between the topic word distribution and label word distribution.

Our technique is inspired by the vector based method of Mei et al. [16] and Aletras and Stevenson [17], and work on learning vector representations of words using neural networks [18–20]. The basic intuition is that a topic and a label are semantically related in a semantic vector space.

3 Methodology

3.1 Preliminaries

Distributional vectors can be used to model word meaning, in the form of latent vector representations [18]. 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. The candidate label which has the highest similarity is chosen as the label for that topic.

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

We experiment with three kinds of vectors to label topics: letter trigram vectors from [21], and two word vectors: CBOW (continuous bag-of-words model) and Skip-gram [20]. A letter trigram vector is able to capture morphological variants of the same lemma in close proximity in a letter trigram vector space. CBOW and Skip-gram are methods for learning word embeddings based on distributional similarity. They each capture latent features from a corpus.

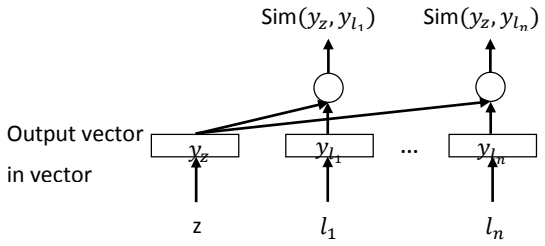


Fig. 1: Outline of our method

3.2 Candidate Label Extraction

We first identify topic-related document sets according to a topic summarization method [15]. The predominant topic of a document d can be calculated by:

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

Given a topic z, the set of documents whose predominant topic is z_d is then simply the set of documents that have z as their predominant topic. For each topic z, we then use `OpenNLP`[1] to full-text chunk parse each document in z_d, and extract chunks that contain at least words in the top-10 words in z, as candidate labels.

3.3 Vector Generation

CBOW Vectors 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 [22].

Recent research has extended the CBOW model to go beyond the word level to capture phrase- or sentence-level representations [23, 24, 22]. We simply apply the weighted additive method [23] to generate a phrase vector. Word vectors of candidate labels and LDA topics are generated as follows:

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

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

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

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

Skip-gram Vectors The Skip-gram model [22] 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 Skip-gram model can capture latent document features and has been shown to perform well over semantic relation classification tasks [22].

Phrase vectors are, once again, generated using the weighted additive method [23]. Skip-gram vectors of candidate labels and LDA topics are generated in the same manner as CBOW, based on $y_{w_j}^{skip}$ (i.e. word vectors from Skip-gram).

Letter Trigram Vectors We use the method of [21] to generate vectors for the topic and its candidate labels based on letter trigrams. Each dimension in a letter trigram vector represents a letter trigram (e.g. *abc* or *acd*). We generate a letter trigram set for each phrase l. A letter trigram set is defined as the multiset of letter trigrams from the phrase. For example, the letter trigram multiset of the phrase *stock market* is { *ˆst*, *sto*, *toc*, *ock*, *ck␣*, *␣ma*, *mar*, *ark*, *rke*, *ket*, *et$*}. For each dimension i in the letter trigram vector of phrase l, we assign an integer value based on the frequency of the corresponding letter trigram in the multiset, and normalize the counts to sum to one.

Using a similar method, we generate the letter trigram multiset of each of the top-10 LDA words, and take the union of the individual letter trigram multisets to calculate the overall letter trigram distribution for the top-10 words. We derive a vector representation for the topic based on the combined letter frequencies, and once again, normalize the counts to sum to one.

3.4 Topic Label Selection

After generating vectors for candidate labels and LDA topics, we then calculate the similarity between them based on cosine similarity.

4 Experiments

4.1 Dataset & Gold Standard

We use three corpora in our experiments: (1) News, (2) Twitter, and (3) NIPS. The News and Twitter corpora are from [15], while the NIPS corpus is a collection of NIPS abstracts from 2001 to 2010, commonly used for topic model evaluation.

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 a within-topic entropy-based method to filter bland topics, i.e. topics where the probability distribution over the component words is relatively uniform, based on:

$$H(z) = -\sum_{i=1}^{V} \phi_z(w_i) \log_2(\phi_z(w_i)) \tag{4}$$

Corpus	# documents	# topics (T)	# pruned topics
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

Table 2: The datasets used in this research

In the News and Twitter corpora, topics with an entropy higher than 0.9 were eliminated, and in the NIPS corpus, topics with an entropy higher than 1.4 were eliminated; these thresholds were set based on manual analysis of a handful of topics for each document collection. For the Twitter corpus, we further filtered topics which lack a meaningful gold-standard topic label, based on the method described later in this section. Table 2 provides details of the datasets.

Yang [25] observed that gold standard labels from human beings suffer from inconsistency. The inter-annotator F-measure between human annotators for our task is 70–80%. In an attempt to boost agreement, we developed 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 is a good representation of a document. Therefore a phrase from a title can be a good label for the predominant topic associated with that document.

Note that the News and NIPS corpora have titles for each document, while the Twitter corpus has no 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, based on [15]: (1) calculate the cosine similarity between each pair of Twitter and News topics, based on their word distributions; (2) 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) label for Twitter topic i.

4.2 Evaluation Metrics

We evaluate our results automatically and via human evaluation.

Automatic Evaluation Method Because of the potential for semantically similar but lexically divergent labels, we can't compare the generated label directly with the GS automatically. Rather, we propose the following evaluation:

$$score_z = \frac{\sum_{w \in GS} \max_{w' \in l} Lin'(w, w') + \sum_{w' \in l} \max_{w \in GS} Lin'(w, w')}{\#words(GS) + \#words(l)} \quad (5)$$

$$Lin'(w, w') = \begin{cases} 1 & \text{if } stem(w) = stem(w') \\ Lin(w, w') & \text{otherwise} \end{cases} \quad (6)$$

where Lin is word similarity based on WordNet, in the form of the information-theoretic method of Lin [26]. GS and l represent the gold standard and the label generated for topic z, respectively. The Porter stemmer[2] is used to stem all words. The score is used to measure the semantic similarity between an automatically-generated and GS label.

Human Evaluation Method We also had six human annotators manually score the extracted labels. Each annotator was presented with the top-10 LDA words for a given 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. We average the scores from the six annotators to calculate the overall topic label score.

4.3 Baseline Methods

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

DistSim This method was proposed by [16], and involves generating a word vector of candidate labels according to first-order cooccurrence-based 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` toolbox[3] 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 2, 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.

[2] `http://tartarus.org/~martin/PorterStemmer/`
[3] `https://github.com/NLPchina/Word2VEC_java`

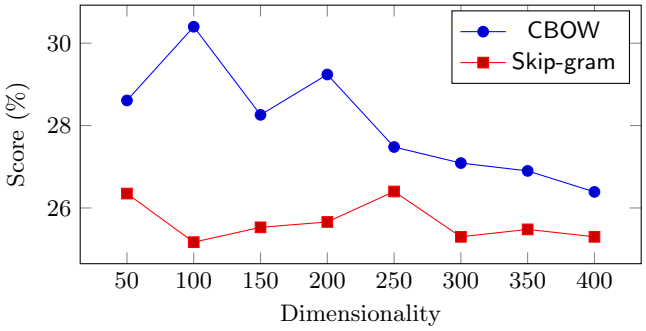


Fig. 2: Automatic evaluation results for word vectors over the NEWS corpus for different embedding dimensionalities

Method	Automatic evaluation (%)			Human evaluation		
	NEWS-50	TWITTER-40	NIPS-50	NEWS-50	TWITTER-40	NIPS-50
LDA-1	33.73	32.65	41.62	1.04	1.00	1.02
DistSim	42.77	41.90	42.74	1.60	1.60	1.70
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 trigram	47.17	44.08	53.14	1.86	1.50	2.13

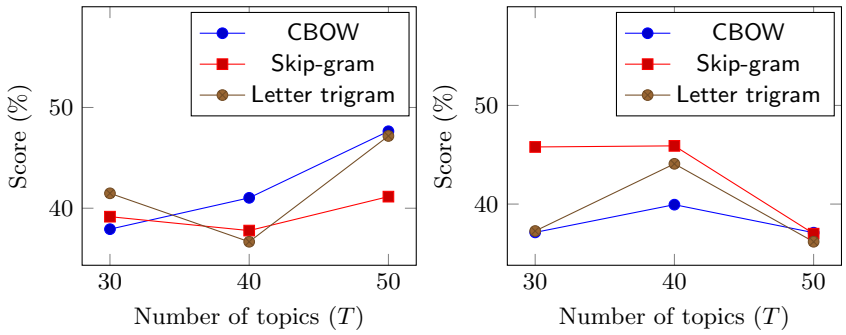
Table 3: Evaluation of the baselines and the proposed approaches

Figure 3 shows the automatic evaluation results for topic labelling with different numbers of topics. We can see that the results vary 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.

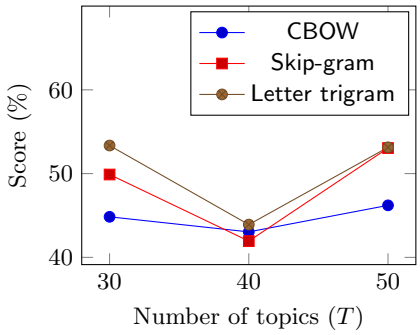
Table 3 shows the result between the baseline methods (LDA-1 and DistSim) and our methods, over the NEWS-50, TWITTER-40 and NIPS-50 corpora.

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

1. Most methods perform better over NIPS than NEWS and TWITTER. The primary 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; CBOW performs relatively badly over TWITTER. The reason is that Skip-gram works better over sparse corpora like TWITTER and NIPS, while CBOW works better over dense corpora. Mikolov et al. [22] show that Skip-gram performs better over semantic tasks while CBOW performs better over shallow syntactic tasks, based on which we assumed that Skip-gram should be better for topic



(a) News (b) Twitter



(c) NIPS

Fig. 3: Automatic evaluation results

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 agents and details of a particular 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 corpus genre. Compared with DistSim, 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-1 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 DistSim baseline in most cases. Our result shows that trigram vectors are more suitable for topic labelling

	News	Twitter	NIPS
LDA top-10	⟨*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
DistSim	Derry airport closed	human rights violation	motion estimation

Table 4: Sample topic labels

over different types of corpus. Skip-gram is better than CBOW for Twitter and NIPS, while CBOW is more suitable for News.

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 somewhat 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; in fact, we calculated the Pearson correlation between the two and found it to be remarkably high at $r = 0.84$. This 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

To show the effectiveness of our method, some sample topic labels from News, Twitter and NIPS are shown in the Table 4. Full results over the three corpora are available for download from:

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

5 Conclusion

We have proposed a novel method for topic labelling using embeddings and letter trigrams. 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. 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. 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. The novelty of our work includes the use of embeddings for label ranking, the automatic method to generate gold-standard labels, and the method to automatically evaluate labels. In the future, we plan to do more experiments on different types of corpora. 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, and also extrinsic evaluation of the topic labels [8].

Acknowledgements

This research was supported in part by the Australian Research Council.

References

1. Xu, J., Croft, W.B.: Cluster-based language models for distributed retrieval. In: Proceedings of 22nd International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), Berkeley, USA (1999) 254–261
2. Haghighi, A., Vanderwende, L.: Exploring content models for multi-document summarization. In: Proceedings of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies 2009 (NAACL HLT 2009), Boulder, USA (2009) 362–370
3. Lau, J.H., Cook, P., McCarthy, D., Newman, D., Baldwin, T.: Word sense induction for novel sense detection. In: Proceedings of the 13th Conference of the EACL (EACL 2012), Avignon, France (2012) 591–601
4. Lin, C., He, Y.: Joint sentiment/topic model for sentiment analysis. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009), Hong Kong, China (2009) 375–384
5. Gimpel, K.: Modeling topics. Information Retrieval **5** (2006) 1–23
6. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of 22nd International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), Berkeley, USA (1999) 50–57
7. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. Journal of Machine Learning Research **3** (2003) 993–1022
8. Aletras, N., Baldwin, T., Lau, J.H., Stevenson, M.: Evaluating topic representations for exploring document collections. Journal of the Association for Information Science and Technology (to appear)
9. Lau, J.H., Newman, D., Karimi, S., Baldwin, T.: Best topic word selection for topic labelling. In: Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010), Posters Volume, Beijing, China (2010) 605–613
10. Wang, X., McCallum, A.: 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, Philadelphia, USA (2006) 424–433

11. Mei, Q., Liu, C., Su, H., Zhai, C.: A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In: Proceedings of the 15th International Conference on the World Wide Web (WWW 2006), Edinburgh, UK (2006) 533–542
12. Lau, J.H., Grieser, K., Newman, D., Baldwin, T.: Automatic labelling of topic models. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL HLT 2011), Portland, USA (2011) 1536–1545
13. Grieser, K., Baldwin, T., Bohnert, F., Sonenberg, L.: Using ontological and document similarity to estimate museum exhibit relatedness. ACM Journal on Computing and Cultural Heritage **3**(3) (2011) 1–20
14. Blei, D.M., Lafferty, J.D.: Visualizing topics with multi-word expressions. arXiv preprint arXiv:0907.1013 (2009)
15. Cano Basave, E.A., He, Y., Xu, R.: Automatic labelling of topic models learned from twitter by summarisation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014). (2014) 618–624
16. 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). (2007) 490–499
17. Aletras, N., Stevenson, M.: Measuring the similarity between automatically generated topics. (2014) 22–27
18. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. The Journal of Machine Learning Research **3** (2003) 1137–1155
19. Collobert, R., Weston, J.: A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning (ICML 2008), Helsinki, Finland (2008) 160–167
20. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of Workshop at the International Conference on Learning Representations, 2013, Scottsdale, USA (2013)
21. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM Conference on Information and Knowledge Management (CIKM 2013), San Francisco, USA (2013) 2333–2338
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. (2013) 3111–3119
23. Mitchell, J., Lapata, M.: Composition in distributional models of semantics. Cognitive Science **34**(8) (2010) 1388–1429
24. Socher, R., Huval, B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning 2012 (EMNLP-CoNLL 2012), Jeju Island, Korea (2012) 1201–1211
25. Yang, J., Xu, W., Tan, S.: Task and data designing of sentiment sentence analysis evaluation in COAE2014. Journal of Shanxi University (Natural Science Edit) **1**(3) (2015)
26. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the International Conference on Machine Learning, Madison, USA (1998) 296–304