2016 IEEE First International Conference on Data Science in Cyberspace

A Hybrid Document Feature Extraction Method Using Latent Dirichlet Allocation and Word2Vec

Zhibo Wang, Long Ma, and Yanqing Zhang
Department of Computer Science
Georgia State University
Atlanta, GA 30302-5060, USA
{zwang6, lma5}@student.gsu.edu, yzhang@gsu.edu

***Abstract*— Latent Dirichlet Allocation (LDA) is a probabilistic topic model to discover latent topics from documents and describe each document with a probability distribution over the discovered topics. It defines a global hierarchical relationship from words to a topic and then from topics to a document. Word2Vec is a word-embedding model to predict a target word from its surrounding contextual words. In this paper, we propose a hybrid approach to extract features from documents with bag-of-distances in a semantic space. By using both Word2Vec and LDA, our hybrid method not only generates the relationships between documents and topics, but also integrates the contextual relationships among words. Experimental results indicate that document features generated by our hybrid method are useful to improve classification performance by consolidating both global and local relationships.**

Keywords—bag-of-distances, classification, document representation, feature extraction, Latent Dirichlet Allocation, text mining, Word2Vec

I. INTRODUCTION

With the explosive growth of online resources such as web pages, blogs, and social networks, text mining plays a more and more important role to analyze and organize these documents. An excellent representation of textual data should contain as much information as possible from the original document. Generally, there are two ways to represent a document — one-hot encoding and word embedding.

One-hot encoding describes a word in a high-dimensional vector, which is a dictionary composed of all words occurred in a set of documents. Each word is represented by a vector with 1 at its corresponding position and 0 in other positions. A model to represent a document called “bag-of-words” was proposed [1]. It sums up all the one-hot vectors in a document; and each element in the resulting vector becomes the occurrence of a word. Furthermore, Term Frequency-Inverse Document Frequency (TF-IDF) model was given to replace the counts with TF-IDF score [2]. The new model calculates TF scores to the selected high frequency words in a document, and also measures how unique these words occur across all documents using IDF scores. Using the product of the TF-IDF scores, the high frequency but less meaningful words can be eliminated, such as “that”, “this”, “the”, etc.

LDA is a probabilistic topic model to discover latent topics from a large volume of documents and describe each document with a probability distribution over the discovered topics [3]. It is commonly considered as a feature reduction method by grouping words in different topics, thus a document can be mapped to a lower dimensional space. Additionally, words are assumed to occur independently in LDA, and documents are treated as bag-of-words. Therefore, LDA does not study the contextual relationship among words. Moreover, LDA is also a doubly sparse model which prefers fewer topics in each document and fewer words to describe a topic. Thus, the document vector is very sparse.

Recently, Word embeddings has been a strong trend in Natural Language Processing. It distributes a word in a low-dimension vector that is highly correlated with the real semantics. Generally speaking, there are two approaches: one builds a co-occurrence matrix for the entire document and reduces the size of the matrix to generate words and context, such as Glove, Spectral Word Embeddings, and Word Embeddings through Hellinger PCA (HPCA) [4, 5, 6]. The other one, such as Latent Semantic Analysis (LSA), density based word embeddings, and Word2Vec, predicts a word by inspecting its surroundings [7, 8, 9]. For example, if two words “soccer” and “basketball” occur in a same “position” in two sentences “I like soccer” and “I like basketball”, “soccer” and “basketball” are more likely related either in semantics or syntactic. A method clusters the embedded word vectors as features and uses a count distribution as document representations [10]. A Doc2Vec model trains a document vector by a linear combination of the embedded word vectors [11].

LDA with strong capability to extract the main contents of the article is quite interpretable by humans. Therefore, many researchers use LDA on the text classification tasks. A feature-enhanced smoothing method was developed [12]. Those words existing in testing documents but not in the training corpus are very useful to improve accuracy of classification and the quality of features. An improved algorithm gLDA was designed by containing categories for each document [13]. The probability distribution of each document is generated by the most relevant categories of documents. The word-topic mapping performance was improved by using a large-scale trained corpora applied to the data with smaller corpus [14]. Websites were divided into different subjects with slash tags and a relationship between each subject and topics used to classify the data was found [15]. A novel classifier named Multi-LDA Boost applied a boosting strategy by choosing the best scenario from multiple models with different parameters, and performed a weighted method to improve the accuracy of categorization [16].

978-1-5090-1192-6/16 $31.00 © 2016 IEEE
DOI 10.1109/DSC.2016.110

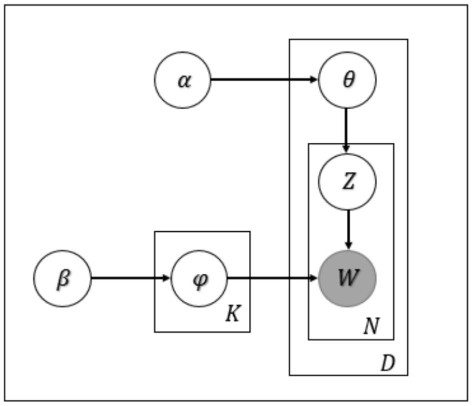


Fig. 1. A graphic of LDA model

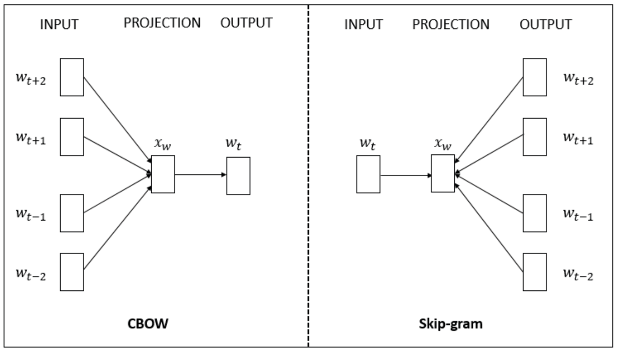


Fig. 2. Word2Vec has two models: Continuous Bag of Word (CBOW) and Skip-gram, where w_t is the target word and $\{w_{(t-2)}, w_{(t-1)}, w_{(t+1)}, w_{(t+2)}\}$ are contextual words of w_t

Similar ideas but different methods and purposes by integrating LDA and Word2Vec are implemented. LDA models a global relationship from each document to all topics, and Word2Vec in the other hand captures the relationships by learning the target word from its contexts. Topic2Vec integrates the word contextual information from Word2Vec to learn topic representations in LDA, whose resulting topics are much more distinguishable than those generated by LDA [17]. LDA2Vec successfully uses the contextual word information to learn much more interpretable topics by adding the document vector in the step of generating word vectors [18].

In this paper, we propose a new hybrid method to represent documents in a comprehensive way. Incorporating Word2Vec with LDA obtains relationships between documents and topics from LDA, as well as the contextual relationships from Word2Vec. Euclidean distance is used to measure and interpret similarity between document and topic in the space. To investigate its performance, the experiment is set up with the Support Vector Machine (SVM) model on the 20NewsGroups dataset. Compared with other methods such as TF-IDF+SVM, Word2Vec+SVM, LDA+SVM, our new hybrid method performs better in terms of discrimination and classification.

The remaining part of this paper is organized as follows. LDA and Word2Vec are briefly introduced in Section 2. In Section 3, our proposed model is described. In Section 4, experimental results are given. Finally, conclusions and future work are discussed in Section 5.

II. Related Models

A. Latent Dirichlet Allocation

LDA is an unsupervised method to discover the latent topics Z from a collection of documents D. In LDA, each document d is represented as a probability distribution θ_d over topics, where each topic z is a probability distribution φ_z over all words in vocabulary. Fig. 1 shows the generative process. Both θ and φ have prior distributions with hyperparameters α and β. For every word w_{d_i} in document d, a topic z_{d_i}can be extracted by equations (1) and (2), a word w_{d_i}can be returned. Repeat (1) and (2) N times, a document d is generated, where N is the size of document d.

$$\theta_d \sim Dirichlet(\alpha) \quad z_{d_i} \sim Multinomial(\theta_d) \qquad (1)$$

$$\varphi_z \sim Dirichlet(\beta) \quad w_{d_i} \sim Multinonial\left(\varphi_{z_{d_i}}\right) \qquad (2)$$

By using Gibbs Sampling, θ and φ can be inferred to discover the latent topics in documents, and predict any new document with a topic proportion distribution.

B. Word2Vec

Word2Vec includes two alternative models to update parameters. 1) Continuous Bag of Words (CBOW) is a way to predict words by using contexts of its surroundings; 2) in contrast, Skip-gram uses a word's information to predict its neighboring words. As shown in Fig. 2, both models contain three layers: an input layer, a projection layer and an output layer. We take CBOW as example to briefly explain how Word2Vec works.

Given a sentence $W = \{w_{t-2}, w_{t-1}, w_t, w_{t+1}, w_{t+2}\} \in \mathbb{R}^m$, where w_t is the target word.

Input layer: $Context(v(w_t)) = \{v(w_{t-2}), v(w_{t-1}), v(w_{t+1}), v(w_{t+2})\} \in \mathbb{R}^m$.

Projecting layer: a contextual vector $v(x_w)$ is calculated by $v(x_w) = \sum_{i=t-2}^{t+2} contxt(v(w_i))$ where $i \neq t$.

Output layer: A word in vocabulary is treated as a leaf node in a Huffman tree according to its occurrence in the corpus. Therefore, each word has a unique path from root node to leaf node. At each node except for the leaf node, the probability of selecting left child or right child can be estimated by the llogistic model by (3)

$$left\ child : \sigma(v(x_w)^T\theta) = \frac{1}{1+e^{-v(x_w)^T\theta}}$$

$$right\ child: 1 - \sigma(v(x_w)^T\theta) \qquad (3)$$

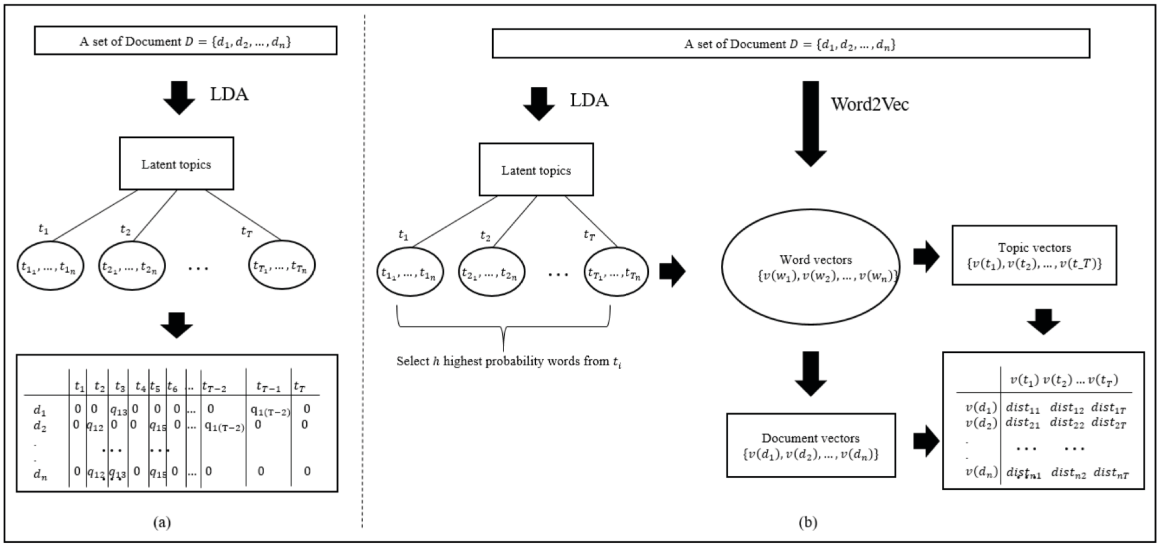

A set of Document $D = \{d_1, d_2, \dots, d_n\}$
LDA
Latent topics
t_1
t_2
t_T
$t_{1_1}, \dots, t_{1_n}$
$t_{2_1}, \dots, t_{2_n}$
$t_{T_1}, \dots, t_{T_n}$
t_1 t_2 t_3 t_4 t_5 t_6 .. t_{T-2} t_{T-1} t_T
d_1 0 0 q_{13} 0 0 0 ... 0 $q_{1(T-2)}$ 0
d_2 0 q_{12} 0 0 q_{15} 0 ... $q_{1(T-2)}$ 0 0
d_n 0 q_{12} q_{13} 0 q_{15} 0 ... 0 0 0
(a)
A set of Document $D = \{d_1, d_2, \dots, d_n\}$
LDA
Word2Vec
Latent topics
t_1
t_2
t_T
$t_{1_1}, \dots, t_{1_n}$
$t_{2_1}, \dots, t_{2_n}$
$t_{T_1}, \dots, t_{T_n}$
Select h highest probability words from t_i
Word vectors $\{v(w_1), v(w_2), \dots, v(w_n)\}$
Topic vectors $\{v(t_1), v(t_2), \dots, v(t\_T)\}$
Document vectors $\{v(d_1), v(d_2), \dots, v(d_n)\}$
$v(t_1)$ $v(t_2)$ ... $v(t_T)$
$v(d_1)$ $dist_{11}$ $dist_{12}$ $dist_{1T}$
$v(d_2)$ $dist_{21}$ $dist_{22}$ $dist_{2T}$
$v(d_n)$ $dist_{n1}$ $dist_{n2}$ $dist_{nT}$
(b)

Fig. 3. Processes and results of LDA (a) and our hybrid method (b).

$p\left(v(x_w)\middle|Context(v(x_W))\right)$ can be learned in the tree by a production of probabilities at each node by (4), where $d_j^w \in \{0,1\}$ is the j^{th} digit in word w's Huffman code and j is any node on the path except as the leaf node.

$$p\left(v(x_w)\middle|Context(v(x_W))\right) = [\sigma(v(x_w)^T\theta_{j-1}^w)]^{1-d_j^w}[1 - \sigma(v(x_w)^T\theta_{j-1}^w)]^{d_j^w} \quad (4)$$

The objective function (5) can be learned by maximizing the log-likelihood, and then use gradient descent method to update θ, $v(x_w)$ and its contextual words.

$$\mathcal{L} = \sum_{w \in C} log \prod_{j=2}^{n} \left\{ [\sigma(v(x_w)^T\theta_{j-1}^w)]^{1-d_j^w}[1 - \sigma(v(x_w)^T\theta_{j-1}^w)]^{d_j^w} \right\} \quad (5)$$

III. New Framework

As we have discussed, LDA is a way to describe a global relationship among documents, while Word2Vec predicts words in a very local manner. So we combine these two techniques to use a more comprehensive vector to represent documents, meanwhile, the new representation with a density vector enhances the capability of discrimination and predication applied to Natural Language Process tasks.

Our new method as shown in Fig. 3 (b) projects words, documents, and topics in a high-dimension semantic space. A document vector is considered as a single vector, which is the centroid of all words in the document as what Word2Vec does in the projection layer. In addition, each document has its individual length, thus its vector is divided by the number of words in the document to guarantee the measurements with same scale. We construct topic vectors in a similar way, but it is a little more complicated. A subset of h high-probability words in each topic is employed to represent the topic, and then their probabilities are rescaled as the weights of words. Hence different words have different contributions to the topic. We measure Euclidean distances from each document to topics so that a document can be represented with a distance distribution.

In details, given a set of documents $D = \{d_1, d_2, \dots, d_n\}$, whose vocabulary is built with N words $\{w_1, w_2, \dots, w_N\}$. By training D , LDA outputs latent topics $\{t_1, t_2, \dots, t_T\}$ and probabilities of words in each topic t_i, where the j^{th} word in t_i is denoted as θ_{i_j}. Word2Vec trains D and vectorizes each word in vocabulary into a fixed length vector $\{v(w_1), v(w_2), \dots, v(w_N)\}$. To generate topic vectors, h highest-probability words in t_i are selected. Meanwhile, the probabilities of words in t_i are rescaled as weights in (6). In (7), the topic vector $v(t_i)$is calculated by summing the productions of each word vector and its weight.

$$\omega_i = \frac{\theta_i}{\sum_{n=1}^{h}\theta_n} \quad (6)$$

$$v(t_i) = \sum_{n=1}^{h}\omega_{i_n} v(w_{i_n}) \quad (7)$$

Next, we calculate document vectors $v(d_i)$ by (8), where c is the number of words in the document.

$$v(d_i) = \frac{\sum_{n=1}^{c} v(w_{i_n})}{c} \quad (8)$$

Therefore, each document can be represented by a distance distribution from the document to all topics in a semantic space, and a distance is calculated as (9).

$$distance(v(d_i), v(t_i)) = |v(d_i) - v(t_i)| \quad (9)$$

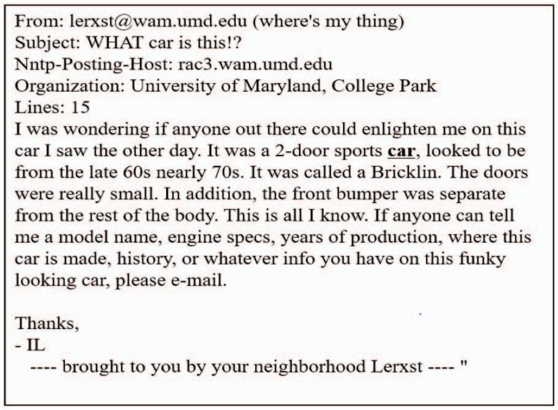
Therefore, the new defined vector is no longer sparse by comparing the results in (a) and (b) of Fig. 3, which distributes

TABLE I. TOPIC DISTRIBUTION AND DISNTANCE DISTRIBUTION OF THE EXAMPLE DOCUMENT

Topic Distribution (Index, Probability)	**Distance Distribution (Index, Distance)**						
(9, 0.3677) (11, 0.018) (22, 0.345) (47, 0.113) (49, 0.092) *Others topics not listed are with probabilities of 0.	(1, 2.005)	(2, 1.597)	(3, 2.376)	(4, 1.285)	(5, 2.119	(6, 1.646)	(7, 1.889)
	(8, 2.704)	**(9, 1.698)**	(10, 2.254)	**(11, 1.692)**	(12, 1.691)	(13, 2.646)	(14, 1.743)
	(15, 1.154)	(16, 1.928)	(17, 1.269)	(18, 1.706)	(19, 2.147)	(20, 1.775)	(21, 1.792)
	(22, 1.603)	(23, 1.927)	(24, 2.030)	(25, 1.421)	(26, 2.169)	(27, 1.418)	(28, 1.803)
	(29, 1.774)	(30, 2.323)	(31, 1.793)	(32, 1.568)	(33, 2.010)	(34, 1.604)	(35, 2.067)
	(36, 1.107)	(37, 1.591)	(38, 1.594)	(39, 1.743)	(40, 2.096)	(41, 1.879)	(42, 1.703)
	(43, 1.655)	(44, 1.286)	***(45, 3.728)***	(46, 2.107)	**(47, 1.269)**	(48, 1.638)	**(49, 0.985)**
	(50, 2.002)						
	*Mean = 1.820 Minimum=0.985						

TABLE II. WORDS CONTAINED IN TOPIC 15 AND TOPIC 45

Topic 15	**Topic 45**
writes, **bike**, article, dod, lines, organization, org, posting, nntp, host, apr, rochester, **bmw**, mitre, ride, upenn, clarkson, dog, att, riding, sas, **motorcycle**, john, **bikes**, reply, shaft, inc, rec, rider, noise, helmet, chain, well, mail, list, ahl, **motorcycles**, like, tek, ysu, **sport**, dave, corporation, road, bill, wave, wax, yfn, lock, pink	max, bhj, giz, scx, rlk, chz, qax, bxn, biz, air, fij, okz, gcx, nrhj, rck, ync, frustrated, uww, fil, cho, mvs, hernia, nei, mbs, tct, rmc, lhz, umu , wwiz, nuy, ahf, qtm, ghj, kjz, vmk, ecs, mcx, fpl, syx, pmf , dct, barman, srcs, gizw, mkg, qvf, bhjn, mgb, mas, khf'



From: lerxst@wam.umd.edu (where's my thing)
Subject: WHAT car is this!?
Nntp-Posting-Host: rac3.wam.umd.edu
Organization: University of Maryland, College Park
Lines: 15

I was wondering if anyone out there could enlighten me on this car I saw the other day. It was a 2-door sports **car**, looked to be from the late 60s nearly 70s. It was called a Bricklin. The doors were really small. In addition, the front bumper was separate from the rest of the body. This is all I know. If anyone can tell me a model name, engine specs, years of production, where this car is made, history, or whatever info you have on this funky looking car, please e-mail.

Thanks,
- IL
---- brought to you by your neighborhood Lerxst ---- "

Fig.4. An original document sample

risks to all elements $dist_{ij}$ in the vector, also possesses more information in $dist_{ij}$ to be discriminated from other documents. Moreover, the probability distribution over topics in LDA is still held in the space generated by the new method; and these topics are closer to the specific document. Meanwhile, other related important topics are found as well because of more word-level information is involved.

IV. EXPERIMENT RESULTS

The 20Newsgroups dataset contains 18846 newsgroup documents collected by Ken Lang, which is organized into 20 different newsgroups [19]. We use all the documents in the dataset to train LDA and Word2Vec to extract latent topics and word vectors. Both LDA and Word2Vec are implemented with Gensim which is a free Python package widely used for topic models [20]. In LDA, the hyperparameter α is set to 0.1 and $passes$ is set to 20 to guarantee convergence. Word2Vec uses the CBOW model with default settings in Gensim. We use the Python scikit-learn package to perform SVM for the classification with $gamma = 0.001$ [21].

TABLE III. AVERAGE 10 FOLD MIRCRO-F1 SCORE OF DIFFERENT METHODS

	Average 10 Fold micro-F1 score
TF-IDF+SVM	0.822
Word2Vec+SVM	0.717
LDA +SVM	0.639 (# topic = 100)
Our method	0.803 (# topic = 250)

In the first part of our experiment, we tested whether our new representation carried the relationships from LDA, and enriched results with extra benefits Word2Vec brings. Next, we compared our method with three other methods on classification tasks.

Fig. 4 shows an original news document from the 20 Newsgroups dataset. After learned by LDA, the topic distribution is shown in the left column of Table 1, which has 5 very relevant topics 9, 11, 22, 47, and 49. Except for these, other topics with values of zero are considered as irrelevant. The right column is the distance distribution using our method. The bolded topics underlined are corresponding to the topics listed

TABLE IV. AVERAGE 10 FOLD MICRO-F1 SCORE OF LDA AND OUR METHOD UNDER DIFFERENT NUMBER OF TOPICS

	Average 10 fold micro-F1 score (Standard deviation)					
	50 topics	**100 topics**	**150 topics**	**200 topics**	**250 topics**	**300 topics**
LDA+SVM	0.615 (0.016)	0.639 (0.016)	0.639 (0.016)	0.619 (0.015)	0.589 (0.010)	0.536 (0.019)
Our method	0.748 (0.016)	0.777 (0.014)	0.794 (0.013)	0.796 (0.015)	0.803 (0.012)	0.789 (0.015)

in the left column, where the first value is the index of topics and the second value is the normalized distance. We can see that all the highlighted values are below the mean of 1.82 and topic 49 is the minimum value 0.985 among all topics. Therefore, the conclusions of LDA are well held in our new representations. Moreover, we are also interested in the italicized and bolded topics in the right column, such as the topic 15 with a very short distance and topic 45, who has the longest distance. To interpret these, we investigate Table 2 at first, which has two word lists of topic 15 and topic 45. Topic 15 is found by our method but missed by LDA. The words in topic 15 such as “motorcycle”, ‘bmw’, ‘bike’, ‘sport’, etc. are quite related to the word ‘car’ mentioned in the example document, where ‘car’ and ‘motorcycle’, ‘bike’ belong to vehicles, also ‘bmw’ is a brand of ‘car’. Therefore, topic 15 and the example document are highly relevant at the word-level. Topic 45 obviously contains a lot of ‘trash’ words without any semantic meanings, therefore, it makes sense that it has the longest distance to the document.

To investigate the performance, we compare TF-IDF, Word2Vec, LDA and our method using the SVM model. From Table 3, TF-IDF performs the best prediction with about 2% more accuracy rate over our method, but it takes more than 4 times of the running time comparing with our method with about 20000 features. In contrast, Word2Vec only trains a predefined number of features no more than 500, while LDA and our method only train the same number of features as topics. With the growing words occurred in documents, the performance of TF-IDF begins to decrease more drastically. Moreover, similar to LDA, TF-IDF describes the occurrences of words without semantic meanings. Considering the prediction accuracy, running time, and the volume of semantic information involved, our method performs effectively among any single methods.

Another experiment is conducted to evaluate the performances of LDA and our method under different number of topics. As shown in Table 4, 100 - 150 topics are the optimal range of LDA to categorize the newsgroups data, while the range of 150 -250 topics is the best for our model. Therefore, our method requires more topics to predict documents because our document representation takes more contextual information in consideration.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a more comprehensive representing method for documents by integrating of Word2Vec and LDA. The proposed hybrid document vector not only preserves the statistical relationships between topics and documents, but also assimilates the relationships among words into the document vector. In practice, our method performs effectively compared with three other single models.

Based on our proposed work in this paper, we have two future plans. One hand, we further evaluate the performances in classification by applying our new document features in more supervised models, such as Naïve Bayes model, Neural Network model, and deep learning models. Additionally, we also implement these document features in unsupervised models such as K-Means to cluster documents into different groups. The documents in each group are semantically related, which is able to investigate how precisely an extracted document representation performs.

The other hand, we aim to continue improving the performance of our document representations. Our experiment results are much closed to TF-IDF which uses more than 10000 features, in contrast our method only uses less than 300 features. Feature reduction methods such as Principal Component Analysis (PCA) are useful to optimize the classification performance with fewer features. Under a same scenario, our features are easily reduced to a small number like 20 features, while TF-IDF is hard to handle this. Therefore, after optimization, our method is advantageous for better performance with a small number of features. Moreover, during the experiments, we found some topics are similar to each other. Clustering the closer topics as one new feature not only reduces the number of features in the training, but also separates the features more independently. Finally, the initial topics generated by LDA are very significant to the resulting document representations. Therefore, a way to accurately select the number of topics is crucial for the quality of representations.

REFERENCES

[1] Z. Harris, "Distributional Structure". Word. Vol. 10, pp. 146–62, 1954.

[2] G. Salton and C. Buckley, “Term Weighting Approaches in Automatic Text Retrieval,” Information Processing and Management, 24(5), pp. 513-23 1988.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” Journal of Machine Learning Research, vol. 3, no. 4-5, pp. 993–1022, 2003.

[4] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” Proceedings of the Empirical Methods in Natural Language Processing, 2014.

[5] P. Dhillon, D. Foster, and L. Ungar, “Eigenwords: Spectral word embeddings,” Journal of Machine Learning Research, 2015

[6] R. Lebret, and R. Collobert, “Word embeddings through Hellinger PCA,” In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 482-490, 2014.

[7] T. K. Landauer, P. W. Foltz, and D. Laham, “Introduction to latent semantic analysis,” Discourse Processes, pp. 259–284, 1998.

[8] L. Vilnis and A. McCallum, “Word representations via Gaussian embedding,” In Proceedings of the International Conference on Learning Representations, 2015.

[9] Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” In Advances in Neural Information Processing Systems, pp. 3111–3119, 2013.

[10] L. Ma, and Y. Zhang, “Using Word2Vec to Process Big Text Data,” 2015 IEEE International Conference on Big Data, pp. 2895-2897, 2015.

[11] Q. V. Le and T. Mikolov, “Distributed Representations of Sentences and Documents,” International Conference on Machine Learning, 2014.

[12] D. Liu, W. Xu and J. Hu, "A feature-enhanced smoothing method for LDA model applied to text classification," Natural Language Processing and Knowledge Engineering, 2009. NLP-KE 2009. International Conference on, Dalian, 2009, pp. 1-7.

[13] D. Zhao, J. He and J. Liu, "An improved LDA algorithm for text classification," Information Science, Electronics and Electrical Engineering (ISEEE), 2014 International Conference on, Sapporo, 2014, pp. 217-221.

[14] D. Q. Nguyen, R. Billingsley, L. Du, and M. Johnson, “Improving Topic Models with Latent Feature Word Representations,” Transactions of the Association for Computational Linguistics 3, pp. 299–313, 2015.

[15] L. Niu, and X. Dai, “Topic2Vec: Learning Distributed Representations of Topics,” arXiv preprint arXiv:1506.08422. 2015 Jun 28.

[16] Y. Wang and Q. Guo, "Multi-LDA hybrid topic model with boosting strategy and its application in text classification," Control Conference (CCC), 2014 33rd Chinese, Nanjing, 2014, pp. 4802-4806.

[17] C. Moody, “A Word is Worth a Thousand Vectors,” http://multithreaded.stitchfix.com/blog/2015/03/11/word-is-worth-a-thousand-vectors/

[18] K. Lang, “20 newsgroup data set,” 30-Sep-2015, qwone.com/~jason/20Newsgroups/

[19] R. Reh˚ ˇ uˇrek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” In New Challenges for NLP Frameworks, 2010.

[20] F. Pedregosa, Fabian, et al., "Scikit-learn: Machine learning in Python," The Journal of Machine Learning Research 12 (2011): 2825-2830.

[21] C. Yang, and W. Jun Wen,” Text Categorization Based on Similarity Approach,” In Proceedings of International Conference on Intelligence Systems and Knowledge Engineering (ISKE), 2007.