Document Classification with Distributions of Word Vectors

Chao Xing*†, Dong Wang*, Xuewei Zhang*‡, Chao Liu*
*Center for Speaker and Language Technologies (CSLT)
Tsinghua University, Beijing 100084, P.R.China
E-mail: wangdong99@mails.tsinghua.edu.cn; liuc@cslt.riit.tsinghua.edu.cn
†School of Software, Beijing Jiaotong University, Beijing 100044, P.R.China
E-mail: xingchao@cslt.riit.tsinghua.edu.cn
‡Shenyang Jianzhu University, Sheyang 110168, P.R.China
E-mail: zxw@cslt.riit.tsinghua.edu.cn

Abstract—**The word-to-vector (W2V) technique represents words as low-dimensional continuous vectors in such a way that semantic related words are close to each other. This produces a semantic space where a word or a word collection (e.g., a document) can be well represented, and thus lends itself to a multitude of applications including document classification. Our previous study demonstrated that representations derived from word vectors are highly promising in document classification and can deliver better performance than the conventional LDA model. This paper extends the previous research and proposes to model distributions of word vectors in documents or document classes. This extends the naive approach to deriving document representations by average pooling and explores the possibility of modeling documents in the semantic space. Experiments on the sohu text database confirmed that the new approach may produce better performance on document classification.**

I. Introduction

The increasingly accumulated text documents on the Internet require effective document classification techniques. A typical document classification system is composed of four components: text pre-processing, document vector extraction, discriminative modeling and document classifier. Among these components, the document vector extraction component is particularly important. Since different documents may involve different numbers of words, it is not trivial to represent a variable-length document as a fixed-length vector while keeping the most class discriminant information.

A popular approach to document vector extraction is based on various topic models. This approach first represents a document as a raw vector (e.g., TF-IDF), and then learns a group of topics $\mathcal{B} = \{\beta_1, \beta_2, ...\beta_K\}$ based on the raw vectors of a large corpus. A document is then represented by the image vector of its raw vector projected onto the topic group $\mathcal{B}$. Typical methods based on topic models include Latent Semantic Indexing (LSI) [1] and its probabilistic version, probabilistic Latent Semantic Indexing (pLSI) [2]. A more comprehensive approach is based on the Latent Dirichlet Allocation (LDA) model [3], which places a Dirichlet distribution as a prior on the topic distribution over $\mathcal{B}$, and a document is represented by the posterior distribution that the document belongs to topics in $\mathcal{B}$. The LDA-based approach usually delivers highly competitive performance on a number of NLP tasks including document classification, partly due to its nature of embedding documents in the low-dimensional semantic (topic) space.

Despite the considerable success on document classification, the LDA model suffers from a number of disadvantages. First, LDA learns semantic clusters based on word co-occurrences, which ignores semantic relevance among words and therefore is still a bag-of-words model. Second, the learning and inference process is based on variational Bayesian approximation, which is sensitive to the initial condition and is fairly slow. Third, the topics learned by LDA is highly determined by word frequencies, which leads to difficulty in learning with less frequent but important topics.

In the previous study [4], we proposed a document classification approach based on word vectors and achieved very promising results. Word vectors are continuous representations of words derived from certain word-to-vector (W2V) model such as a neural network. By this representation, semantic or syntactic related words are located close to each other in the word vector space [5]. A seminal work on word vectors is conducted by Bengio and colleagues when studying neural language modeling [6], and the following work involves various W2V models and efficient learning algorithms [7], [8], [9], [10], [11]. Recently, word vectors have been applied to a multitude of applications, including sentiment classification [12], biometric name entity extraction [13], dependency parsing [14], synonyms recognition [15].

Although word vectors have exhibited great potential on document classification in our previous work [4], the previous methods were rather simple. Particularly, we used the simple average pooling to extract document vectors, which ignores the distribution of word vectors in a class or a document, leading to less representative document vectors. This paper extends the research on W2V-based document classification. Basically, we argue that the distribution of word vectors is a more appropriate representation than the pooled centroid for a document or a class. Two models are presented in this paper: a class specific Gaussian mixture model (CSGMM) which models word vectors of a document class, and a semantic space allocation (SSA) model which represents a document as a posterior probability over a global GMM components in the word vector space.

The rest of the paper is organized as follows: Section II introduces the W2V-based document classification and compares it with the LDA-based approach; Section III presents the proposed statistical models for word vectors. The experiments are presented in Section IV, and the paper is concluded by Section V.

978-616-361-823-8 © 2014 APSIPA

II. W2V-based document classification

A. Word vectors

Word representation is a fundamental problem in natural language processing. The conventional one-hot coding represents a word as a sparse vector of size $|V|$ and all the dimensions are zeros except the one that corresponds to the word. This simple presentation is discrete, high-dimensional and does not embed any semantic relationship among words. This often leads to much difficulty in model training and inference, for example, the well-known smoothness problem in language modeling [16].

An alternative approach embeds words in a low-dimensional continuous space where *relevant* words are close to each other. The 'relevance' might be in the sense of semantic meanings, syntactic roles, sentimental polarities, or any others depending on the model objectives [7], [8], [9], [10], [11]. This dense and continuous word representation, often called word vector (WV), offers a multitude of advantages: first, the dimension of the vector is often much lower than the one-hot coding, leading to much more efficient models; second, the word vector space is continuous, offering the possibility to model texts using continuous models; third, the relationships among words are embedded in their word vectors, providing a simple way to compute aggregated semantics for word collections such as paragraphs and documents. For these reasons, word vectors have been quickly adopted by the NLP community and have been applied to a multitude of text processing tasks [17], [13], [14], [15].

A simple and efficient W2V model that we choose in this work is the skip-gram model [18], where the training objective is to predict the left and right neighbours given a particular word, as shown in Fig. 1. This model is a neural network where the input is a token of word w_i, denoted by e_{w_i}. This input token is mapped to its word vector c_{w_i}, by looking up a embedding matrix U. This word vector, c_{w_i}, is then used to predict the word vectors of its left and right C neighbouring words ($C = 2$ in Fig. 1).

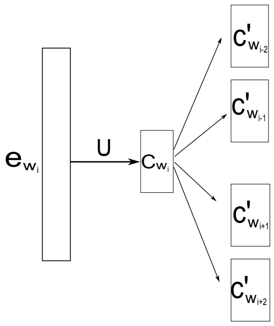

Fig. 1. Architecture of the skip-gram model.

Given a word sequence $w_1, w_2 ... w_N$, the training process maximizes the following objective function by optimizing the embedding matrix U and the weights of the neural network:

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{-C \le j \le C, j \ne 0} logP(w_{i+j}|w_i)$$

where

$$P(w_{i+j}|w_i) = \frac{\exp(c_{w_{i+j}} c_{w_i})}{\sum_w \exp(c_w c_{w_i})}.$$

B. From word vector to document vector

Document classification largely relies on quality of the document representations, or document vectors. The simple vector space model (VSM) represents a document as a raw TF-IDF vector, and the LDA-based approach, as well as other approaches based on topic models, represents a document as an image of the document's raw vector on a topic group $\mathcal{B}$. Due to the semantic interpolation of the topics, LDA can learn semantic relationships among words and thus usually delivers considerably good performance on document classification [3].

The concept of word vectors offers a new approach to deriving document vectors. Since word vectors represent semantic meanings of words[1], and the meaning of a document can be regarded as an aggregation of meanings of the words it involves, document vectors can be derived from word vectors.

In the previous study [4], a simple average pooling approach was proposed to derive document vectors from word vectors. Letting $c_{i,j}$ denote the word vector of the j-th word token of document i, the document vector v_i can be computed as:

$$v_i = \frac{1}{J_i}\sum_{j=1}^{J_i} c_{i,j} \tag{1}$$

where J_i is the number of word tokens in the document.

Both the LDA-based approach and W2V-based approach represent a document as a low-dimensional continuous vector, and both embed semantic meanings in the vector. However, there are a number of fundamental differences between them. First, the LDA model learns the semantic group (topics) from a collection of documents, and thus the topics are specific to the training corpus; the W2V model learns the semantic embedding for each word, and hence the semantic meaning is 'general' for all documents in the same language. Second, the LDA model extracts topics by inferring shared patterns of word co-occurrences, hence purely frequency-driven; the W2V model, in contrast, extracts word semantic meanings by looking at context similarities, thus beyond a bag-of-words model. Third, the LDA-based approach derives document vectors from global topics and so can be regarded as a top-down approach, whereas the W2V-based approach derives document vectors from word vectors hence a bottom-up approach.

A number of experiments were conducted in [4] to investigate potential of the W2V-based approach and compare it with the LDA-based approach. The results demonstrated that, the W2V-based approach, even if represented by the simple average pooling, can deliver highly competitive performance. Actually, the W2V-based approach considerably outperformed the LDA baseline on a 9 classification task, based on a public database provided by sohu research center and using the naive Bayesian (BN) classifier.

III. Document classification based on word vector distributions

The average pooling approach (Eq. (1)) derives a document vector as the centroid of the word vectors that the document involves, which is simple and efficient but not ideal. An obvious disadvantage is that the distribution of the word vectors of a class or a document is overlooked. We argue that

[1] To make it more precise, word vectors learned by the skip-gram model encode both semantic meanings and syntactic roles, though we do not differentiate them in this paper.

the nuance semantic meaning of a class or a document should be represented by the distribution of the word vectors that are involved, and so document classification should be conducted by modeling these distributions.

This section provides two classification approaches based on word vector distributions. The first approach ignores the document boundary and models the word vectors of each class as a Gaussian mixture model (GMM); the classification is then cast to a task of maximum posterior inference. The second approach constructs a global GMM and derives a document vector as the posterior probabilities over the GMM components.

A. Class-specific GMM (CSGMM)

In this model our assumption is that the word vectors of a document class follow a Gaussian mixture distribution and can be modeled by a class-specific GMM (CSGMM). Let the number of classes be K, and the number of Gaussian components of each CSGMM to be M. The probability of a word vector $c_{i,j}$ given by the CSGMM of class k is written by:

$$p_k(c_{i,j}) = \sum_m \pi_{k,m} N(c_{i,j}; \theta_{k,m}) \tag{2}$$

where $\theta_{k,m}$, $\pi_{k,m}$ are the Gaussian parameters and the prior probability of the m-th component, respectively. These model parameters can be estimated with the maximum likelihood (M-L) criterion. Specifically, it involves maximizing the following likelihood function:

$$L_k(\{\theta_{k,m}\}, \{\pi_{k,m}\}) = \prod_{i \in \Delta_k} \prod_j \sum_m \pi_{k,m} N(c_{i,j}; \theta_{k,m})$$

where Δ_k represents the training documents of class k. This optimization problem can be effectively solved by an expectation-maximization (EM) procedure.

Once the CSGMMs are well trained, the class of a test document d can be determined in a maximum posterior fashion, formulated as:

$$l(d) = \arg\max_k P(k|d)$$

where $P(k|d)$ is the posterior probability that d belongs to class k:

$$\begin{aligned} P(k|d) &= \frac{p(d|k)}{\sum_r p(d|r)} \\ &= \frac{\prod_{c_j \in d} p_k(c_j)}{\sum_r \prod_{c_j \in d} p_r(c_j)} \end{aligned} \tag{3}$$

where $p_k(c)$ is computed by Eq. (2). Note that this is a purely generative approach, and the classification is based on the generative models (CSGMMs) directly. Therefore, there are no document vectors derived, and no additional discriminative models used for classification.

B. Semantic space allocation (SSA)

A potential problem of the CSGMM approach is that the document boundaries are ignored, which may lose some document specific patterns. In addition, the pure generative modeling itself tends to be less discriminative in classification tasks. A possible improvement is to build a global GMM, and then derive document vectors from the GMM components. A discriminative classifier is finally employed to conduct classification with the derived document vectors.

Specifically, a global GMM is built on the word vector spaces with all the word vectors in the training set, without considering the document boundaries and the class labels. The parameters are estimated by maximizing the following likelihood function:

$$L(\{\theta_m\}, \{\pi_m\}) = \prod_k \prod_{i \in \Delta_k} \prod_j \sum_m \pi_m N(c_{i,j}; \theta_m)$$

where m is the number of Gaussian components, and $\{\theta_m\} and \{\pi_m\}$ are model parameters to be estimated.

Given the global GMM well trained, a document d can be represented by the posterior probabilities that d belongs to GMM components:

$$v = [P(1|d), P(2|d), ..., P(M|d)]^T$$

where $P(m|d)$ is the posterior probability that d belongs to component m, and is computed by Eq. (3).

This model is similar to LDA in the sense that both of them rely on a global semantic representation: in LDA this is the topic group $\mathcal{B}$, and in our model this is the GMM. In addition, both the two models are generative and sample a semantic component from the group for every word when deriving document vectors. The difference is also clear: the LDA model samples discrete word tokens following a multinomial distribution, while our model samples continuous word vectors following a Gaussian distribution. For this reason, we call our model as semantic space allocation (SSA). We note that the LDA model places a prior on the conditional distribution and therefore is a Bayesian approach, whereas our model is a maximum likelihood approach. The Bayesian SSA will be published elsewhere.

IV. Experiments

A. Data and configurations

The experiments were conducted with a text database published by sohu research center[2]. This database involves 9 classes of web documents, including Chinese articles in the area of automobile, IT, finance, health, sports, tour, education, recruitment, culture and military. The total number of documents amounts to 16110, from which we selected 14301 documents (1589 per class) for model training, and the rest 1809 documents for test. The same database has been used in our previous work [4].

The training and test documents were first purified by removing some unrecognized characters, and then were segmented into words by the SCWS word segmentation tool[3]. The dictionary used for the word segmentation consists of $150,000$

[2] http://www.sogou.com/labs/dl/c.html

[3] http://www.xunsearch.com/scws/index.php

Chinese words. The word2vec tool provided by Google[4] was used to train the skip-gram W2V model and produce word vectors. A tool provided by Blei[5] was used to train the LDA model and conduct inference.

We experimented with a multitude of discriminative models, including naive Bayesian, k-NN and SVM. The naive Bayesian and k-NN model were trained using the weka toolkit[6], and the SVM model was trained using the scikit-learn tool[7].

B. Average pooling

The first experiment examines the average pooling approach. This approach has been studied in [4] where the naive Bayesian model was used as the classifier. We experiment this approach with different classifiers in this study, and the results in terms of classification precession rate are reported in Table I. The dimension of the word vectors and the number of the topics in LDA are all set to 50, and hence the dimensions of the document vectors derived using these two approaches are equal to 50. From the results in Table I, we observe the W2V-based approach outperforms the LDA-based approach significantly, no matter which classifier is used. When the three classifiers are compared, the k-NN method outperforms the other two. Nevertheless, k-NN is a non-parametric approach and hence does not apply to large training data, so we choose SVM as the classifier in the following experiments.

TABLE I
CLASSIFICATION PRECESSION WITH AVERAGE POOLING

Classifier	W2V	LDA
NB	72%	63%
k-NN	83.91%	81.21%
SVM	83.91%	70.04%

C. CSGMM and SSA

Fig. 2 and Fig. 3 report the results of the two proposed approaches: the CSGMM model and the SSA model. The x-axis represents the number of Gaussian components M in the CSGMMs or the global GMM, and the y-axis represents the classification precession rate. For comparison, the performance of the LDA and W2V baseline (average pooling) are also presented. In addition, we also experiment a hybrid approach which concatenates the document vectors derived from the SSA model and by the average pooling approach, whose results are also shown in Fig 3.

We can observe that the CSGMM approach tends to be less effective than the W2V baseline, even inferior to the LDA-based approach. This may be attributed to the generative nature of the model, which leads to inferiority on classification tasks. However, this approach enjoys the advantage with dynamic classes: for any newly added class, its GMM can be trained separately and added into the class group without difficulty. This is not possible for other approaches as the SVM needs to be fully re-trained.

The SSA model works well and may outperform the W2V pooling approach with a sufficiently large M. This is highly promising and indicates that the distributions of word vectors might be a better representation for documents. On the other hand, the gain that the SSA model obtained is marginal, and at the cost of more computation. The hybrid approach reduces the gap between the SSA and pooling baseline when M is small, but does not improve the best performance in a significant way. It seems that the W2V baseline is rather hard to compete unless more powerful modeling techniques are applied, such as the Bayesian SSA approach.

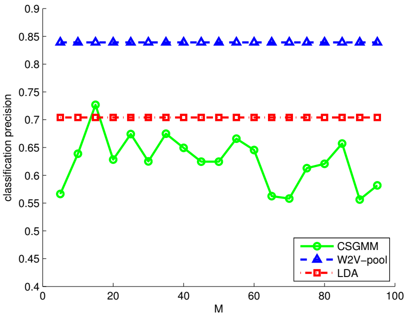

Fig. 2. Performance of the CSGMM model.

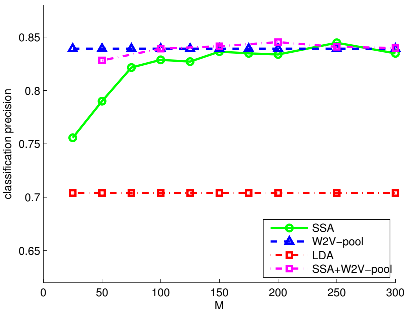

Fig. 3. Performance of the SSA model.

V. CONCLUSIONS

This paper is a following work on W2V-based document classification. We argue that the distributions of word vectors within a document or a class provide a good semantic representation for the document or the class. Two approaches are proposed: the CSGMM approach models word vectors of a class with a class specific GMM and conducts document classification on these models, and the SSA model derives document vectors as posterior probabilities over the components of a global GMM. The experimental results show that the CSGMM approach works generally not as well as the average pooling baseline; however it is superior in the situation of dynamic classes. The SSA model delivers better performance than the average pooling baseline. Although marginal, this gain suggests that modeling distributions of word vectors is a promising research direction for document classification and related applications.

VI. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation of China (NSFC) under the project No. 61271389. It was also partially supported by the Huilan Ltd.

[4] https://code.google.com/p/word2vec
[5] http://www.cs.princeton.edu/ blei/lda-c
[6] http://www.cs.waikato.ac.nz/ml/weka
[7] http://scikit-learn.org/stable/index.html

References

[1] S. T. Dumais, “Latent semantic analysis,” *Annual review of information science and technology*, vol. 38, no. 1, pp. 188–230, 2004.

[2] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[4] R. Liu, D. Wang, and C. Xing, “Document classification based on word vectors,” in *ISCSLP'14*, 2014.

[5] G. Hinton, J. McClelland, and D. Rumelhart, *Distributed representations*, ser. Parallel dis-tributed processing: Explorations in the microstructure of cognition. MIT Press, 1986, ch. 1.

[6] Y. Bengio, H. Schwenk, J.-S. Senécal, F. Morin, and J.-L. Gauvain, “Neural probabilistic language models,” in *Innovations in Machine Learning*. Springer, 2006, pp. 137–186.

[7] A. Mnih and G. Hinton, “Three new graphical models for statistical language modelling,” in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 641–648.

[8] A. Mnih and G. E. Hinton, “A scalable hierarchical distributed language model.” in *NIPS*, 2008, pp. 1081–1088.

[9] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[10] T. Mikolov, “Statistical language models based on neural networks,” Ph.D. dissertation, Ph. D. thesis, Brno University of Technology, 2012.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *Computation and Language*, 2013.

[12] A. L. Maas and A. Y. Ng, “A probabilistic model for semantic word vectors,” in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[13] B. Tang, H. Cao, X. Wang, Q. Chen, and H. Xu, “Evaluating word representation features in biomedical named entity recognition tasks,” *BioMed research international*, vol. 2014, 2014.

[14] M. Bansal, K. Gimpel, and K. Livescu, “Tailoring continuous word representations for dependency parsing,” in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2014.

[15] T. Dao, S. Keller, and A. Bejnood, “Alternate equivalent substitutes: Recognition of synonyms using word vectors,” 2013.

[16] R. ROSENFELD, “Two decades of statistical language modeling: Where do we go from here?” *PROCEEDINGS OF THE IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.

[17] T. Mikolov, Q. V. Le, and I. Sutskever, “Exploiting similarities among languages for machine translation,” *Computation and Language*, 2013.

[18] T. M. amd Ilya Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *NIPS 2013*, 2013, pp. 3111–3119.