Encoding Dual Semantic Knowledge for Text-Enhanced Cloud Services

Shicheng Cui[1], Qianmu Li[1,2(✉)], Shu-Ching Chen[3], Jun Hou[4], Hanrui Zhang[1], and Shunmei Meng[1]

[1] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
qianmu@njust.edu.cn
[2] Intelligent Manufacturing Department, Wuyi University, Jiangmen 529020, China
[3] School of Computing and Information Sciences, Florida International University, Miami, FL 33199, USA
[4] Nanjing Vocational University of Industry Technology, Nanjing 210023, China

Abstract. Topic modeling techniques have been widely applied in many cloud computing applications. However, few of them have tried to discover latent semantic relationships of implicit topics and explicit words to generate a more comprehensive representation for each text. To fully exploit the semantic knowledge for text classification in cloud computing systems, we attempt to encode topic and word features based on their latent relationships. The extracted topical information reorganizes the original textual structures from two aspects: one is that the topic extracted by Latent Dirichlet Allocation (LDA) is viewed as a textual extension; the other is that the topic feature performs as a counterpart modality to the word. This paper proposes a Dual Semantic Embedding (DSE) method, which uses Convolutional Neural Networks (CNNs) to encode the dual semantic features of topics and words from the reorganized semantic structures. Experimental results show that DSE improves the performance of text classification and outperforms the state-of-the-art feature generation baselines on micro-F_1 and macro-F_1 scores over the real-world text classification datasets.

Keywords: Text classification · Dual Semantic Embedding · Convolutional Neural Networks

1 Introduction

Artificial Intelligence (AI) has been widely adopted in many real-world cloud computing services. Text classification [1] plays an important role in recent AI cloud computing platforms. For example, in some online recommendation systems [6,36], there are large quantities of reviews of items needed to be analyzed to estimate the sentiment [27] or to classify the items based on the textual information in cloud environments. Another interesting example is that in social networks [41], the techniques of text classification provided by some

L. Qi et al. (Eds.): CloudComp 2020, LNICST 363, pp. 146–159, 2021.
https://doi.org/10.1007/978-3-030-69992-5_12

social computing infrastructures, can help node classification [9,10,15,29] and link prediction [8,45]. With the rapid development of the deep learning techniques [13,21,22,31,32,38], plenty of AI applications have achieved great success. As a key component in intelligent cloud services, text classification still draws great attention. Thus, this study attempts to apply deep learning techniques to improve the effectiveness of text classification, which could help a large amount of intelligent data analytics in cloud computing systems.

In many Vector Space Model (VSM) [33] based methods, each text is represented as a vector in which each element indicates the frequency of a word appearing in the text. Although VSM is easy to comprehend and implement, there are some drawbacks in this model. The first drawback is the high dimensionality problem. VSM requires a dictionary that contains all words in a corpus. Usually, the form that represents a word is one-hot representation and the dimensionality of a dictionary is more than ten thousand magnitude, sometimes even higher. Next, it has the data sparseness problem, meaning that the words amongst a text only use a small portion of the vocabulary, but the whole dictionary is needed. Last but not the least, the format of the bag-of-words [16] in VSM causes semantic information loss. All of these result in high computational complexity, tremendous memory consumption, and lower accuracy of classification in VSM.

To deal with the aforementioned problems, dimensionality reduction methods were proposed. χ^2 statistics [43] and Information Gain (IG) [2,42] are two classical methods that select important terms in texts to build the vocabulary of keywords. These methods and their variants [34,37,40] indeed reduce the dimensionality and data sparseness to some extent. However, they still fail to consider the semantic relationships among words and phrases and only use the extrinsic structure of the texts. Hence, several semantic-based feature generation methods [5,7,11,18,30,35,39] have been proposed. Latent Dirichlet Allocation (LDA) [5,17] topic modeling method is one of the most classical semantic methods. It extracts latent topic features from words in the content with the assumption that each text is generated by a group of discriminative latent topics. Different from the aforementioned methods, the neural network based approaches [4,19,20,24,25,28,46] try to learn deeper semantics in the corpora. Continuous-Bag-Of-Words (CBOW) and skip-gram methods [24,25] are proposed for computing continuous vector representations of words from very large datasets and attempt to preserve the linear regularities among words. The Convolutional Neural Networks (CNNs) based methods [19,46] that represent the sentences or documents as a matrix perform efficiently in natural language processing.

However, to our best knowledge, most existing text classification approaches employed in popular intelligent cloud computing platforms neglect the latent semantic relationships of the topic and word features. Besides, the novel CNN-based ones lack the means of integrating features with diverse semantics, and fail to capture the cross semantic interactions among those extracted features. They only consider one single semantic information, such as word layer or sentence layer semantics, which might confine the usage of other meaningful textual

features. Hence, this paper proposes Dual Semantic Embedding (DSE) method that attempts to extract the dual semantic features from topics and words for text classification using CNNs. Two structure-based fusion strategies are provided for reconstructing the textual information: one is that we directly extend the original contents by the topics; the other is that we generate the topic-based corpus by treating the topic as a counterpart feature to the corresponding word. We encode the dual semantic information into a shared representation from the reorganized semantic structures. Experimental results show that DSE can better represent the characteristics of each text.

The rest of the paper is organized as follows. Section 2 introduces text mining techniques that are involved in our method. In Sect. 3, the pipeline of the DSE method is presented. Next, DSE is evaluated over several state-of-the-art baselines and the detailed experiments are given in Sect. 4. Finally, we conclude our work and point out the future work in Sect. 5.

2 Preliminaries

In this section, we present the preliminaries of LDA [5,17] and skip-gram [24,25].

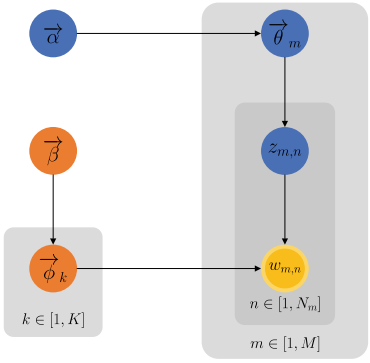

Fig. 1. The Bayesian network of LDA.

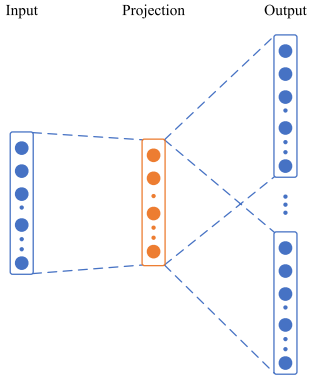

Fig. 2. Skip-gram framework.

2.1 LDA

To determine the distribution of latent topics, we need to construct a topic model. LDA is a probabilistic generative model that is able to estimate the probabilities of multinomial observations by utilizing the co-occurrence structure of words in documents to recover the latent topic structure without using any background knowledge [14]. It assumes that there are K independent topics shared by M documents, where document doc_m has N_m words (m=1 ..., M). Each topic is a polynomial probabilistic distribution of words, and each document

is randomly generated by those topics. According to the Bayesian network of LDA shown in Fig. 1, the process of $\overrightarrow{\alpha} \rightarrow \overrightarrow{\theta}_m \rightarrow z_{m,n}$ denotes that a topic indicator $z_{m,n}$ is sampled by topic proportion $\overrightarrow{\theta}_m$ from a Dirichlet distribution with hyperparameter $\overrightarrow{\alpha}$, and the process of $\overrightarrow{\beta} \rightarrow \overrightarrow{\phi}_k \rightarrow w_{m,n}|k = z_{m,n}$ denotes that topic-specific word $w_{m,n}$ is emitted by the corresponding topic-specific term distribution $\overrightarrow{\phi}_k$ from a Dirichlet distribution with hyperparameter $\overrightarrow{\beta}$. For a corpus, $\boldsymbol{W} = \{\overrightarrow{w}_m\}_{m=1}^{M}$, the joint distribution is defined as follows.

$$p(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{\Phi}) = \prod_{m=1}^{M} \prod_{n=1}^{N_m} \sum_{k=1}^{K} p(w_{m,n}|\overrightarrow{\phi}_k) p(z_{m,n} = k|\overrightarrow{\theta}_m), \tag{1}$$

where $\boldsymbol{\Theta} = \{\overrightarrow{\theta}_m\}_{m=1}^{M}$ and $\boldsymbol{\Phi} = \{\overrightarrow{\Phi}_k\}_{k=1}^{K}$. The gibbs sampling method is used for approximate inference in LDA. The dimension values z_i of the distribution are sampled once at a time, based on all the other dimension values $z_{\neg i}$. Let V be the vocabulary size, and then the estimations of parameters $\hat{\theta}_{m,k}$ and $\hat{\phi}_{k,v}$ are derived as follows.

$$\begin{aligned} \hat{\theta}_{m,k} &= \frac{n_{m,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K} (n_{m,\neg i}^{(t)} + \alpha_k)}, \\ \hat{\phi}_{k,v} &= \frac{n_{k,\neg i}^{(v)} + \beta_v}{\sum_{v=1}^{V} (n_{k,\neg i}^{(v)} + \beta_v)}. \end{aligned} \tag{2}$$

2.2 Skip-Gram

Given a corpus $\mathbf{W}$, the objective under skip-gram is defined to maximize the following conditional probability and map each word w to a d-dimensional representation $\mathbf{x} \in \mathbb{R}^{d \times V}$, where V is the vocabulary size.

$$\arg\max_{\mathbf{x}} \prod_{w \in \mathbf{W}} \prod_{\Gamma \in \mathcal{C}(w)} p(\Gamma|w; \mathbf{x}) \tag{3}$$

where Γ is a context, and $\mathcal{C}(w)$ is the set of contexts of word w.

As shown in Fig. 2, there are three layers in the skip-gram framework: input, projection and output layers. The projection layer is the hidden layer of skip-gram that takes the embedding of w as the input and learns output embeddings of words in the contexts. Either hierarchical softmax or negative sampling can be applied to training process.

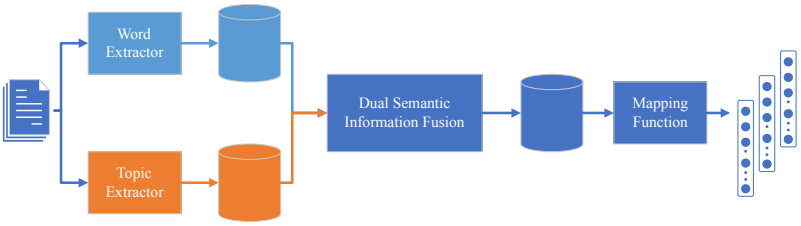

Fig. 3. An Illustration of DSE framework. At first, the topic and word extractors are used to preprocess the original text corpus. Then, the topic and word features are fused in the next step, where we pretrain the topic and word embeddings. After that, CNNs are applied to extract the dual semantic information by mapping the topic and word features into a shared representation.

3 Method

We present the detailed pipeline of DSE method that extracts the dual semantic features of topics and words. Figure 3 illustrates the framework of DSE.

3.1 Pretrained Embeddings

The pretrained word vectors have benefited a large number of applications. Inspired by that, we introduce skip-gram to pretrain the topic and word embeddings. Since applying LDA would be easy to obtain topics from a given corpus, we will focus on the strategy of generating embeddings with topic-word semantics. Suppose that after the LDA process, we have learned the topic set $\mathcal{T} = \{1, 2, ..., K\}$. We assign each word w the top_1 topic $t^w \in \mathcal{T}$ inferred by LDA.

$$\arg\max_{k} \bigcup_{k=1}^{K} \left(\frac{n_{k,\neg i}^{(w)} + \beta_w}{\sum_{v=1}^{V} (n_{k,\neg i}^{(v)} + \beta_v)} \right). \tag{4}$$

Two fusion strategies are provided to learn the embeddings with dual semantic knowledge for both topics and words based on the context-related structure. The key idea is that topics can be treated as words, which is commonly used in text enriching [23,30,44].

Mixed Strategy: As shown in Fig. 4, the mixed strategy makes the latent topic embedded directly after its corresponding word. For example, the topics of the content "a cat is sleeping" are $\{9, 5, 2, 7\}$. Based on the strategy, "a 9 cat 5 is 2 sleeping 7" would be fed into skip-gram.

Split Strategy: Figure 5 presents the split strategy. It simply yields topic sequences based on the word sequences and feeds two sequences into skip-gram separately.

We believe that these two strategies can generate meaningful embeddings for topics and words. Let τ denote both topics and words here. Due to the strategy

of skip-gram which utilizes τ to predict $\mathcal{C}(\tau)$, the training set contains all the pairs of (γ, τ), where γ is a topic or a word in the context $\Gamma \in \mathcal{C}(\tau)$. That is, by applying the mixed strategy, we can conclude that a topic is affected by its surrounding topics and words, or vice versa. For the Split strategy, as the counterpart modal features to the words, the topic embeddings depict the sequential information of words and the semantic interactions among the topics. Hence, no matter which strategy is applied, the pretrained embeddings will learn the dual semantic information with the interactions among the topic and word features after the skip-gram process. In order to derive the pretrained embeddings more efficiently, negative sampling is applied.

$$\arg\max_{\lambda} \sum_{(\gamma,\tau)\in D} \log \sigma(\boldsymbol{v}_\gamma \cdot \boldsymbol{v}_\tau) + \sum_{(\gamma',\tau')\in D'} \log \sigma(-\boldsymbol{v}_{\gamma'} \cdot \boldsymbol{v}_{\tau'}), \tag{5}$$

where D denotes the set of pairs extracted using the strategies, D' denotes the generated set of random pairs which are assumed to be negative examples, $\sigma(x) = \frac{1}{1+e^{-x}}$, $\boldsymbol{v}_\gamma$ and $\boldsymbol{v}_\tau$ are the vectors of γ and τ respectively, and λ is the parameter in the neural network.

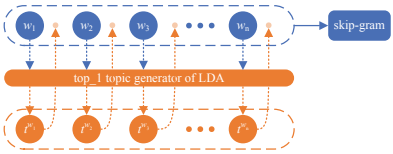

Fig. 4. The mixed strategy.

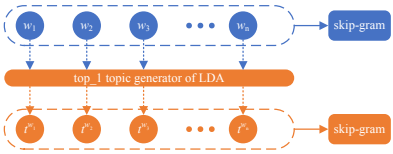

Fig. 5. The split strategy.

3.2 Encoding Dual Semantic Information

Two CNNs are designed to implement the aforementioned fusion strategies, with different structures of the shared representations. The multimodal joint representations based on multimodal deep learning [26] are used to form the shared representations. The key idea is that the technique of multimodal joint representations projects multimodal data into a common space, and is suited for scenarios when all of the modalities are present during inference [3]. Hence, we consider the topic and word features as different modalities and project them into a common feature space. Two architectures are shown in Fig. 6 and Fig. 7, respectively.

The inputs of the CNN in Fig. 6 are the shared representations, and the order of the sequence follows the description of the mixed strategy. That is,

$$\mathcal{I} = [\boldsymbol{w}_1, \boldsymbol{t}^{w_1}, \boldsymbol{w}_2, \boldsymbol{t}^{w_2}, ..., \boldsymbol{w}_l, \boldsymbol{t}^{w_l}]^{\mathsf{T}}, \tag{6}$$

where $\mathcal{I}$ is the 2-dimensional input, l is the length of the content. $\boldsymbol{w}_i$ and $\boldsymbol{t}^{w_i}$ are the pretrained embeddings of word w_i and topic t^{w_i} respectively, where t^{w_i} denotes the top_1 topic of w_i.

$$\boldsymbol{v}_{shared} = \boldsymbol{v}_w \oplus \boldsymbol{v}_t. \tag{7}$$

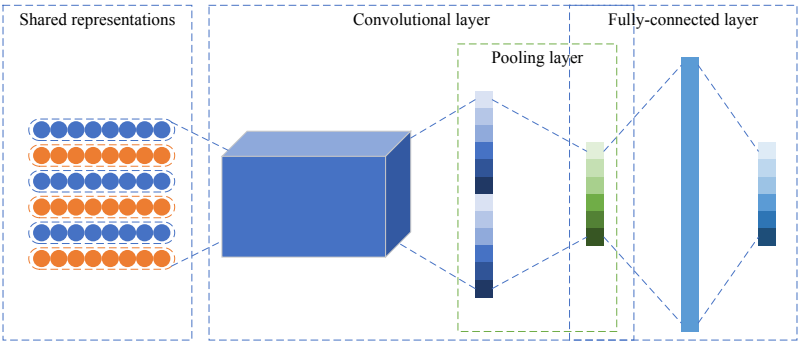

Fig. 6. The proposed CNN architecture related to the mixed strategy.

Different from the architecture in Fig. 6, Fig. 7 presents a Bi-CNN architecture, which processes word matrices and topic matrices respectively. The shared representations are used in the middle of the fully-connected layer, which combines the hidden vectors of words and topics. That is,

In general, a convolutional operation involves a kernel $\mathbf{k} \in \mathbb{R}^{n \times m}$, which is used to produce a new feature from $n \times m$ pixels. It is reasonable to use the kernels whose width is equal to the dimensionality d of the pretrained embeddings (i.e., $m = d$) because the rows in matrices represent the discrete topics and words. Suppose the kernel receives n adjacent rows of embeddings $\mathbf{x}_{i:i+n-1}$ each time, and the sub-matrix of $\mathbf{E}$ is $\mathbf{E}[i : i + n - 1]$. A feature c_i is generated by:

$$c_i = f(\mathbf{k} \cdot \mathbf{E}[i : i + n - 1] + \mathtt{b}), \tag{8}$$

where f is an activation function such as ReLU [12] and *tanh* (i.e., the hyperbolic tangent) [19,46], and $\mathtt{b}$ denotes a bias term.

To generate a better semantic representation from the text, a fully-connected layer is inserted into both CNN architectures. One is used for encoding the max-pooling results, while another one unifies the shared representations into a common space.

The cross-entropy loss function (as shown in Eq. (9)) is applied to optimize the CNNs for multi-class classification tasks.

$$\mathcal{L} = -\sum_{label=1}^{N} y_{label,o} \log(p_{label,o}), \tag{9}$$

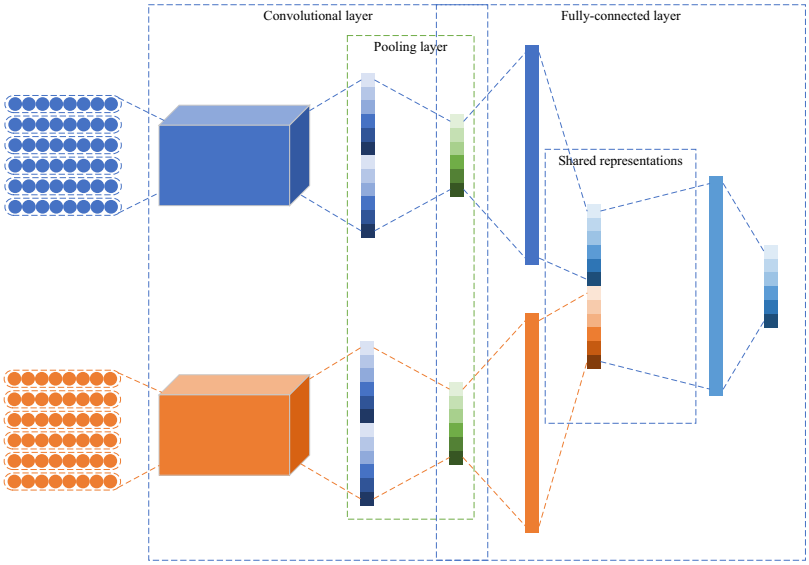

Fig. 7. The proposed CNN architecture related to the split strategy.

where $y_{label,o}$ denotes a binary indicator (0 or 1) representing whether class *label* is correct for observation *o*. $p_{label,o}$ is the predicted probability of observation *o* that how likely it belongs to class *label*.

4 Experimental Results

The performance of DSE is validated on several real-world datasets. The experimental results demonstrate the effectiveness of the proposed DSE for text classification tasks.

Table 1. Statistics of the datasets

Datasets	20 Newsgroups	Reuters-r8	WebKB
#Training samples	11,293	5,485	2,785
#Testing samples	7,527	2,189	1,383
#Labels	20	8	4

4.1 Datasets

The following datasets are used in the experiments. All datasets have been preprocessed and split into training sets and testing sets from the source, where some samples with missing textual values have been filtered. Table 1 lists the detailed statistics of these datasets.

20 Newsgroups. This dataset comprises of approximately 20,000 newsgroup documents, partitioned across 20 different groups. It is a popular dataset in machine learning for text analysis.

Reuters-r8. It is a widely used text dataset extracted from Reuters-21578. Those documents are assembled and indexed with 8 categories.

WebKB. This dataset contains WWW-pages collected from the computer science departments of various universities. The number of samples is around 4,000 divided into 4 classes.

4.2 Metrics

The Linear SVM (which is implemented using scikit-learn) is used as a classifier for both DSE and the state-of-the-art baselines on the benchmarks. The metrics used for performance evaluation include micro-F_1 and macro-F_1.

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}, \tag{10}$$

where P denotes precision and R denotes recall.

- micro-F_1. This method sums up the individual true positives, false positives, and false negatives of the dataset for different classes, which can be used for imbalanced data problems.
- macro-F_1. It calculates the average of the precision and recall of the dataset on different classes, and finds their unweighted mean, which does not take label imbalance into account.

4.3 Baselines

The performance of DSE is compared with the state-of-the-art baseline methods as follows. In our experiments, for reducing the external factors and just focusing on the method itself, the word embeddings for all methods were pretrained.

- Doc2Vec [20]. This method aims at constructing a representation of a document, regardless of its length. It takes the documents in a corpus as the inputs and produces a vector space where each document in the corpus is assigned a corresponding vector in the space.
- TextCNN [19]. This method directly processes the intrinsic structure of each document using a CNN with little hyperparameter tuning and pretrained word vectors.

Table 2. Results of multi-classification in 20 Newsgroups on varying the dimensionality of document representations

	Dimensionality	DSE-m	DSE-s	Doc2Vec	TextCNN
micro-F_1 (%)	100	63.332	66.946	42.939	66.826
	200	63.226	66.401	33.479	66.109
	300	61.791	**67.743**	25.282	65.498
macro-F_1 (%)	100	63.065	66.555	42.223	66.091
	200	62.965	65.954	33.205	65.750
	300	61.409	**67.106**	25.021	65.305

Table 3. Results of multi-classification in Reuters-r8 on varying the dimensionality of document representations

	Dimensionality	DSE-m	DSE-s	Doc2Vec	TextCNN
micro-F_1 (%)	100	93.924	93.878	66.880	93.513
	200	94.381	92.005	62.814	93.741
	300	**95.021**	94.427	59.662	92.828
macro-F_1 (%)	100	81.468	78.695	37.008	76.342
	200	79.212	70.782	33.882	76.285
	300	79.690	**81.745**	30.503	74.354

Table 4. Results of multi-classification in WebKB on varying the dimensionality of document representations

	Dimensionality	DSE-m	DSE-s	Doc2Vec	TextCNN
micro-F_1 (%)	100	86.696	86.623	55.242	86.406
	200	86.623	85.900	48.301	86.551
	300	86.406	85.683	47.505	**86.913**
macro-F_1 (%)	100	85.029	85.021	50.683	85.159
	200	85.334	84.713	43.449	85.510
	300	84.435	83.799	43.398	**85.952**

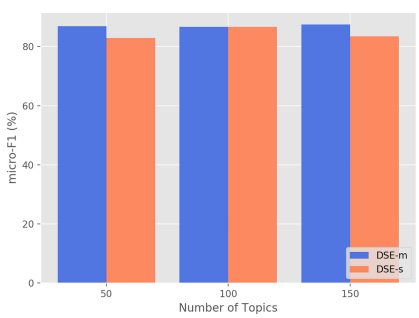

Fig. 8. The micro-F_1 scores on varying numbers of topics.

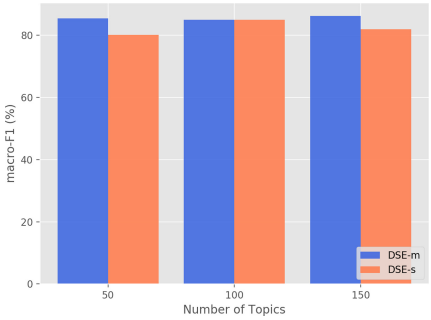

Fig. 9. The macro-F_1 scores on varying numbers of topics.

4.4 Text Classification

For comparison with the baselines, the values of the following parameters of DSE are set. The maximum length $L_{max} = 1000$ for each document, the number of topics $K = 100$, and the size of the pretrained embeddings is set to 100. The kernel size of CNN for the mixed strategy (DSE-m) is $(height = 20, width = 100)$ with $stride = 20$, and half of $height$ and $stride$ parameters are used for the split one (DSE-s). Tables 2, 3, and 4 present the micro-F_1 and macro-F_1 scores of all methods by varying the dimensionality of document representations from 100 to 300 on three benchmarks, where the best scores are highlighted in bold.

On the 20 Newsgroups dataset, DSE-s with 300 dimensions achieves the highest micro-F_1 (67.743%) and macro-F_1 (67.106%) scores. Doc2Vec performs poorly on this dataset and shows an unstable trend with the increasing dimensionality values. The CNN-based methods have all achieved competitive results, but DSE with the split strategy outperforms TextCNN.

For the Reuters-r8 dataset, DSE also achieves the highest micro-F_1 (95.021%) and macro-F_1 (81.745%) scores. Doc2Vec is still affected by the feature dimensionality. TextCNN outperforms Doc2Vec, but there is a significant margin on macro-F_1 values between DSE and TextCNN, where our proposed method improves the results by more than 5%.

The WebKB dataset with four categories has fewer samples than the aforementioned datasets. In this scenario, DSE and TextCNN are considered indistinguishable by varying the dimensionality from 100 to 300. Since different numbers of topics may affect the semantic meaning of the pretrained embeddings, more experiments on the topic sensitivity is conducted in the next subsection.

4.5 Topic Sensitivity

Since DSE learns deep semantics partly from the latent topics, the numbers of topics are varied to evaluate its key parameter sensitivity on the WebKB dataset. The results are shown in Figs. 8 and 9.

It can be seen from the figures that DSE-s is somehow sensitive to the number of topics, while DSE-m seems to be quite stable. The highest micro-F_1 (87.419%) and macro-F_1 (86.179%) scores are achieved by DSE-m with the number of topics $K = 150$, which also outperforms the best results of TextCNN.

5 Conclusion and Future Work

In this paper, the Dual Semantic Embedding (DSE) method is proposed which attempts to encode dual semantic knowledge for text classification. Two CNNs are designed to implement the structure-based fusion strategies for capturing the dual semantic information. Experimental results demonstrate that the generated feature representations with topic-word semantics improve the effectiveness of text classification on micro-F_1 and macro-F_1 scores over the real-world datasets.

In the future, the strategy of coordinated representations will be considered, which learns separate representations for multiple features, instead of projecting the features together into a joint space. Furthermore, we also consider reframing the training process of DSE by applying distributed and parallel cloud computing techniques to improve its learning effectiveness.

Acknowledgment. This work was supported in part by the China Scholarship Council (201706840112), Fundamental Research Funds for the Central Universities (30918012204), Jiangsu province key research and development program (BE2017739), 2018 Jiangsu Province Major Technical Research Project "Information Security Simulation System" (BE2017100), the 4th project "Research on the Key Technology of Endogenous Security Switches" (2020YFB1804604) of the National Key R&D Program "New Network Equipment Based on Independent Programmable Chips" (2020YFB1804600), Military Common Information System Equipment Pre-research Special Technical Project (315075701) and Industrial Internet Innovation and Development Project in 2019 - Industrial Internet Security On-Site Emergency Detection Tool Project.

References

1. Aggarwal, C.C., Zhai, C.: A survey of text classification algorithms. In: Mining Text Data, pp. 163–222. Springer, Berlin (2012)
2. Aghdam, M.H., Ghasem-Aghaee, N., Basiri, M.E.: Text feature selection using ant colony optimization. Expert Syst. Appl. **36**(3), 6843–6853 (2009)
3. Baltrušaitis, T., Ahuja, C., Morency, L.P.: Multimodal machine learning: a survey and taxonomy. IEEE Trans. Pattern Anal. Mach. Intell. **41**(2), 423–443 (2018)
4. Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. **3**, 1137–1155 (2003)
5. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
6. Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowl.-Based Syst. **46**, 109–132 (2013)
7. Chen, E., Lin, Y., Xiong, H., Luo, Q., Ma, H.: Exploiting probabilistic topic models to improve text categorization under class imbalance. Inform. Process. Manag. **47**(2), 202–214 (2011)

8. Cui, S., Li, Q., Chen, S.C.: An adversarial learning approach for discovering social relations in human-centered information networks. EURASIP J. Wireless Commun. Netw. **2020**(1), 172 (2020). https://doi.org/10.1186/s13638-020-01782-6
9. Cui, S., Li, T., Chen, S.C., Shyu, M.L., Li, Q., Zhang, H.: Disl: deep isomorphic substructure learning for network representations. Knowl.-Based Syst. **189**, 105086 (2020). https://doi.org/10.1016/j.knosys.2019.105086
10. Cui, S., et al.: Simwalk: learning network latent representations with social relation similarity. In: 2017 12th International Conference on Intelligent Systems and Knowledge Engineering, pp. 1–6. IEEE (2017)
11. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inform. Sci. **41**(6), 391–407 (1990)
12. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, pp. 315–323 (2011)
13. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep Learning, vol. 1. MIT Press, Cambridge (2016)
14. Gregor, H.: Parameter estimation for text analysis. Technical report (2005)
15. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864. ACM (2016)
16. Harris, Z.S.: Distributional structure. Word **10**(2–3), 146–162 (1954)
17. Hoffman, M., Bach, F.R., Blei, D.M.: Online learning for latent dirichlet allocation. In: Advances in Neural Information Processing Systems, pp. 856–864 (2010)
18. Hofmann, T.: Probabilistic latent semantic analysis. In: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, pp. 289–296. Morgan Kaufmann Publishers Inc. (1999)
19. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751 (2014)
20. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196 (2014)
21. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
22. Li, D., Li, Q.: Adversarial deep ensemble: evasion attacks and defenses for malware detection. IEEE Trans. Inform. Forensics Secur. **15**, 3886–3900 (2020)
23. Liu, Y., Liu, Z., Chua, T.S., Sun, M.: Topical word embeddings. In: AAAI, pp. 2418–2424 (2015)
24. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
25. 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, pp. 3111–3119 (2013)
26. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), pp. 689–696 (2011)
27. Pang, B., Lee, L., et al.: Opinion mining and sentiment analysis. Found. Trends® Inform. Retrieval **2**(1–2), 1–135 (2008)
28. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)

29. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710. ACM (2014)
30. Phan, X.H., Nguyen, C.T., Le, D.T., Nguyen, L.M., Horiguchi, S., Ha, Q.T.: A hidden topic-based framework toward building applications with short web documents. IEEE Trans. Knowl. Data Eng. **23**(7), 961–976 (2011)
31. Pouyanfar, S., Chen, S.C., Shyu, M.L.: An efficient deep residual-inception network for multimedia classification. In: Proceedings of the IEEE International Conference on Multimedia and Expo, pp. 373–378. Hong Kong, China (2017)
32. Pouyanfar, S., et al.: Dynamic sampling in convolutional neural networks for imbalanced data classification. In: Proceedings of the First IEEE International Conference on Multimedia Information Processing and Retrieval, pp. 112–117. Miami, FL, USA (2018)
33. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
34. Shang, C., Li, M., Feng, S., Jiang, Q., Fan, J.: Feature selection via maximizing global information gain for text classification. Knowl.-Based Syst. **54**, 298–309 (2013)
35. Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., Chanona-Hernández, L.: Syntactic n-grams as machine learning features for natural language processing. Expert Syst. Appl. **41**(3), 853–860 (2014)
36. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**, 4–22 (2009)
37. Sun, X., Liu, Y., Xu, M., Chen, H., Han, J., Wang, K.: Feature selection using dynamic weights for classification. Knowl.-Based Syst. **37**, 541–549 (2013)
38. Tian, H., Zheng, H.C., Chen, S.C.: Sequential deep learning for disaster-related video classification. In: Proceedings of the First IEEE International Conference on Multimedia Information Processing and Retrieval, pp. 106–111. Miami, FL, USA (2018)
39. Tomović, A., Janičić, P., Kešelj, V.: n-gram-based classification and unsupervised hierarchical clustering of genome sequences. Comput. Methods Programs Biomed. **81**(2), 137–153 (2006)
40. Uysal, A.K., Gunal, S.: A novel probabilistic feature selection method for text classification. Knowl.-Based Syst. **36**, 226–235 (2012)
41. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications, vol. 8. Cambridge University Press, Cambridge (1994)
42. Yang, Y.: Feature selection in statistical learning of text categorization. In: Proceedings of 14th International Conference on Machine Learning, pp. 412–420 (1997)
43. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. ICML **97**, 412–420 (1997)
44. Zhang, H., Zhong, G.: Improving short text classification by learning vector representations of both words and hidden topics. Knowl.-Based Syst. **102**, 76–86 (2016)
45. Zhang, M., Chen, Y.: Weisfeiler-lehman neural machine for link prediction. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 575–583. ACM (2017)
46. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. arXiv preprint arXiv:1510.03820 (2015)