CHINESE NEWS TEXT CLASSIFICATION ALGORITHM BASED ON ONLINE KNOWLEDGE EXTENSION AND CONVOLUTIONAL NEURAL NETWORK

CHUN-HUI HE, CHONG ZHANG, SHENG-ZE HU, ZHEN TAN, HUI-MING ZHU, BIN GE

Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology
Changsha 410073, China
E-MAIL: xtuhch@163.com, springsun@outlook.com

Abstract:

Chinese text classification is an important task in data mining, which extracts category features from unstructured contents. Conventional Chinese text classification models only leverage the surface features in the original text, which omits the potential extensional knowledge of each word. To capture the semantic features of each word more comprehensive, this paper proposed a Chinese news text classification algorithm based on an online knowledge extension and convolutional neural network (OKE-CNN), which leverages both knowledge graph to extend latent semantic information and CNN to obtain the category. Compared with other baseline methods, OKE-CNN can utilize the surface and latent features, simultaneously, which can be adapted to complex scenes, e.g., sparse data and unclear topics. In our experiment, OKE-CNN exhibits superior performance and achieves 97.94% and 87.03% on THUCNews and TouTiao datasets, separately, over SOTA competitors.

Keywords:

Text classification; OKE-CNN; Bi-LSTM-CRF; CN-DBpedia; Knowledge graph

1. Introduction

With the rapid growth of text messages, such as news, blogs, and social messages on the Internet. Users urgently need to accurately classify these large-scale text data, and so that they can quickly get the information they expect. Text classification belongs to the scope of supervised learning, which causes new texts can be divided into predefined categories [1-2]. Nowadays, relevant researches with the text classification task are mainly divided into two categories. First, long text classification, such as the classification of news content [3]. Second, short text classification, such as the microblog and title text classification [4]. The main difference between them is the lengths of the contents are different. In general, the long text contains more content, which is more conducive to the completion of text classification tasks. However, short text only contains little content, and usually has the features of text sparsity, which is a great challenge for the text classification task [5]. Most studies have shown that long text classification performance far exceeds the short text.

Early in the study, Teahan [6] using the Markov model and cross-entropy effectively to increase the classification accuracy. Zhang [7] designed a Chinese text classification system based on the vector space model (VSM). For the semi-structured text classification task, Kudo [8] first time using the Boosting algorithm to capture substructures in text, and apply it to text classification. Also, combining feature engineering and using traditional machine learning algorithms as classifiers is a popular text classification method. Zhang [9] uses the statistical method TF-IDF to calculate the word weight and filter out important features to classify of the text. Based on the feature engineering, text classification has also achieved good results by combining N-grams [10], Naïve Bayes [11] and the SVM [12] algorithm. In recent years, with the popularity of deep learning. Using deep neural network algorithms to carry out text classification has also achieved many breakthrough results. In 2014, Kim [13] proposed a text classification method based on a convolutional neural network, and the performance has been greatly improved. Then based on the deep learning theory, a series of Chinese text classification methods are proposed [14]. Liu [15] posed a text classification algorithm for multi-task learning based on the recurrent neural network (RNN). Wu [16] by introducing character-level vector embedding technology raised the CNN-SVM Chinese text classification algorithm. To carry out the task of classification of large-scale Chinese text, Liu [17] proposed Parallel Naive Bayes (PNB) algorithm based on the Spark platform. Besides, with the development of attention mechanisms and vector representation techniques, Qiao [18] uses hybrid vector embedding technology and attention mechanism finished Chinese text classification.

Because of the short-text exists data-sparse problem, so

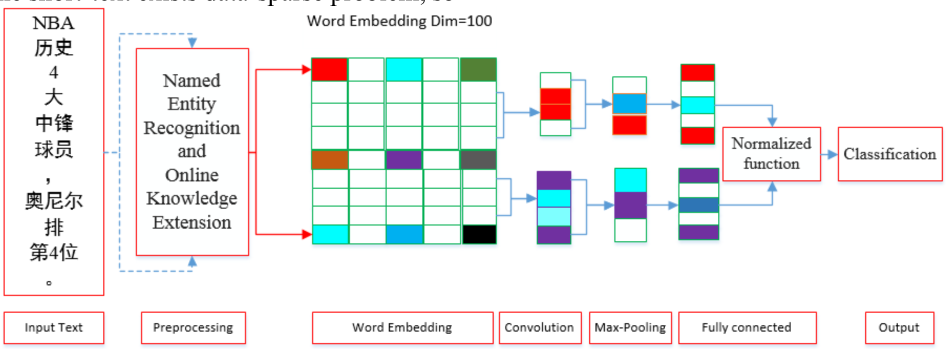

Fig.1 The framework of Chinese news classification method based on online knowledge extension and CNN

the classification is more difficult than long text. Pu [19] dished a short-text classification method based on the ICA and LSA methods. Ge [20] via domain dictionary and LSTM to complete the sentiment classification of short texts. With further research, researchers realized that the sparseness of short-text data has a great impact on the accuracy of text classification. To break through this case, a series of feature extensions [21-25] methods are advanced. Fan [26] and Xiao [27] respectively proposed a Chinese short-text classification method based on Wikipedia and domain knowledge. For the semi-supervised learning method, Yin [28] proposed an improved SVM method for short-text classification. Since Kim [13] applies the convolutional neural network to the text sentence classification task. Many scholars also try to use different deep learning algorithms to carry out the short-text classification. Zeng [29] and Pei [30] respectively put forward the short-text classification methods based on topical memory networks and TW-CNN. With the improvement of attention technology [31-32], there are also have some researchers who try to apply it to the short-text classification task. Among them, Liu [33] uses attention mechanisms, bidirectional gated recurrent unit and convolutional neural network to pose the BiGRU-CNN model.

Through the analysis of related literature, it is found that there are two major challenges in the field of Chinese news text classification. First, how to continue to improve the classification accuracy of Chinese news texts? Second, how to better overcome the problem of data sparseness? To solve the two core challenges, we using entity recognition and online knowledge graph to extend external knowledge, and proposed the Chinese news text classification algorithm based on online knowledge [26][34] extension and convolutional neural network. The overall framework is shown in fig.1.

According to fig.1, the entity recognition in the original corpus is completed by using the named entity recognition technology [35]. Then, combined with an online knowledge graph to develop the disambiguation and automatic alignment of the entities. Next, by searching the online knowledge graph CN-DBpedia [34], the knowledge involved in the relevant entities is extracted and extended to the original corpus. After the above processing, we can extend the original content into a longer content. Finally, the extended corpus is used to train and test the convolutional neural network model. In this paper, our core contributions are as follows:

(1). We using the named entity recognition technology and online knowledge graph to finish the external knowledge extension with text classification task.

(2). The method fully adopted the large-scale online knowledge graph to carry out the disambiguation and alignment of the entity.

(3). We proposed a Chinese news text classification method based on online knowledge extensive and convolutional neural network, which can effectively improve the classification performance.

The rest of the paper is organized as following. Named entity recognition and online knowledge extension are outlined in Section 2. The convolutional neural network classification model will be presented in Section 3. And then, give the experimental results and analysis in Section 4. In Section 5 and 6, we did the discussions, and give the conclusion and future work of the paper.

2. NER and online knowledge extension

Through using entities and online knowledge graphs to carry out knowledge extension is part of the innovations in this paper. We select extensive knowledge only from organization and person name two types of entities. The framework of named entity recognition and online knowledge extension was shown in fig.2.

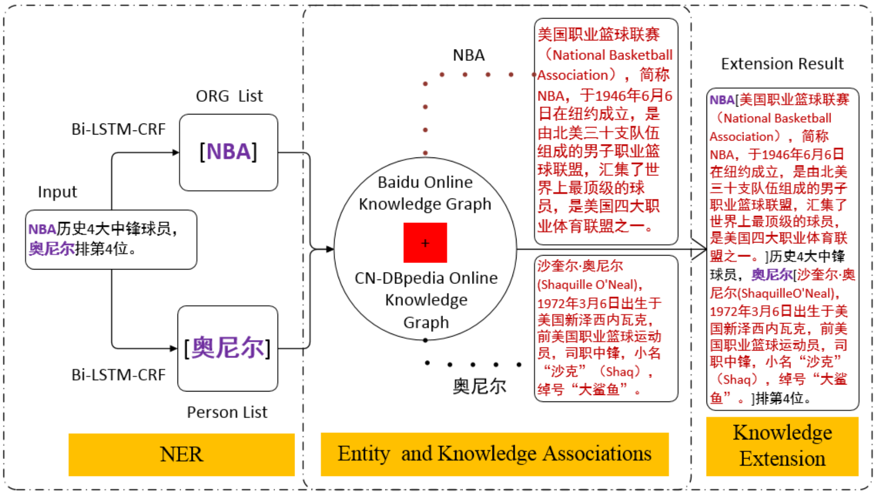

Fig.2 The framework of named entity recognition and online knowledge extension.

2.1. Named entity recognition

In this paper, the Bi-LSTM-CRF model is used to carry out the named entity recognition. The text mainly identifies the organization and person name two types of entities. The specific formula of the LSTM neural unit in the model is as follows:

$$i_t = \sigma(w_{xi}x_t + w_{hi}h_{t-1} + w_{ci}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma\left(w_{fi}x_t + w_{hf}h_{t-1} + w_{cf}c_{t-1} + b_f\right) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(w_{xo}x_t + w_{ho}h_{t-1} + w_{co}c_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \tanh\,(c_t) \quad (5)$$

where， i_t, f_t, o_t, c_t respectively denote input gate, forget gate, output gate, and a cell memory at time t. σ represents the activation function, w_x, w_h, w_c are weight matrix, and b_i， b_f， b_c， b_o means bias vectors. For the convenience of description, the LSTM (i_t, h_{t-1}) is used to indicate the basic LSTM model. For the bidirectional long short-term memory (Bi-LSTM) model, the function is to extract potential features from the statement. The extracting model can obtain the embedding of each word and input it into Bi-LSTM as a sequence L= (i_1, i_2, …, i_s), here $i_s \in R^{d_s}$. It contains two LSTMs in Bi-LSTM, one is extracting forward hidden feature

$\overrightarrow{h_t}$, and another will extract the backward hidden feature $\overleftarrow{h_t}$. The Bi-LSTM can be expressed as follows:

$$\overrightarrow{h_t}, \overrightarrow{c_t} = LSTM(i_t, \overrightarrow{h_{t-1}}, \overrightarrow{c_{t-1}}) \quad (6)$$

$$\overleftarrow{h_t}, \overleftarrow{c_t} = \mathrm{LSTM}(i_t, \overleftarrow{h_{t+1}}, \overleftarrow{c_{t+1}}) \quad (7)$$

Combining the forward and backward hidden layer features will get the output $h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}]$. An activation function is used to extract the features of $\overrightarrow{h_t}$和$\overleftarrow{h_t}$, and then map them into a space of d_l dimensions.

$$\mathrm{l} = \tanh\,(w_l \tanh(w_t h_t + b_t) + b_l) \quad (8)$$

where d_l means the number of entity types that need to be recognized, and w_l, w_t is a feature matrix, and b_t, b_l denote bias vectors. After the features of Bi-LSTM output, the model uses CRF to recognize the specific label y_t at each location of the sequences. The model merges the output l of the Bi-LSTM layer in a sequence $\mathrm{L} = [l_1, l_2, \dots, l_{s-1}, l_s]^T$ input into the CRF model. Here the size of L is $\mathrm{s} \times d_l$， $L_{i,j}$ indicates the probability that the i-th word in the sentence is recognized as the j-th label. For a prediction sequence $\mathrm{y} = (y_1, y_2, \dots, y_{s-1}, y_s)$， and the scoring function of CRF was defined as follows:

$$\mathrm{f(X,y)} = \sum_{i=0}^{s+1} T_{y_i y_{i+1}} + \sum_{i=1}^{s} L_{i,y_i} \quad (9)$$

Where T represents a transfer matrix, $T_{i,j}$ indicates the probability that the i–th label is transferred to the j-th label. y_0 and y_{s+1} denote the labels of the start and end position. Finally, a normalized SoftMax function is used to calculate the probability of the sequence y.

$$\mathrm{p(y|X)} = \frac{e^{f(X,y)}}{\sum_{\tilde{y} \in Y} e^{f(X,\tilde{y})}} \quad (10)$$

Where Y is all sequences that X may generate. In our method, we use the public annotation corpus to train the Bi-LSTM-CRF model. The BIOSE five-level labeling system is used to carry out the data labeling of the organization and person name entity and the non-entity for each character. Finally, the trained model is used to recognize the organization and person name entity in the input text. As shown in fig.2， the input text is O'Neal ranked fourth in the NBA's four famous center players.)". The output entity is labeled as NBA(N>B-ORG) (B>I-ORG) (A>E-ORG) and O'Neal(O>B-Person) ('>I-Person) (N>I-Person) (e>I-Person) (a>I-Person) (l>E-Person). NBA be labeled as an organization entity, and O'Neill labeled as a person name entity.

2.2. Online knowledge extension

According to fig.2, our online knowledge extension module will rely on named entity recognition. After you input the original text that needs to expands. First, the Bi-LSTM-CRF model is used to carry out the named entity recognition, getting the types of entities with the organization and personal name. Then, we use the candidate entities list as the query object input into the Baidu Online Knowledge graph to get the entity results that have the highest match with the input entities. For example, we input the entity is “O'Neal”. In the knowledge graph, existing the highest matching degree entity with the input entity is “Shaquille O'Neal”. In particular, if there is no matching entity in the knowledge graph for the input entity, then the input entity itself is returned as the result of the query. In this way, through Baidu online knowledge graph, the entity disambiguation and automatic alignment tasks can be efficiently performed. Then, use the aligned entity as the query object input CN-DBpedia Online Knowledge graph to make a query and return the knowledge that is most relevant to the input entity. such as input “Shaquille O'Neal”, return the most relevant knowledge is “Shaquille O'Neal born in Newark, New Jersey, the USA on March 6, 1972, former professional basketball player, center, nicknamed "Shaq", nicknamed "Big Shark". Next, extend return knowledge to the behind position of the corresponding entity in the original text until all entities have finished the extension. While all the corpora have completed the above extensions, they can be used as a new corpus to carry out training and testing of the convolutional neural network classification algorithm.

3. CNN text classification model

With the development of deep learning algorithms, Convolutional Neural Network (CNN) has been widely used in the areas of speech recognition, image processing, and natural language processing. We lead into a CNN Model based on the Tensorflow [36]. The CNN model consists of 7 layers, and it contains an input layer, preprocessing, word embedding, convolution, pooling, fully connected and output layer. The framework overview of the CNN model is shown in fig.1. The proposed model takes the original text that needs to be processed as input. The pre-processing layer preprocesses the input original text, mainly including named entity recognition and online knowledge extension. The word embedding layer implements the representation of the word vector with all the words after preprocessing. Thereby forming a vector matrix. Let $x_i \in R^k$ be the 100-dimensional word vectors corresponding to the i-th effective word for the input extended text. Assuming that the input text contains M effective words, then the vector corresponding to the entire input content is:

$$x_{1:M} = x_1 \oplus x_2 \oplus \cdots \oplus x_M \tag{11}$$

Where $\oplus$ is the connection operator. The network architecture of the CNN model is introduced as follows, which includes the convolutional, the max-pooling, and the fully connected layer. The convolutional layer is composed of several units, and the parameters of each convolution unit are obtained by the back-propagation process. The pooling layer is also called the sampling layer, which is used to reduce the dimension of the feature, compress the amount of data and number of parameters, reduce the complexity of the model and improve the fault tolerance and training speed of the model. In our experiments, the max sampling method is used. After the convolution layer and the pooling layer, the obtained feature maps are sequentially expanded in rows and connected into vectors, which are then transferred to the fully connected layer. In the above process, let $x_{i:j}$ be the connection of the word vector $x_i, x_{i+1}, \cdots, x_{i+j}$. The convolution kernel is $w \in R^{s*d}$, where s is the convolution window size, and it’s setting to 5 in our method. Here, d is a word vector dimension. The feature C_i generated by the convolution is:

$$C_i = f(w \cdot x_{i:i+5} + b) \tag{12}$$

Where b is a bias vector, and f is the activated function, the Relu function is used in our experiments. After that, the convolution kernel is applied to every possible window $\{x_{1:5}, x_{2:6}, \cdots x_{M-5+1:M}\}$, and finally a feature map is generated:

$$\mathrm{C} = [C_1, C_2, \cdots, C_{M-5+1}] \tag{13}$$

The feature map C is processed by a maximum pooling operation to obtain the feature $\hat{C} = Max(C)$. Finally, the generated feature map is input into the fully connected layer for calculation, and the last category and the corresponding probability value are given by the SoftMax function before the output layer. The classification is performed in the output layer based on the magnitude of the probability.

4. Experimental results and analysis

4.1. Dataset introduction

To evaluate the performance of the OKE-CNN model, we carried out experiments based on two public Chinese news datasets. The first original dataset is a sub-dataset of

THUCNews, which has 10 categories, and it contains 65,000 Chinese news [3] text. The second original dataset have 15 categories, and it consists of 382,688 Chinese news title of TouTiao News website. For the THUCNews sub-dataset, and it is a long-text classification dataset. We use the title and content of each news document as the input corpus. But, for the TouTiao dataset, we just use the title from each news document as the input corpus. So, the TouTiao dataset is a short-text classification dataset. For the two datasets, we only use organization and person name two types of entities in their title to carry out the extension of knowledge. All datasets are summarized in table 1, which shows the number of documents (Doc Num) and average character length (AvgCL) of the document with the original dataset and extended dataset.

Table 1 a summary of all the experimental datasets

Origina/ Extension	**OriginalDataset**		**Extended Dataset**	
	DocNum	**AvgCL**	**DocNum**	**AvgCL**
THUCNews Train	50000	917	50000	995
THUCNews Val	5000	886	5000	962
THUCNews Test	10000	973	10000	1058
TouTiao Train	298688	41	298688	65
TouTiao Val	42000	40	42000	67
TouTiao Test	42000	42	42000	68

4.2. Evaluation metrics

In our experiment, we use Accuracy Rate, Loss Rate, Precision, Recall, and F-measure (F_1) to measure the performance of the OKE-CNN classification model. The precision and recall are widely used as quality metrics in the area of information retrieval. Precision and recall indicators sometimes may face extreme conditions. Therefore, it needs to process them comprehensively. The most common method is using the F-Measure index, which is the comprehensive indicator of precision and recall. When the β is set to 1, it becomes the common F_1 measure, which is a comprehensive indicator. Accuracy and F_1 value are calculated as follows:

$$\text{Accuracy} = \frac{\text{Correctly categorized sample size}}{\text{Total number of samples}} \tag{14}$$

$$F_1 = \frac{2*Precision*Recall}{Precision+Recall} \tag{15}$$

The higher accuracy and F_1 value show that better performance of the algorithm. Finally, by calculating the average value of each indicator, the overall performance of the classification algorithm can be obtained.

4.3. CNN model core hyper-parameters

The hyper-parameters of the OKE-CNN model in our experiment is shown in table 2.

Table 2 hyper-parameters setting of the OKE-CNN model

Parameter Name	**Value**	**Parameter Name**	**Value**
Dropout_rate	0.5	Kernel_size	5
Num_filters	256	Vocab_size	100000
Embedding_dim	100	Hidden_dim	256
Batch_size	64	Num_epochs	15

Categories size is 10 in THUCNews sub-dataset, and it's 15 to the TouTiao dataset.

4.4. Overall performance

In table 3, we show the results of the comparison of OKE-CNN with the baseline methods, in terms of average Accuracy, Loss, and average F_1 on the THUCNews Val sub-dataset.

Table 3 experimental results on THUCNews Val dataset

Model	**The Comparison experimental results [3]**				
	Accuracy	**Loss**	**Precision**	**Recall**	**F_1**
TF-IDF	90.27	0.31	---	---	0.86
SVM	93.49	0.25	---	---	0.92
LSTM	94.26	0.21	---	---	0.94
CNN	95.61	0.15	---	---	0.96
Bi-LSTM-CNN	96.45	0.11	---	---	0.99
OKE-CNN	**97.30**	**0.08**	**---**	**---**	**0.998**

Through the results of table 3. On the THUCNews Val sub-dataset, we can know that the average accuracy and F_1 of the OKE-CNN model have reached 97.3% and 99.8%, with 0.08 Loss value. table 4 indicates the experiment result of the OKE-CNN model on the THUCNews Test sub-dataset.

Table 4 experimental results on THUCNews Test dataset

Model	**Accuracy**	**Loss**	**Avg Precision**	**Avg Recall**	**Avg F_1**
OKE-CNN	97.94	0.07	0.98	0.98	0.98

As shown in the results in table 4, we can see that the performance of the OKE-CNN model on the independent THUCNews Test sub-dataset has exceeded the-state-of-art method. Besides, we also used the OKE-CNN model and comparison methods to finish the classification experiment on the public TouTiao News classification dataset. The experimental results on the independent TouTiao Test dataset are shown in table 5.

Table 5 experimental results of the models on TouTiao dataset

Model	**The Comparison experiments result**				
	Accuracy	**Loss**	**Avg Precision**	**Avg Recall**	**Avg F_1**
Baseline	83.81	0.57	0.85	0.84	0.84

LSTM	84.33	0.52	0.85	0.84	0.84
CNN	85.43	0.51	0.86	0.85	0.86
OKE-CNN	**87.03**	**0.44**	**0.88**	**0.87**	**0.87**

According to the experimental results in table 5, the average Accuracy, Precision, Recall and F_1 of the OKE-CNN model have arrived above 87% and the minimum loss value was 0.44, which are also better than the classic CNN and LSTM models.

5. Discussion

In OKE-CNN model, character-level and word-level two different vector embedding methods are used to experiment. The two methods' loss value has shown in fig.3.

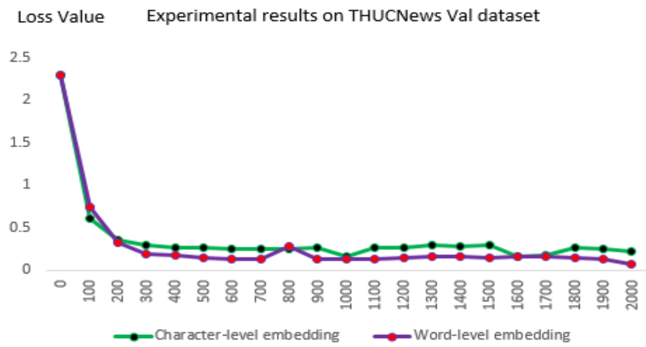

Fig.3 The OKE-CNN Loss value with differents vector embedding on THUCNews Val dataset

According to the results of fig.3, it is found that the loss value of using the word-level vector embedding is slightly lower than the character-level vector.

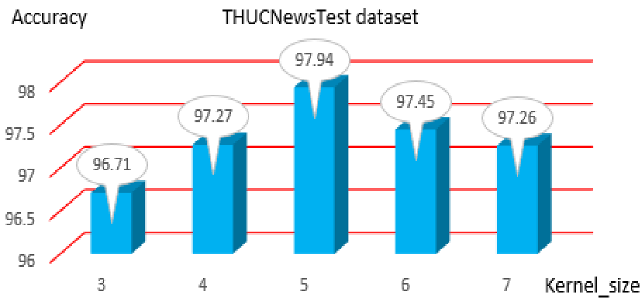

Fig.4 The OKE-CNN model classification accuracy with differents kernel_size on THUCNews Test dataset.

For the adjustment of hyper-parameters, we found that the number of convolution kernels setting in the OKE-CNN model is important, which will have a greater impact on the classification accuracy. Fig.4 shows the experimental results for OKE-CNN on the different number of convolution kernel with THUCNews Test and TouTiao Test datasets.

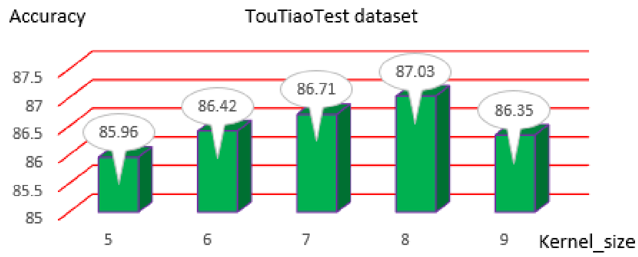

Fig.5 The OKE-CNN model classification accuracy with differents kernel_size on the TouTiao Test dataset.

According to fig.4 and fig.5, while the number of convolution kernels respectively set to 5 and 8, and the OKE-CNN model can achieve the best classification performance on the two different datasets.

Our experimental results show that using an online knowledge graph to carry out entity disambiguation will have some certain impacts on improving the classification accuracy compared with no entity disambiguation. Adding the entity disambiguation strategy, and the introduction of noise data can be reduced in the extended knowledge. Especially, for Chinese news text, we only select organization and person name two types of entities to extend knowledge. For other fields, it may be better to choose other types of entity to do the extension of knowledge, such as domain terminology.

6. Conclusion and future work

In this paper, we proposed a Chinese news text classification method based on online knowledge extension (OKE) and convolution neural networks (CNN), which including three core steps. First of all, it uses named entity recognition technology to recognize entities. Second, merges with entity and online knowledge graph to extend the original text. In the last step, it utilizes the CNN model as a text classifier. We have described a series of comparative experiments with different models based on two public Chinese datasets. And we have demonstrated that our method performs best among all the contrast methods. In particular, our method by extending knowledge, it improves the problem with data sparsity of text.

Applying a knowledge graph to a relevant text mining task is a very challenging study. We initially tried to apply online knowledge to complete a text classification task. Currently, our method is only verified on the Chinese public news text classification datasets. In the future, we will further optimize the method and apply it to other text mining tasks, such as the classification of Twitter text or text clustering.

Acknowledgments

This research received the support of the National Natural Science Foundation of China (NSFC) via grant No.61872446 and No.61902417. Thanks to the Open Online API offered by Baidu AI Open Platform and the CN-DBpedia Online Knowledge Graph of Fudan University. Also special thanks to Professor Weidong Xiao and Associate Professor Yanli Hu for their valuable suggestions during the article writing.

References

[1] Liu, Bing. Web data mining: exploring hyperlinks, contents, and usage data. Springer Science & Business Media, 2007.
[2] Manning, Christopher, Prabhakar Raghavan, and Hinrich Schütze. "Introduction to information retrieval." Natural Language Engineering16.1 (2010): 100-103.
[3] Li, Chenbin, Guohua Zhan, and Zhihua Li. "News Text Classification Based on Improved Bi-LSTM-CNN." 2018 9th International Conference on Information Technology in Medicine and Education (ITME). IEEE, 2018.
[4] Wang, Yongguan, et al. "A News Headlines Classification Method Based on the Fusion of Related Words." National CCF Conference on Natural Language Processing and Chinese Computing. Springer, Cham, 2017.
[5] Mirończuk, et al. "A recent overview of the state-of-the-art elements of text classification. "Expert Systems with Applications 106 (2018): 36-54.
[6] Teahan, et al. "Text classification and segmentation using minimum cross-entropy." Content-Based Multimedia Information Access-Volume 2, 2000.
[7] Zhang, Dongli, Dongsheng Wang, and Weimin Zheng. "Chinese text classification system based on VSM." Qinghua Daxue Xuebao/Journal of Tsinghua University (China) 43.9 (2003): 1288-1291.
[8] Kudo, Taku, and Yuji Matsumoto. "A boosting algorithm for classification of semi-structured text." Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. 2004.
[9] Yun-tao, Zhang, Gong Ling. "An improved TF-IDF approach for text classification." Journal of Zhejiang University-Science A 6.1 (2005): 49-55.
[10] Wei, Zhihua, et al. "N-grams based feature selection and text representation for Chinese Text Classification." International Journal of Computational Intelligence Systems 2.4 (2009): 365-374.
[11] Chen, Jingnian, et al. "Feature selection for text classification with Naïve Bayes." Expert Systems with Applications 36.3 (2009): 5432-5435.
[12] GONG, Zhi-le, De-xian ZHANG, and Ming-ming HU. "An Improved SVM Algorithm for Chinese Text Classification [J]." Computer Simulation 7 (2009).
[13] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).
[14] Li, Yue, Xutao Wang, and Pengjian Xu. "Chinese text classification model based on deep learning." Future Internet10.11 (2018): 113.
[15] Liu, Pengfei, et al. "Recurrent neural network for text classification with multi-task learning." arXiv preprint arXiv:1605.05101 (2016).
[16] Wu, Huaiguang, et al. "Chinese Text Classification Based on Character-Level CNN and SVM." International Symposium on Intelligence Computation and Applications. Springer, Singapore, 2018.
[17] Liu, Peng, et al. "Parallel naive Bayes algorithm for large-scale Chinese text classification based on spark." Journal of Central South University 26.1 (2019): 1-12.
[18] Qiao, Xue, et al. "Word-character attention model for Chinese text classification." International Journal of Machine Learning and Cybernetics (2019): 1-17.
[19] Pu, Qiang, and Guo-Wei Yang. "Short-text classification based on ICA and LSA." International Symposium on Neural Networks. Springer, Berlin, Heidelberg, 2006.
[20] Bin G, Chunhui H, Chong Z, et al. Classification Algorithm of Chinese Sentiment Orientation Based on Dictionary and LSTM. Proceedings of the 2nd International Conference on Big Data Research. ACM, (2018): 119-126.
[21] Fan, Xinghua, and Hongge Hu. "A new model for chinese short-text classification considering feature extension." 2010 International Conference on Artificial Intelligence and Computational Intelligence. Vol. 2. IEEE, 2010.
[22] Fan, Xinghua, and Hongge Hu. "Construction of high-quality feature extension mode library for chinese short-text classification." 2010 WASE International Conference on Information Engineering. Vol. 2. Ieee, 2010.
[23] Jun, Hu Y., Jiang J. Xin, and Chang H. You. "Chinese short-text classification based on topic model with high-frequency feature expansion." Journal of Multimedia 8.4 (2013): 425.
[24] Zhang, Xinwei, and Bin Wu. "Short text classification based on feature extension using the n-gram model." 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD). IEEE, 2015.

[25] Wang, Peng, et al. "Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification."Neurocomputing 174 (2016): 806-814.
[26] Yunjie, Fan, and Liu Huailiang. "Research on Chinese short text classification based on Wikipedia." Data Analysis and Knowledge Discovery 28.3 (2012): 47-52.
[27] Feng, Xiao, et al. "Chinese short text classification based on domain knowledge." Proceedings of the Sixth International Joint Conference on Natural Language Processing. 2013.
[28] Yin, Chunyong, et al. "A new SVM method for short text classification based on semi-supervised learning." 2015 4th International Conference on Advanced Information Technology and Sensor Application (AITS). IEEE, 2015.
[29] Zeng, Jichuan, et al. "Topic memory networks for short text classification."arXiv:1809.03664 (2018).
[30] Pei, Kefeng, et al. "Short Text Classification Research Based on TW-CNN." PACIS. 2018.
[31] Zhou, Yujun, et al. "Hybrid attention networks for Chinese short text classification." Computación y Sistemas 21.4 (2017): 759-769.
[32] Tao, Hanqing, et al. "A Radical-aware Attention-based Model for Chinese Text Classification." Thirty-Third AAAI Conference on Artificial Intelligence. 2019.
[33] Liu, Jin, et al. "Attention-based BiGRU-CNN for Chinese question classification." Journal of Ambient Intelligence and Humanized Computing (2019): 1-12.
[34] Xu, Bo, et al. "Cn-dbpedia: A never-ending chinese knowledge extraction system." International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer, Cham, 2017.
[35] Huang, Zhiheng, Wei Xu, and Kai Yu. "Bidirectional LSTM-CRF models for sequence tagging." arXiv preprint arXiv:1508.01991 (2015).
[36] Abadi, Martín, et al. "Tensorflow: A system for large-scale machine learning." 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). 2016.