Word embedding and text classification based on deep learning methods

Saihan Li[1,*], and *Bing* Gong[1]

Xi'an Eurasia University, Rd Dongyi 8, Xi'an, China

Abstract. Traditional manual text classification method has been unable to cope with the current huge amount of data volume. The improvement of deep learning technology also accelerates the technology of text classification. Based on this background, we presented different word embedding methods such as word2vec, doc2vec, tfidf and embedding layer. After word embedding, we demonstrated 8 deep learning models to classify the news text automatically and compare the accuracy of all the models, the model '2 layer GRU model with pretrained word2vec embeddings' model got the highest accuracy. Automatic text classification can help people summary the text accurately and quickly from the mass of text information. No matter in the academic or in the industry area, it is a topic worth discussing.

1 Introduction

In recent years, with the rapid development of Internet technology and information technology, especially with the arrival of the era of big data, a huge amount of data is flooding every field of our life. This increasing amount of text information has caused some troubles for people to find what they need. In the past, people chose to classify text information manually, which is time consuming, laborious, and high cost. Nowadays, it is obvious that manual text classification alone can't meet the needs. Based on the background, automatic text classification has emerged.

Text classification is the process of assigning labels to text according to its content, it is one of the fundamental tasks in natural language processing (NLP). NLP methods change the human language to numeral vectors for machine to calculate, with these word embeddings, researchers can do different tasks such as sentiment analysis, machine translation and natural language inference.

This paper introduces the process of text classification, and divides the process into 3 parts, which are text preprocessing, word embedding and classification models. In each part, the methods and models used have been described in detail.

* Corresponding author: lisaihan@eurasia.edu

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

2 Embedding methods

2.1 data preprocessing

The dataset is provided by Sougo Lab which offers open-source dataset for AI research. The dataset includes news from different channels of Sohu news, the size is around 50MB. There are punctuation, numbers, English letters in the text, and they contribute little to the text classification. Hence the regular regression is used to remove them. At the same time, stop words will also be removed. Word segmentation is the separation of the morphemes and the same with tokenization for languages without 'space' character. Chinese is very different from English and other alphabet languages. In English, words are separated by space, but there is no space between words in Chinese. In Chinese, one character can be a word, two or three characters can also make up a word, even 4 characters is also a word. Thus, there are several ways to split a sentence, and it's a big challenge to split the Chinese sentence. Jieba is an open-source Python Chinese word segmentation library and it is used to do segmentation in the experiment.

2.2 word embedding

Words are map into vectors using word embedding models, Word embedding is a collection of statistical language modelling and techniques in NLP area. It maps words and phrase to vectors of real numbers, they capture both semantic and syntactic information of words[1]. Word embedding can be used to calculate word similarity which can be used in many tasks such as information retrieval. In this paper, 4 ways are used for word embeddings which are word2vec, doc2vec, tfidf and embedding layer.

Word2vec

Word2vec is a NLP technology which takes a text corpus as input and words are represented as vectors. The proximity in vector space indicates semantic or functional similarity. The resulting word vector file can be used as features in many natural language processing and machine learning applications[2].

Doc2vec

Doc2vec model changes text document to numeric representations. Each sample is one vector, the dimension of the vector can be determined by user. Both word2vec and Doc2vec are unsupervised learning methods and Doc2vec was developed on the base of Word2vec. Doc2vec inherits the advantages of word2vec such as take semantic and word order into consideration[3].

The corpus is the collection of all the samples. Doc2vec method uses the corpus to train the model and use this model to map every sample to a fixed dimension vector. We set the dimension of text to be 300, then no matter how long the text is, the output is a 300 dimension numeric vector. The same corpus is used for word2vec method, after generating the word2vec model, every word will be mapped to a 100-dimension vector.

Tfidf

Tfidf(term frequency inverse document frequency) is a commonly used weighted technology for information retrieval and data mining. In scikit learn library, TfidfVectorizer is offered to map text to a numeric matrix. Each element is the tfidf value of the word.

Embedding layer

In keras, the first layer of a text classification task is embedding layer, there are 2 ways in this paper to do word embedding. The first one is using word2vec to map words to vectors, each word is 100 dimension long. The second method is using function 'Tokenizer' offered by keras, This function allows to vectorize a text corpus, by turning each text into a sequence of integers, each integer is the index of a token in a dictionary. Hence each sample text will

be transferred to a bag of numbers. In the embedding layer of a deep learning model, the words will be trained and transferred to the next layer, the dimension of each word is set to be 100 here, the same as the word2vec method. The dimension of each sample after these 4 methods is very different as table 1 shows

Table 1. The dimension of each sample using different embedding methods.

Embedding model	Dimension
Doc2vec	300
Word2vec	300*100
Tfidf	Corpus_length
Embedding layer	300*100

3 Deep learning classification models

The research of deep learning begins with artificial neural network, which aims to simulate the operation mechanism of human brain.[4]. The neural network opened up the research of deep learning theory in academic and industry. This series of developments have made breakthroughs in the fields of image speech recognition, automatic translation and NLP tasks[5].

The word embedding obtained from last step will be trained by deep learning methods. Different deep learning models such as CNN (Convolutional neural networks) and RNN (Recurrent neural networks) are trained to compare with machine learning models.

All the deep learning models take categorical cross entropy as loss function, adam as optimizer, batch size is 1000 and each model will run 20 epochs. For each algorithm, the pretrained word2vec embeddings and without pretrained embeddings are the input layer of these models separately. Embeddings without pretrained will be trained at the first layer, on the contrary, word2vec embeddings will be transferred to the next layer directly. There are 8 models are trained and tested in this paper.

Multilayer Perceptron(MLP)

There are 3 layer in this MLP model as Figure 4.5 shows, the input layer is the word vectors, the hidden layer includes 1000 nodes and uses relu as activation function and has 50% dropout, the output layer is a fully connected and uses softmax as the activation function.

Convolutional Neural Network(CNN)

The structure of this CNN model is shown in figure 1, the first layer is a convolutional layer, there are 256 filters and each filter size is 5*5 and then followed by a max pool layer with size 3*3. The third layer is another con-volutional layer with 128 filters with 3*3 size, and followed by a max pool layer with size 3*3. The fifth layer is also a convolutional layer, there are 64 filters in this layer and each kernel is 3*3 size, then after these convolutional and pooling layers, there is a flatten layer to flatten these multi dimensional vectors, after that, there is a hidden layer with 256 units, and its fully connected and the activation is relu, and the output layer is the number of labels with softmax activation.

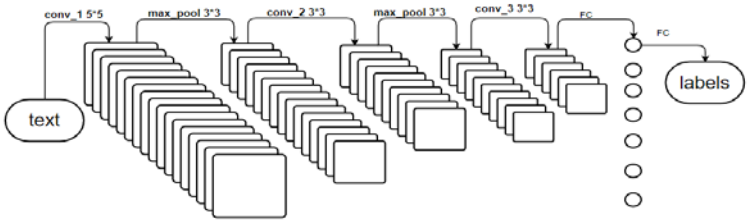

Fig. 1. Structue of CNN model.

Long Short-Term Memory(LSTM)

Recurrent Neural Network(RNN) is a type of artificial neural network de-signed to recognize patterns in sequences of data, such as text, genomes, the spoken word, or numerical times series data emanating from news, stock markets and government agencies [6]. Long Short Term Memory(LSTM) is a variant of RNN, it was first intro-duced by Hochreiter and Juergen Schmidhuber [7].In this LSTM model, the hidden layer units are 300, activation is tahn, and recurrent activation is sigmoid, dropout is 20% and the units to drop for the linear transformation of the recurrent state is 10%.

Gated recurrent units (GRU)

Gated Recurrent Network(GRU) is also a variant of RNN and but simpler than LSTM, it also performs good in NLP area. GRU has 2 gates which are an reset gate and update gate compared with 3 of LSTM. In this GRU model, the hidden layer units are 256, activation is tahn, and recurrent activation is sigmoid, dropout is 20% and the units to drop for the linear transformation of the recurrent state is 10%.

2 layer GRU

In this GRU model, there are 2 hidden layer with the previous GRU model structure, the output of the first layer is the input of second hidden layer.

TextCNN

This model is first introduced by Yoo Kim in 2014 [8], he introduced a way to combine 3 filters with different size, which is 3*3,4*4 and 5*5, so they can handle different length short text. And then concatenate the results after these 3 filters. The next layer is a fully connected layer and the output layer.

CNNGRU and CNNGRU_Merge

CNNGRU model makes a combination of CNN and GRU, the output of CNN is the input of GRU. CNNGRU_Merge model also makes a combination of CNN and GRU like the previous model, the difference is the combination method. The previous model is a sequential model, but CNNGRU_Merge model is the concatenate of the result of CNN and GRU.

Results

The accuracy of the models above is in table 2. Accuracy with pretrained embedding means the embedding layer take word embeddings using word2vec method.

Table 2. Classification accuracy of all the deep learning models.

Deep learning models	Accuracy(%)	Accuracy with pretrained model(%)
MLP	88.4	87.7
CNN	86.0	88.2
LSTM	86.7	87.2
GRU	89.0	91.8
2 layer GRU	86.1	93.0
TextCNN	91.2	87.4
CNNGRU	86.0	92.7
CNNGRU_Merge	92.6	90.6

From table 2 we can see, 2 layer GRU with pretrained word embeddings gets the highest accuracy which is 93%. The 'CNNGRU with pretrained word em-beddings' and 'CNNGRU Merge' model also have very high accuracy which are 92.7% and 92.6% separately. The accuracy of 'CNN' and 'CNNGRU' are lowest compared to other models, which is only 86%.

MLP model has the simplest structure, but from the results we can see, the simplest model is not the worst. The structure of a model determines the model complexity, which means, the running time and space will also be affected.

2 layer GRU with pretrained word2vec embedding has the highest accuracy within all the models using the normal sized dataset. It is the same for half sized dataset and double sized dataset. It performs very well for the Sohu news dataset.

4 Conclusion

In this paper, the process of text classification is introduced, there are 4 ways to do word embedding and in the classification models, 8 deep learning models are taken to do classification. The '2 layer GRU model with pretrained word2vec embeddings' model gets the highest accuracy.

The limitation of the experiment is that it is hard to execute the experiment in all the Chinese news dataset, thus it is a hint for Chinese news classification. Text classification not only can be used in news classification as described in this paper, but also in other areas such as spam detection and sentiment analysis. It will surely help people save time and easy to find what they need. The development of natural language processing and big data also have a good impact on text classification, which will make the classification faster and more accurate.

In this paper, the research was sponsored by the China Association of higher education (Project No.2020XXHYB13), Social science fund of Shaanxi Province (Project No.2019Q019), Scientific research project of higher education of Shaanxi Higher Education Association (Project No.XGH19056) and Shaanxi higher Education teaching Reform Research Project (Project No.19BY130).

References

1. Yoshua Bengio, Holger Schwenk, Jean-S´ebastien Sen´ecal, Fr´ederic Morin, and Jean-Luc Gauvain. Neural Probabilistic Language Models, pages 137–186. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
2. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. ICLR(International Conference on Learning Representations), 2013.
3. Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14, pages II–1188–II–1196. JMLR.org, 2014.
4. Wissal Farsal, Samir Anter, and Mohammed Ramdani. Deep learning: An overview. In Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications, SITA'18, pages 38:1–38:6, New York, NY, USA, 2018. ACM.
5. T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing [review article]. IEEE Computational Intelligence Magazine, 13(3):55–75, Aug 2018.
6. Kyunghyun Cho, Bart van Merri¨enboer, Caglar Gulcehre, Dzmitry Bah-danau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical ma-chine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.
7. Sepp Hochreiter and J¨urgen Schmidhuber. Long short-term memory. Neural Comput., 9(8):1735–1780, November 1997.
8. Yoon Kim. Convolutional neural networks for sentence classification. CoRR, abs/1408.5882, 2014.