

GRU based Convolutional Neural Network with Initialized Filters for Text Classification

Linhong Weng¹[0000-0001-7182-331X], Qing Li²[0000-0001-7002-7127] and Xuehai Ding³[0000-0001-6258-007X]

School of Computer Engineering and Science, Shanghai University, Shanghai, China
{wenglinhong, qli, dinghai}@shu.edu.cn

Abstract. Text classification is a classical task of natural language processing, which can quickly find corresponding categories from massive amount of data. The method based on Neural Network is currently the most concerned by researchers, such as convolutional neural networks(CNNs). However, convolutional operation in CNNs which extracts features is regional. It fails to consider about the contextual information of text. Besides, inappropriate parameter setting will greatly limit the performance of CNNs. It is noteworthy that contextual information plays an important role in the task of text classification. In order to make full use of context information to improve classification performance, we propose a new neural network-based method-GRU based convolutional neural network with initialized filters (IF-CGNN), which fully combines CNN and multi-layers GRU. In our method, convolutional layer extracts features from the text and then the k-max pooling chooses the k features. Multi-GRU layers access the preceding and succeeding contextual features by combining hidden layers. It can help to understand the sentence semantics. Besides, for reducing the time of parameter adjustment in convolutional layer, training data is used to initialize filters. Finally, experimental results on three open datasets and one medical dataset validate the effectiveness of the proposed method.

Keywords: Text classification, Neural network, Contextual information, Weight initialization

1 Introduction

With the rapid development of Internet, massive amount of electronic texts is generated. How to extract useful information from massive data quickly and effectively is a big challenge. As a classic task of natural language processing, text classification can quickly find the corresponding categories from massive data and realize automatic classification. Therefore, text classification plays an essential role in text data retrieval and mining [1].

The traditional text classification mainly uses machine learning methods. Text is first represented as a vector by use of the feature engineering, and then through the machine learning algorithm to realize the classification process [2][3]. In the past, the most classical feature engineering is bag-of-words [4]. Although this representation is simple and convenient, it is based on the assumption that each word in the document is independent. Therefore, text is regarded as a set of several words. It focuses on the frequency of words in the text and fails to consider the word order, grammar, syntax in the sentence [21]. Because of this, more and more researchers pay attention to deep learning methods.

Deep learning is a huge scientific and technological revolution [5]. It has made outstanding achievements in image processing, speech recognition and other fields, which makes it become a noted method in the field of computer. With the perfect performance of deep learning in the field of image, researchers begin to apply deep learning to the field of natural language processing. For example, Kim [7] in 2014 converts the text to the 4-dimensional vector, which is taken as input to the convolutional neural network (CNN) for training like image process. Compared with the traditional machine learning method, the method has some improvements on classification result. However, it fails to consider the contextual information since the convolutional operation is regional [8][9]. Different from CNNs, recurrent neural networks (RNNs) is the other type of method in deep learning [10], whose structure is a form of sequence. Since the text is composed of a sequence of words [11], RNNs are naturally suitable for natural language processing. However, when the sequence is too

long, RNNs are prone to gradient disappearance or gradient explosion. What's more, it is well known that inappropriate parameter setting will greatly limit the performance of NNs. The neural network tuning process takes a lot of time. In order to improve the accuracy of classification and reduce the time of model tuning, we propose a new model in this paper – GRU based convolutional neural network with initialized filters (IF-CGNN). In IF-CGNN, convolutional layer extracts features from the text. Then the k-maxpooling chooses the k features. Multi-GRU layers accesses the preceding and succeeding contextual features by combining hidden layers. It can help to understand the sentence semantics. Finally, the features are fed into the softmax classifier. With the combination of CNN and multi-GRU layers, the model fully takes contextual information into consideration. Besides, for reducing the time of model tuning, we use training data to initialize filters which is used in convolutional layer.

The main contributions of the paper are as follows:

- 1 In order to make full use of the contextual information in the sentence, we propose a new model which combines CNN with multi-layers GRU.
- 2 As compared to existing NN-based text classification method, our model reduces the time of parameter tuning in convolutional layer.
- 3 The experimental results show that our model outperforms the state-of-the-art methods on three commonly used datasets and a medical dataset.

The remainder of this paper is organized as follows. Section 2 introduces GRU and gives a short literature review on text classification. Section 3 describes our work of initializing filters and the proposed model IF-CGNN in details. In Section 4, we evaluate and analyze the proposed model through experiments on open datasets, which provide experimental settings, performance indicators and results. At last in Section 5, we make a conclusion for the whole text and future work.

2 Gated recurrent unit and Related Work

2.1 Gated recurrent unit

RNNs are a kind of feedforward neural networks which have a recurrent hidden state and the hidden state is activated by the previous states at a certain time. Therefore, RNNs can model the contextual information dynamically and can handle the variable length sequences. GRU is a kind of RNNs architecture and has become the mainstream structure of RNNs at present[31]. GRU addresses the problem of vanishing gradient by using a gating mechanism which tracks the state of sequences without using separate memory cells. There are two types of gates in GRU: the reset gate r_t and the update gate z_t . They control how information is updated to the state together. A GRU unit consists of the four components and it is as illustrated in Fig. 1.

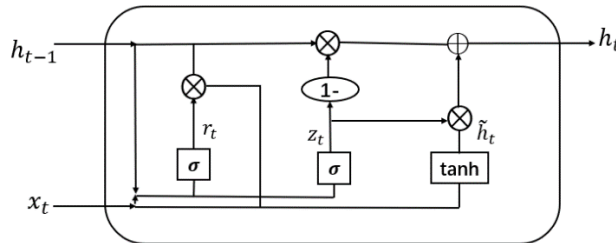


Fig. 1. Gated recurrent unit

The mathematical form of GRU shown in Fig. 1 is given below. The hidden state h_t given input x_t is computed as follows:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (1)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (2)$$

$$\tilde{h}_t = \tanh(W_h x_t + r_t \otimes (U_h h_{t-1}) + b_h) \quad (3)$$

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \tilde{h}_t \quad (4)$$

where, z_t is the update gate. r_t is the reset gate. h_{t-1} is the previous state. $\tilde{h}_t$ is the candidate state at time t . x_t is the sequence vector at time t . $\sigma(\cdot)$ and $\tanh(\cdot)$ is sigmoid and hyperbolic tangent function respectively. b_z , b_h , b_r is a bias term respectively. The operator $\otimes$ denotes element-wise multiplication.

2.2 Related work

In recent years, neural networks have been used to text classification and the breadth of applications for CNN has expanded rapidly. In order to further improve the performance of CNN to handle the variable-length sequential information, many researchers have proposed many methods to improve CNN. In order to capture the relation between words, Kalchbrenner[12] proposes a novel CNN model with a dynamic k-max pooling. Johnson [8] add word order to CNN, demonstrating the state-of-art effectiveness. Johnson[13] find that shallow-level CNNs are more accurate and faster than the most advanced deep networks. Zhang[30] offers an empirical exploration on the use of character-level convolutional networks (ConvNets) for text classification. It takes a sequence of encoded characters as input and doesn't consider of language of text. However, convolutional operation used in these CNNs is regional, whose filters is random initialization. After repeated training of the convolutional neural network, the parameters of the filter are determined. It is known that inappropriate parameter setting will greatly limit the performance of CNN. The results caused by the weight of random initialization will greatly reduce accuracy.

RNNs can extremely calculate and maintain context information by utilizing sequence structure and maintaining such information through the middle layer, which makes it have special advantages in processing sequence data. Based on this, [Liu][14] used the multitask learning framework to jointly learn across multiple related tasks. He proposed three different mechanisms of sharing information to model text with task-specific and shared layers based on recurrent neural network. What's more, Yang [15] proposed hierarchical attention model which incorporates attention mechanism into hierarchical GRU model so that the model can better capture the important information of a document. Wang [16] incorporated position-invariance into RNN by limiting the distance of information flow in RNN, named as disconnected recurrent neural network. However, when the sequence of input is too long, the model tends to gradient explosion. In view of this problem, some researchers try to combine RNN with CNN. Lai [17] applied a recurrent structure to capture contextual information and employed a max-pooling layer that automatically judges which words play key roles in text classification to capture the key components in text. Zhou [18] uses convolution operation to extract a sequence of higher-level phrase representations, and then feed the sequence to LSTM for text classification.

It can be seen that much research has been done in the basic structure of CNN and RNN to enhance its performance and CNN has also achieved outstanding results in text classification. These methods are the basis of IF-CGNN and therefore they are elaborated.

3 Our Work

Convolutional neural network is good at extracting features quickly. However, convolutional operation used in CNN is regional and convolutional neural network is greatly affected by parameter adjustment. These problems affect the text classification accuracy of CNN. In order to improve the performance of text classification, this paper attempts to design the novel architecture which helps to address the drawbacks mentioned above by initializing filters and combining CNN with GRU. The proposed architecture is named GRU based convolutional neural network with initialized filters (IF-CGNN).

3.1 Initializing Filters

Learning from that the embedding layers could be initialized by pre-trained word vectors, we consider that filters in CNNs could be initialized as well. The process of convolutional operation is the inner product result of filters and data from different windows. In natural language processing, convolutional operation can be understood as a process in which a sliding window continuously captures semantic features in finite length sentences. The result of each capture can be equal

to the feature of extracting n-grams at a time. Inspired of this, we replace the process with the n-grams feature to initialize the filters. The whole implementation is shown in Algorithm 1:

Algorithm 1 Initializing filters

Input: a sentence set S

Output: initialized filters $\tilde{w}$

- 1: build a set N_y, L_y, W_y for each class y , which are initialized to be empty
 - 2: for each sentence s in S , obtain n-grams from s and then, add n-grams to N_y , where y is the label of s
 - 3: for each class y , calculate r in N_y by (5)
 - 4: based on r select the first 20% n-grams in N_y and then, add it to L_y
 - 5: conduct K-means function on L_y for each class y , add centroids to W_y
 - 6: obtain $\tilde{w} = \bigcup_y W_y$
-

First, we extract the n-grams from training data and judge the importance of each n-gram by weighting technique in each class y . the weight r calculation formula is as following:

$$r = \log \frac{(p^t + b) / \|p\|_1}{(q^t + b) / \|q\|_1} \quad (5)$$

where, where p^t is the number of texts that contain n-gram t in one class, $\|p\|_1$ is the number of texts in one class, q^t is the number of texts that contain n-gram t in other classes, $\|q\|_1$ is the number of texts in other classes, b is a smoothing parameter.

Then, n-grams of the top 20% is selected according to the weight r in each class y . After that, k-means clustering method is used to cluster the selected n-grams.

The cluster centroid vectors obtained after clustering feeds to the center of filters. The rest position of filters is random initialization. Therefore, the initialized filters are obtained. Once an important feature is found during the training process, the weight of the feature is increased, thereby simplifying the method of random initialization.

After this process, semantic features are encoded into the filters.

3.2 IF-CGNN

The structure of our model IF-CGNN is shown in the Fig.2. The model consists of word embedding, convolutional Layer, k-max pooling Layer, multi-layers GRU and a softmax classifier.

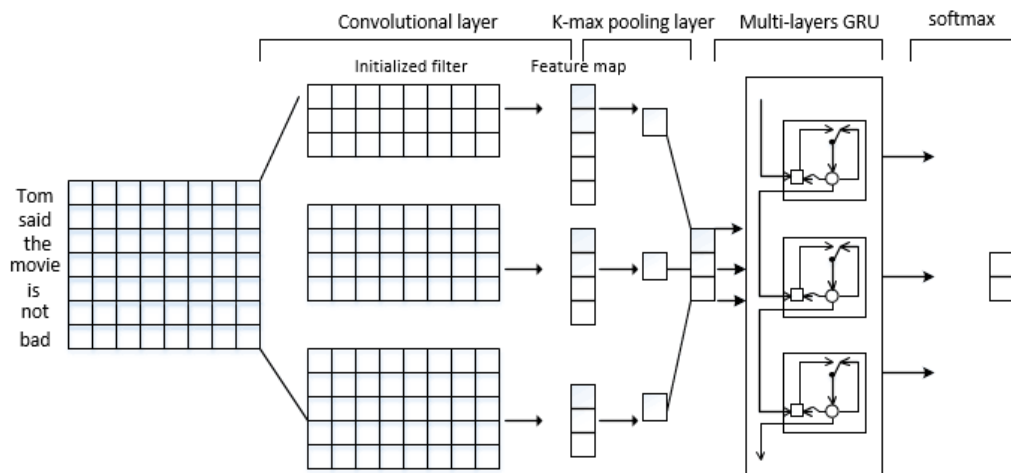


Fig. 2. The structure of proposed model: IF-CGNN.

Word embedding

Traditional word representations, such as one-hot vectors, face the two main problems: losing word order and oversize of dimensionality. Compared to one-hot representations of word embedding, distributed representations of word embedding is more suitable and more powerful. Assume that a text has N words, wr_n with $n \in [1, N]$ represents the vector of the n th words in the text. Given a text with words wr_n , IF-CGNN embeds the words to vectors through an embedding matrix W_e . The x_n is the vector representation of wr_n , which is formulated by Eq. (6)

$$x_n = W_e wr_n \quad (6)$$

Convolutional layer

In IF-CGNN, the convolutional layer is used to capture the sequence information with initialized filters and reduce the dimensions of the input data. The convolutional operation in the convolutional layer is conducted. In the convolutional layer, 100 filters with windows size of 3 move on the textual representation to extract the features. As the filter moves on, many sequences, which capture the syntactic and semantic features, are generated. The illustration of the convolutional layer is shown as Fig. 3. When the sequence of “the weather is not bad” do the operation with initialized filters, the filters find “not bad” is important feature, the weight of “not bad” increases.

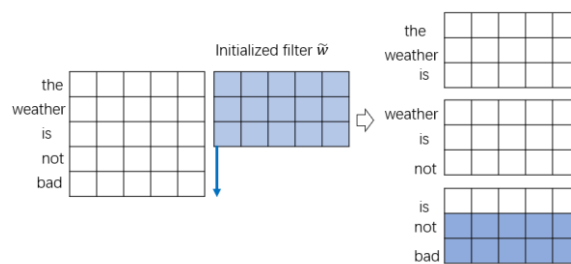


Fig. 3. The convolutional operation in IF-CGNN.

The convolutional layer, which is between k filters $\tilde{W}$ and a word embedding vector x_n , is used to obtain the features for the window of words in the corresponding feature sequences. In our model, the number of k is 100. A feature c is generated from a window of words x_h by

$$c = f(\tilde{W} \cdot x_h + b) \quad (7)$$

where $\tilde{W}$ is the weights of initialized filter, b is a bias term. $f(\cdot)$ represent the nonlinear activation function of the convolutional operation, rectified linear units (ReLU). In IF-CGNN, ReLU is used as the nonlinear activation function because it can improve the learning dynamics of the networks and significantly reduce the number of iterations required for convergence in deep networks.

As filters do convolutional operation from top to bottom in text, we can obtain a feature map $c = [c_1, c_2, \dots, c_{n-h+1}]$.

k-max pooling layer

In natural language processing, pooling layer can convert a variable-length tensor into a fixed length. The pooling layer obtains more important information by subsampling. Average pooling layer and max-pooling layer are two types of pooling layers. In order to keep the order of features, the k-max pooling layer is applied in the network after the convolutional layer. It selects the subsequence p_{max}^k of the k highest values of p , where k is a value and p is feature map ($p > k$) in IF-CGNN. The k-max pooling operation makes it possible to pool the k most active features in p that may be a number of positions apart. it preserves the order of the features, but is insensitive to their specific positions. This guarantees that the input to the next layers is independent of the length of the input sentence. By adjusting k values, the more important map in each feature sequences are extracted, generating a maximum pooling sampling result matrix.

Multi-layers GRU

Text classification is the processing of sequential information. However, the feature sequences obtained in a parallel way from the k-max pooling layer do not fully use sequence information. GRU is specialized for sequential modelling and can extract the contextual information from the feature sequences. The original GRU is comprised of a single hidden GRU layer followed by a standard feedforward output layer. Multi-layers GRU where each layer contains multiple memory

cells makes the model deeper, more accurately earning the description. The illustration of the multi-layers GRU is shown as Fig. 4.

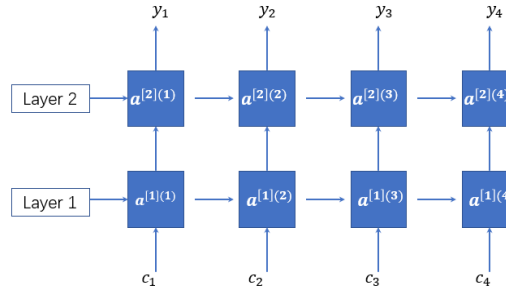


Fig. 4. Multi-layers GRU in IF-CGNN.

The effect of multi-layers GRU is to build the text-level word vector representation. Hence, GRU can improve classification efficiency. GRU obtains the annotations of words by summarizing information for words, and the annotations incorporate the contextual information. $\overrightarrow{GRU}$ reads the feature sequences from c_1 to c_{100} in one layer. The outputs of multi-layers GRU are stated as follows:

$$\vec{h} = \{\overrightarrow{GRU}(c_p)\}_n, p \in (1,100) \quad (8)$$

Where, n is the number of layers, an annotation for a given feature sequence c_p is obtained by the hidden state $\vec{h}$. This state summarizes the information of the entire text centered around c_p . In this way, the context representations are obtained for text classification.

In IF-CGNN, the dropout layer and the softmax layer are used to generate the conditional probabilities over the class space to achieve classification. Dropout has a probability to be temporarily removed from the network during the process of training, thus dropout layer can avoid overfitting.

The cross entropy is a commonly used loss function to evaluate the classification performance of the models. It is often better than the classification error rate or the mean square error. In our approach, Adam optimizer [27] is chosen to optimize the loss function of the network. The model parameters are finetuned by Adam optimizer which has been shown as an effective and efficient backpropagation algorithm. The cross entropy as the loss function can reduce the risk of a gradient disappearance during the process of stochastic gradient descent. The loss function can be denoted as follows in Eq. (9)

$$L = -\frac{1}{n} \sum_{ts} [y \log o + (1 - y) \log(1 - o)] \quad (9)$$

where n is the number of training samples, ts represents the training sample, y is the label of the sample, o is the output of the model.

4 Experiment

Experiments are conducted to evaluate the performance of the proposed approach for text classification on various benchmarking datasets. In this section, the experimental setup and baseline methods followed by the discussion of results are described.

4.1 Experimental Setup

All the experiments are tested with the computer with configuration described as follows:

OS system: Ubuntu 14.04 LTS; GPU Memory:16GB; Python: 3.5.2; Tensorflow:1.7.0.

Datasets

Our model is evaluated on text classification task (including three general text classification and a clinical text classification) using the following datasets. Statistics of these datasets are demonstrated in Table 1.

MR: Movie review sentence polarity dataset is about movie reviews and it contains positive snippets and negative snippets extracted from Rotten Tomatoes web site pages. The classification involves positive reviews and negative reviews [24].

Subj: The subjectivity dataset consists of subjective reviews and objective plot summaries. The task of subjectivity dataset is to classify the text as being subjective or objective [25].

CR: This dataset is a customer reviews dataset about various products (cameras, MP3setc). Task is to predict positive reviews and negative reviews [26].

TCM: Traditional Chinese medicine clinical records from Classified Medical Records of Distinguished Physicians Continued Two (Er Xu Ming Yt Lei An in Chinese, ISBN 7-5381-2372-5). The datasets involves five categories (Internal Medicine, Surgery, Gynaecology, Ear-Nose-Throat & Stomatology and Paediatrics)[28].

Table 1. Dataset statistics.

Dataset	Dataset size	Vocabulary size
MR	10662	18765
CR	3775	5340
Subj	10000	21323
TCM	7037	69530

Experimental Parameters

Our experiments use accuracy as the evaluation metric to measure the overall classification performance. Publicly available word vectors trained on the 100 billion words from Google News are used as pre-trained word embeddings. The vectors have dimensionality of 300 and were trained using the continuous bag-of-words architecture. Word not present in the set of pre-trained words are initialized randomly. The number of filters of length 3 is set to be 100 in the convolutional layer. The memory dimension of GRU is set to be 64. The training batch size for all datasets is set as 50. The dropout rate is 0.5. A backpropagation algorithm with Adam stochastic optimization method is used to train the network through time with the learning rate of 0.001. After each training epoch, the network is tested on validation data. The log-likelihood of validation data is computed for convergence detection.

4.2 Baseline methods

This paper benchmarks the following baseline methods for text classification, they are effective methods and have achieved some good results in text classification:

CNN: Convolutional neural network with pre-trained word embedding vector from word2vec. It is a classical convolutional neural network in text classification which is proposed by Kim [4].

LSTM: Long Short Term Memory. It is a classical recurrent neural network[29].

IFCNN: The model uses the n-grams to initializing the filter, then apply this filter to the shallow convolutional neural network. The method is improved on the basis of CNN by Kim [23].

NBSVM: SVM variant using naive Bayes log-count ratios as feature values proposed by Wang and Manning [22]. It is a traditional machine learning method.

PV: The method is about an unsupervised method that learns fixed-length feature representations from variable-length pieces of texts [21].

Combined-skip: The method through an encoder-decoder model tries to reconstruct the surrounding sentences of an encoded passage [20].

CNN-Rule: The method applies declarative first-order logic rules to the convolution neural network [19].

The comparisons of our method against the six baselines are demonstrated in Table 2.

Table 2. The classification accuracy of proposed method against other models on three datasets (%).

Methods	CR	MR	Subj
CNN (Kim, 2014)	84.3	81.5	93.4
IFCNN (Li, 2017)	86.0	82.1	93.7
CNN-Rule (Hu et al., 2016)	81.7	85.3	-
NBSVM(Wang et al., 2012)	81.8	79.4	93.2
PV(Le et al., 2014)	78.1	74.8	90.5
Combine-skip(Kiros et al., 2015)	-	76.5	93.6
LSTM	80.0	80.1	91.3
IF-CGNN	92.1	82.7	94.5

4.3 Results

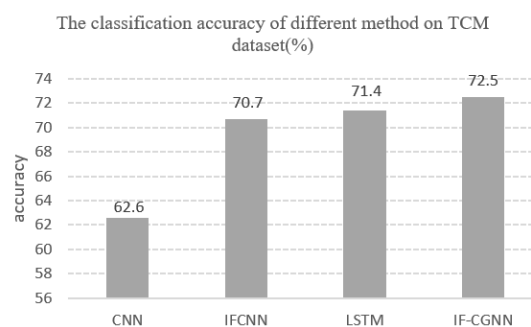
The comparison results for datasets (MR, CR and Subj) are presented in Table 2. The experimental results are evaluated by the classification accuracy. The best results are shown in boldface. From Table 2, IF-CGNN achieves better results than other methods on the majority of the benchmark datasets. Among the seven approaches mentioned above, our approach outperforms other baselines on all datasets except MR. The results of IF-CGNN are 92.1%, 82.7%, 94.5% for CR, MR, Subj. IF-CGNN gives the relative improvements of 6.1% compared to IF-CGNN on CR dataset, respectively.

Firstly, compared with three baselines (CNN, IF-CNN, LSTM), it is observed that the accuracy of our proposed model IF-CGNN increases significantly. The reason is that we fully combine the advantages of CNN and RNN. With passing the important features extracted by CNN through the sequence structure, we take the contextual information of text into consideration. In addition, we use the centroid vectors of the n-grams cluster to initialize the filters. This method allows the text to derive important features directly from the training data, reducing the impact of random initialization.

Secondly, compared with the method CNN-Rule that considers the structure of the word in the text and takes the contextual information into account. The experimental results show that our model IF-CGNN is superior to it in the CR dataset, but CNN-Rule gets the best on the MR data set. It is fully proved that the contextual information of the sentence has a great influence on the classification performance.

Finally, comparing the unsupervised text classification method PV, our method is supervised. A comparison of the three data sets reveals that our methods are all superior to PV, thus confirming the widespread view that supervised machine learning algorithms are generally superior to unsupervised machine learning algorithms.

4.4 Results on medical datasets

**Fig. 5.** classification accuracy of different method on TCM dataset (%).

From Fig.5. we know that the results of TCM dataset are 72.5%, 71.4%, 70.7% ,62.6% for IF-CGNN, LSTM, IFCNN, CNN. Compared with two classical neural networks and one recent model, it is observed that the accuracy of our proposed model IF-CGNN increases significantly. Therefore, we can know that this method can be used in the medical field.

4.5 Further analysis

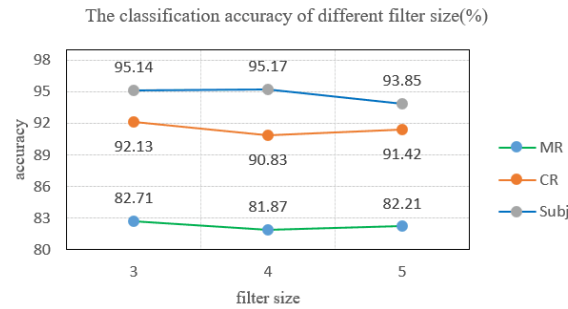


Fig. 6. Different filter size for text classification.

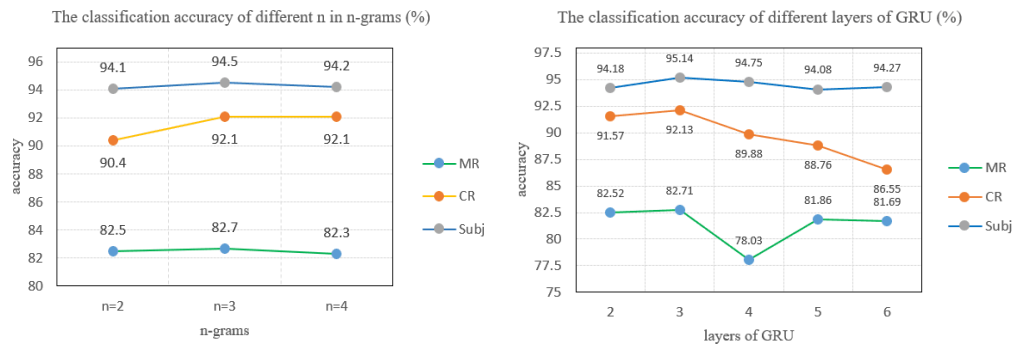


Fig. 7. (a) Left: different n-grams for text classification. (b) right: layers of multi-layers GRU for text classification.

The Fig.6 shows that the classification accuracy difference between different filter sizes (3, 4, 5) is less than 1% on three datasets (MR, CR, Subj). However, for general convolutional neural networks, different filter sizes have a large impact on the classification results. Compare with the general convolutional neural networks, it shows that the model with initialized filters is less affected by filter sizes adjustment. Thus, we can know that our model is less affected by parameter tuning of convolutional layer.

From Fig.7 (a) comparison on different n in n-grams for text classification, we find that the performance of text classification is best in MR, CR, Subj datasets when n=3, but the accuracy in n=2, n=4 and n=3 are basically equal. It shows that the different n values in n-grams have little effect on the classification, and the n values in the n-grams are freely selectable during text pre-training.

In the Fig.7 (b), we find that the accuracy of text classification is best when the number of GRU layers is 3. At the beginning, with the increase of the number of GRU layers, the classification accuracy of the model increases, but when the number of GRU layers exceeds 3, the classification accuracy decreases as the number of layers increases. This result shows that the number of layers of the GRU has a great influence on the model. Although the information of network learning is deeper as the number of layers increases, when the number of layers is too large, the result of classification will be reduced.

5 Conclusion and Future Work

In this paper, we propose a new model IF- CGNN, which combines the advantages of CNNs and RNNs. The model consists of convolutional layer, k-max pooling layer, multi-layers GRU and a softmax classifier. With inputting the important features to GRU of sequence structure, the model can make full use of contextual information in the sentence. Besides, the model uses the initialized filters in order to reduce the time caused by the parameter tuning in convolutional layer. During the process of initializing filters, we cluster the important features with K-means and use the centroid vectors of clusters to initialize the filters. After that, semantic features are encoded into filters. In experiments, we

evaluate our model on three sentiment datasets and one medical dataset. The experimental results show that our model significantly and consistently outperforms state-of-the-art feature-based methods and neural networks-based models.

In the future, we will do the following work: our model use the standard convolutional structure. It can be replaced with tree-structured convolutions or other variants of convolutions.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No.2017YFE0117500) and Science and Technology Committee of Shanghai Municipality (No. 16010500400).

References

1. Sebastiani, Fabrizio. "Machine learning in automated text categorization." *ACM Comput. Surv.* 34 (2002): 1-47.
2. Aggarwal, Charu C. and ChengXiang Zhai. "A Survey of Text Clustering Algorithms." *Mining Text Data* (2012).
3. Wang, Sida I. and Christopher D. Manning. "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification." *ACL* (2012).
4. Wallach, Hanna M.. "Topic modeling: beyond bag-of-words." *ICML* (2006).
5. Collobert, Ronan et al. "Natural Language Processing (almost) from Scratch." *Journal of Machine Learning Research* 12 (2011): 2493-2537.
6. Hinton, Geoffrey E. et al. "Improving neural networks by preventing co-adaptation of feature detectors." *CoRR abs/1207.0580* (2012): n. pag.
7. Kim, Yoon. "Convolutional Neural Networks for Sentence Classification." *EMNLP* (2014).
8. Johnson, Rie and Tong Zhang. "Effective Use of Word Order for Text Categorization with Convolutional Neural Networks." *HLT-NAACL* (2015).
9. Zhang, Yingjie et al. "MGNC-CNN: A Simple Approach to Exploiting Multiple Word Embeddings for Sentence Classification." *HLT-NAACL* (2016).
10. Schuster, Mike and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." *IEEE Trans. Signal Processing* 45 (1997): 2673-2681.
11. Chung, Junyoung et al. "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." *CoRR abs/1412.3555* (2014): n. pag.
12. Kalchbrenner, Nal, E. Grefenstette, and P. Blunsom. "A Convolutional Neural Network for Modelling Sentences." *Eprint Arxiv* 1(2014).
13. Johnson, Rie and Tong Zhang. "Deep Pyramid Convolutional Neural Networks for Text Categorization." *ACL* (2017).
14. Liu, Pengfei et al. "Recurrent Neural Network for Text Classification with Multi-Task Learning." *IJCAI* (2016).
15. Yang, Zichao et al. "Hierarchical Attention Networks for Document Classification." *HLT-NAACL* (2016).
16. Wang, Baoxin. "Disconnected Recurrent Neural Networks for Text Categorization." *ACL 2018* (2018).
17. Lai, Siwei et al. "Recurrent Convolutional Neural Networks for Text Classification." *AAAI* (2015).
18. Zhou, Chunting et al. "A C-LSTM Neural Network for Text Classification." *CoRR abs/1511.08630* (2015): n. pag.
19. Hu, Zhiting et al. "Harnessing Deep Neural Networks with Logic Rules." *CoRRabs/1603.06318* (2016): n. pag.
20. Kiros, Ryan et al. "Skip-Thought Vectors." *NIPS* (2015).
21. Le, Quoc V. and Tomas Mikolov. "Distributed Representations of Sentences and Documents." *ICML* (2014).
22. Wang, Sida I. and Christopher D. Manning. "Baselines and Bigrams: Simple, Good Sentiment and Topic Classification." *ACL* (2012).
23. Li, Shen et al. "Initializing Convolutional Filters with Semantic Features for Text Classification." *EMNLP* (2017).
24. Pang, Bo and Lillian Lee. "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales." *ACL* (2005).

25. Pang, L. Lee. "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts." *ACL* (2004)
26. Hu, B. Liu. Mining and Summarizing Customer Reviews. *ACM SIGKDD* (2004).
27. D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: *Proceedings of the 3rd International Conference for Learning Representations*, Springer Verlag, San Diego, CA, United states, 2015, pp. 1–15
28. Yao, Liang et al. "Traditional Chinese medicine clinical records classification using knowledge-powered document embedding." *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2016): 1926-1928.
29. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
30. Zhang, Xiang et al. "Character-level Convolutional Networks for Text Classification." *NIPS* (2015).
31. J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: <http://arxiv.org/abs/1412.3555>