Text Classification using gated fusion of n-gram features and semantic features

Ajay Nagar[1], Anmol Bhasin[1], Gaurav Mathur[1]

[1]Samsung R&D Institute India - Bangalore, India
ajay.nagar@samsung.com, anmol.bhasin@samsung.com, gaurav.m4@samsung.com

Abstract. We introduce a novel method for text classification based on gated fusion of n-gram features and semantic features of the text. The parallel CNN network captures the n-gram relation between the words based on the filter size, primarily short distance multi-word relations. Whereas for semantic relationship, universal sentence encoder or BiLSTM is used. Gated fusion is used to combine n-gram and semantic features. The model is evaluated on 4 commonly used benchmark datasets (MR, TREC, AG-News and SUBJ), which includes sentiment analysis and question classification. The proposed method is able to surpass the existing state-of-the-art DNN architectures for text classification on these datasets.

Keywords: Text Classification, Convolution Neural Network, Universal Sentence Encoder, BiLSTM.

1 Introduction

Deep learning models have revealed amazing results in numerous Natural Language Processing(NLP) tasks such as Neural Machine Translation (NMT) [1], Named Entity Recognition (NER) [2], Text Summarization [3], Text Classification [4] etc. Among these, text classification is one of the important and challenging task in NLP which aims to assign predefined relevant categories to natural language texts. It is useful in many applications like social media text analysis, sentiment analysis applications, business analysis applications, feedback analysis applications etc. Since, there is no complete set of predefined rules for natural languages, classification algorithms are unable to capture complex semantics of the text.

1.1 Prior Work

Feature representation for text classification is a crucial problem. Initially, bag-of-words model, which uses unigrams, bigrams, n-grams, were used for feature representation. Later, Mikolov et al. [5] proposed distributed representation of words to solve the data sparsity problem and loss of semantic information of words. Character embedding and sentence embedding are the other types of embedding used for text classification. Word2vec [6] and GloVe [7] are two pre-trained word embedding commonly used for text classification.

Different deep learning architectures has been applied for text classification to learn different features. Socher et al., [8] proposed the Recursive Neural Network for text classification by modelling sentence representation. Since, text classification problem has sequential nature, Recurrent Neural Networks (RNN) and its variants Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM) were used to learn long distance dependencies or semantics of text. Yoon Kim [4] proposed Convolution Neural Network (CNN) for text classification using pre-trained word embedding as input to learn n-gram features. CNN captures local correlations of spatial or temporal structures but loses the context of text. To take the advantage of both models, Siwei et al., [9] proposed recurrent convolution neural network for text classification to first provide context to each word using RNN and then use CNN to find n-gram features. Zhou et al., [10] also proposed C-LSTM neural network for text classification by applying CNN first and then LSTM. Since both architectures used CNN and RNN sequentially, error in CNN network is also propagated to RNN and vice versa. Sequential nature of these architectures may lead to erroneous n-gram features, semantic features or long distance dependencies.

1.2 About Our Work

To address the above problem, we propose a novel architecture for text classification using the gated fusion of n-gram features and semantic features. The intention of this work is robust text classification by modelling n-gram features and long distance dependencies or semantics of text (representation of sentence as a embedding) more concretely. Figure 1 represents higher level block diagram of our proposed network.

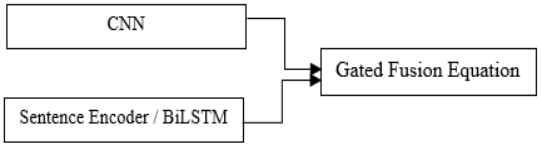

Fig. 1. Higher Level Block Diagram

2 Proposed Models

As discussed in previous section, both n-gram features and semantic features were considered for classification by taking the advantage of CNN, LSTM and Sentence Encoder. To avoid error propagation from n-gram features extraction to semantic features or vice-a-versa, we trained both networks independently. The architecture loss was minimized based on summation of loss of two models. The CNN was used to capture n-gram features. For long distance dependencies or semantic features, in one model we used Bidirectional LSTM and in another we used universal sentence encoder. For fusing both model outputs we used gated fusion equation. Details of two models are discussed in following section.

2.1 Model 1 : Gated Fusion on CNN and Bidirectional LSTM

The basic idea is to process the input sentence in two parallel network as shown in the Figure 2. For this, first we tokenize the sentence by splitting at space to get the word sequence. To represent words as distributed dense vectors, we used L1 dimensional GloVe [7] pre-trained word embedding. Unseen words (words not present in GloVe word embedding) were represented as dense vector using uniform random initialization of L1 dimension. For length of input sequence average sentence length L2 was used. In this way word embedding matrix of dimension L2*L1 for a input sentence was created.

For extracting n-gram features, we used Kim's CNN model [4] as baseline model. Word embedding matrix of a sentence was passed to 4 parallel CNN layers having different filter sizes f1*L1, f2*L1, f3*L1 and f4*L1. The filter height was set to f1=1, f2=2, f3=3 and f4=5, where height represents number of words to be convolved to capture unigram, bigram and n-grams. 128 filters of each type were taken. After the convolution layer max pooling layer was used to compute most important feature from the output of every convolution. We got 128 features from each convolution layer. These 128 features from every layer were concatenated and a dense vector of size 512 was obtained which constitute n-gram features of a sentence.

Since, LSTMs are able to captures the long distance dependencies in sequential data, we fed word embedding matrix of a sentence to bidirectional LSTM layer to model long distance dependencies or semantic features. We utilized bidirectional LSTM to provide forward and reverse context of the text to the network. We considered 256 hidden units in LSTM. The output of this layer was a dense vector of size 512 (concatenated output of forward LSTM and backward LSTM) which can be interpreted as semantics (long distance dependences) of complete text.

The CNN model output and LSTM model output was given as input to gated fusion equation (Z). The output of gated fusion equation was passed to dropout layer and then to fully connected layer and finally, a softmax layer was used for classification.

$$Z = t \,\Theta\, g\,(W_H^1 y^1 + b_H^1) + (1 - t) \,\Theta\, g\,(W_H^2 y^2 + b_H^2) \quad (1)$$

$$t = \sigma\,(W_T\, y + b_T) \quad (2)$$

Equation (1) represents gated fusion equation, where g is a nonlinear activation function and for our experiment, we used it as 'relu'. Equation 2 i.e. 't' is called the weightage gate. It represents the weightage given to n-gram features and (1−t) represents the weightage given to long distance dependencies or semantic features of text. The features generated by the CNN layer and the bidirectional LSTM layer are averaged to generate 'y'. We generated weights by applying sigmoid to 'y' and that are learnable. The intention of using gated fusion was to make model learn to choose itself between features generated by CNN and Bi-LSTM. Since, some texts can be best classified based on short-term dependences and some can be best classified based

on long distance dependences, so for model to itself decide the weightage for both these dependences we used gated fusion.

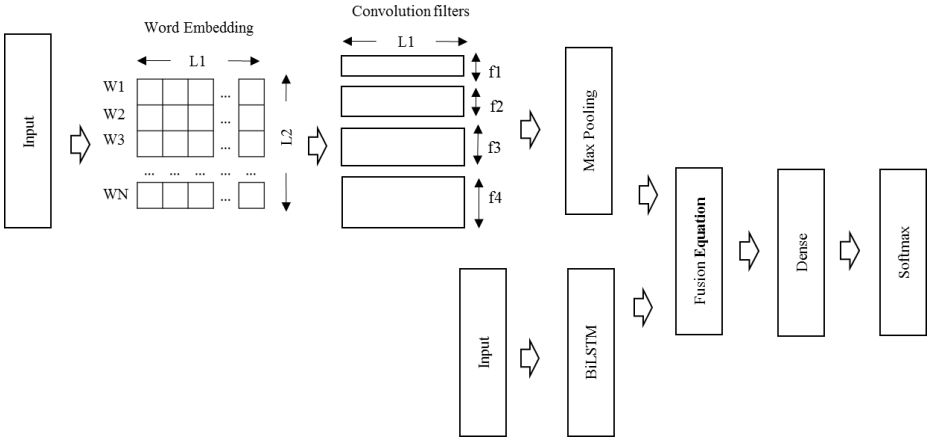

Fig. 2. Architecture of using Gated Fusion Equation on CNN and Bidirectional LSTM.

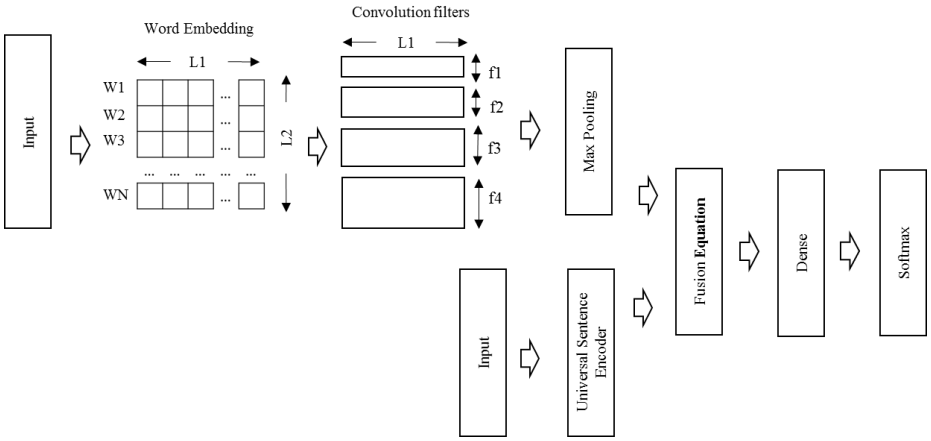

Fig. 3. Architecture of using Gated Fusion Equation on CNN and Universal Sentence Encoder.

2.2 Model 2 : Gated Fusion on CNN and Universal Sentence Encoder

In this model as shown in Figure 3, we used goggle's pre-trained universal sentence encoder instead of bidirectional LSTM to obtain semantic features of the sentence. Since, all the datasets were small, in Model 1 Bi-LSTM was not capable of capturing the complete semantics of text. Universal sentence encoder is trained on huge datasets

from web, Wikipedia etc. It takes sentence as input and gives a dense vector of output size 512. This vector represents the semantics of a sentence.

For n-gram features, first, we construct the word embedding matrix using input sentence as described for Model 1 and then applied convolution layer, max-pooling layer and flatten layer in sequence. Kernel sizes, number of kernels, pooling and output dimensions are same as previous model (Model 1).

Similar to previous model, features generated by both the CNN layer and the universal sentence encoder are averaged to obtain y (Equation 1). The CNN network output and Sentence Encoder output was passed to gated fusion equation (similar to Model 1). The output of fusion layer was then passed to fully connected layer and finally a softmax layer was used to predict the class.

3 Datasets and Experiment Details

In this section, we first present the benchmark datasets we used for our experiments and then experiment details.

3.1 Datasets

We carried out our experiments with 4 benchmark datasets, Table 1 shows the distribution of training and testing data for all four datasets along with number of classes to be predicted.

Table 1. Benchmark Datasets.

Dataset	Train Data	Test Data	Classes
MR	10662	CV	2
TREC	5952	500	6
AG News	120000	7600	4
SUBJ	10000	CV	2

MR: Positive and Negative movie reviews dataset. Aim is to identify a movie review, positive or negative [12]. There is no test dataset defined, so fivefold cross validation (CV) was done.

TREC: This dataset contains 6 types of questions. Objective is to identify the class for given question [13].

AG News: Topic Classification Dataset. Aim is to classify news into different classes. [14]

SUBJ: This dataset has sentences and task is to classify a sentence into objective or subjective type [15]. There is no test dataset defined, so for this also fivefold cross validation (CV) was done.

3.2 Experiment Details

For all the datasets, training was carried out using mini batch gradient descent with batch size of 64. For binary classification 'binary cross entropy' was used and for other datasets 'categorical cross entropy' loss was being used. We used 'Adam' as an optimizer. No of epoch for each model was taken as 50, though for all the datasets the model converged well below 50 epochs. We used 0.5 dropout rate to reduce overfitting of data.

4 Results and Discussion

In this section, we first present our state-of-the-art results on datasets we used for our experiments and then result discussion.

4.1 Results

We evaluated the proposed methods on different benchmark datasets and compared with state-of-the-art results to show effectiveness of the method. Results of both the models are shown in Table 2. We evaluated the datasets on basis of overall accuracy i.e. correct predictions divided by total number of predictions for test dataset. 5 fold cross-validation is used for cross validation datasets. We observed that using both n-gram features and features generated by universal sentence encoder are able to achieve remarkable results. Using these features, we are able to surpass all the results achieved by multichannel CNN model. Compared to C-LSTM model, we got 1.2% accuracy improvement on TREC dataset. Using n-gram features and features generated by RNN, we are able to achieve approximately same results to other models.

Table 2. Accuracy on different datasets achieved by our models.

Models	Dataset			
	MR	AG News	TREC	SUBJ
Yoon Kim [4]	81.5	86.1	93.6	93.4
CharCNN [14]	77	78.3	76	-
WCCNN [16]	83.8	85.6	91.2	-
KPCN [16]	83.3	88.4	93.5	-
C-LSTM [10]	-	-	94.6	-
BiLSTM-CRF	82.3	-	-	-
F-Dropout [4]	79.1	-	-	93.6
Model 1	79.71[1]	88.26	93.8	92.61[1]
	80.39[2]			93.6[2]
Model 2	**83.4[1]**	**88.75**	**95.8**	**94.95[1]**
	84.43[2]			95.85[2]

[1] CV

[2] Max Accuracy

4.2 Discussion

It is observed that with the gated fusion of n-gram features by CNN and features generated by RNN, model was able to achieve only the comparable results but with the gated fusion of n-gram features by CNN and semantics by pre-trained universal sentence encoder, model is able to achieve the state-of-the-art results. Compared with the existing methods, that are using both CNN and RNN for text classification, proposed model performs better. It is noticed that gated fusion can choose between short distance dependency based classification and long distance based dependency classification.

5 Conclusion

In this work, we proposed a method for text classification using n-gram features and semantic features captured by convolution neural network (CNN) and universal sentence encoder respectively. Proposed models are able to achieve state-of-the-art results on four common benchmark datasets. The proposed method can be applied to many other natural language processing tasks such as sentence similarity, machine translation (as an encoder) etc.

References

1. Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).
2. Lee, Ji Young, and Franck Dernoncourt. "Sequential short-text classification with recurrent and convolutional neural networks." *arXiv preprint arXiv:1603.03827* (2016).
3. Allahyari, Mehdi, et al. "Text summarization techniques: a brief survey." *arXiv preprint arXiv:1707.02268* (2017).
4. Kim, Yoon. "Convolutional neural networks for sentence classification." *arXiv preprint arXiv:1408.5882* (2014).
5. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).
6. Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.
7. Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.
8. Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013.
9. Lai, Siwei, et al. "Recurrent convolutional neural networks for text classification." *Twenty-ninth AAAI conference on artificial intelligence*. 2015.

10. Zhou, Chunting, et al. "A C-LSTM neural network for text classification." *arXiv preprint arXiv:1511.08630* (2015).
11. Cer, Daniel, et al. "Universal sentence encoder." *arXiv preprint arXiv:1803.11175* (2018).
12. Pang, Bo, and Lillian Lee. "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales." *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2005.
13. Li, Xin, and Dan Roth. "Learning question classifiers." *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002.
14. Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." *Advances in neural information processing systems*. 2015.
15. Pang, Bo, and Lillian Lee. "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts." *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004.
16. Wang, Jin, et al. "Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification." *IJCAI*. 2017.