Recurrent Convolutional Network in Toxic Comment Detection

Tianyu Yang; Wen Cui
Georgetown University, Washington, U.S.

Abstract

This project focuses on exploring and establishing best model structure for the Kaggle toxic comment classification. Models were evaluated on average ROC AUC for different toxicity labels.

We implemented four structures of deep learning model and compared with a logistic regression baseline. All deep learning models beated logistic regression and the bidirectional lstm with max-pooling model, outperforming all others, landed us around the 55 percentiles

Introduction

The appearances and evolution of online communication platforms offers convenient way of sharing information and opinions. Spawn by this ultimate convenience and lack of strictly enforced regulation, harassment and language abuse become extremely common. Although the intention of expressing negativity cannot be eradicated, platform providers are responsible for providing participants a healthy community by filtering extremely hurtful and disruptive content.

In this project, we evaluated several classification approaches that identify toxic comments and found that the recurrent network with max pooling combined with various dropout regularization outperforms all other models and could be a stepping stone to tackle the problem.

Related Work

General non neural approaches for text classification mainly consist of classification models such as logistic regression, support vector machines etc., with human extracted features or distribution based vector(TF-IDF matrix, Term-Document Matrix). This group of techniques(especially with extracted feature) has computation advantage during training. Also, when important patterns are captured by the extracted features, the model can perform well. However, in the realm of natural language processing, feature extraction requires not only understanding of linguistics but also heavily depends on domain knowledge. The entire process is time consuming.

Hence, neural based approaches which does not require such process, gained abundance of popularity when machine computation capacity exponentially increased. In neural based approaches, text is converted to sequence of semantically vectorized words and various neuron layers can be applied.

Dataset

The dataset contains over ten thousands labeled comments. Possible labels include 'toxic', 'severe toxic', 'obscene', 'threat','insult','identity hate'. 'toxic' is a general label for toxicity. There are 159571 observations, of which 15294 are 'toxic', 1595 are 'severe toxic', 8449 are 'obscene', 478 are 'threat', 7877 are 'insult', 1405 are 'identity hate'. The comments are unprocessed and appear in various forms including but not restricted to capitalization, quotation, misspelling, emoticon.

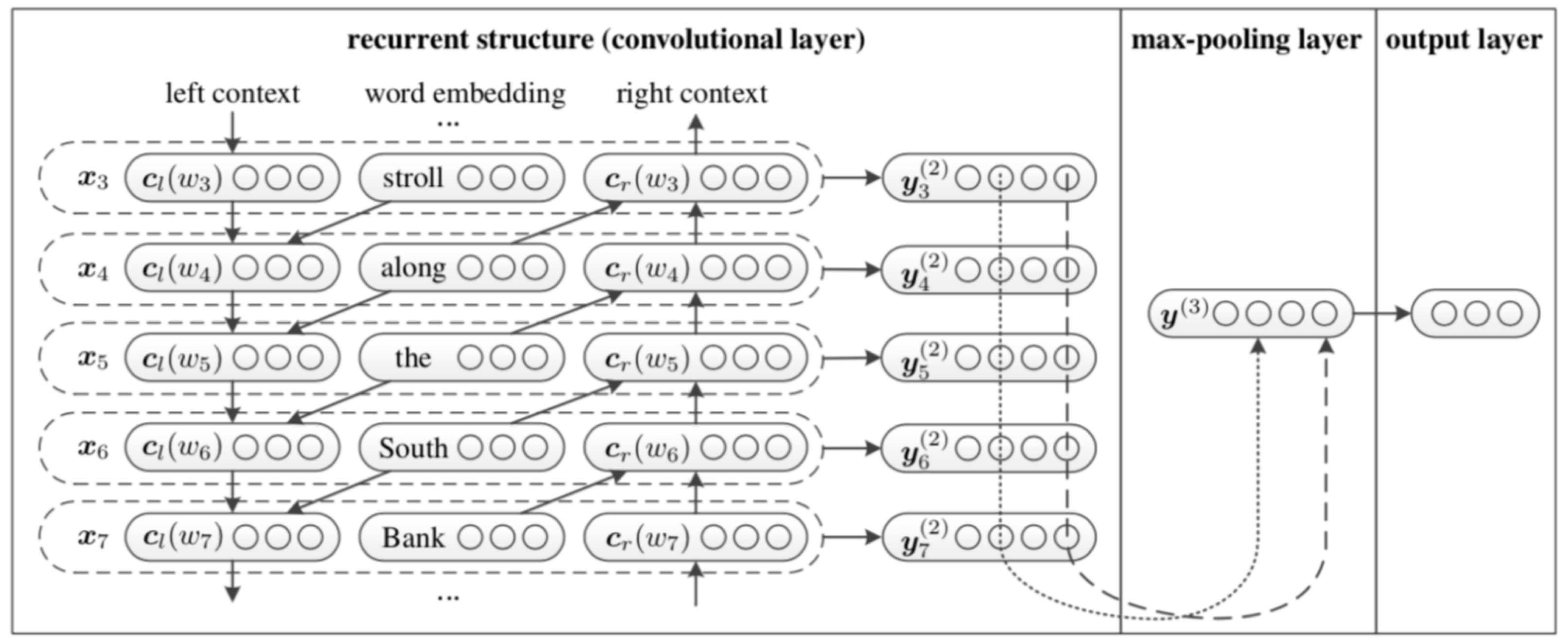

Fig. 1. Best performing model structure

Models

We compared a logistic regression model, the baseline model and three deep learning structures. Since the classes are not mutually exclusive and an observation can have any number of labels, we fit six logistic regression models(one for each label) using bag of words features including unigrams and bigrams. To reduce the dimension of the vectorized words, we removed stop words and converted all comments to lower case.

The first layer of all networks is an embedding layer. We used the Glove pretrained word embedding with 300 dimensions.

The first model(bilstm) contains one layer of 128 bidirectional LSTM cells on top of the embedding layer. Each LSTM cell produces a scalar output instead of a sequence of output. The outputs are fed into a dense layer with six output units activated by sigmoid function, that produce the probability for each class. Dropout on embedding layer and recurrent layer is applied to reduce generalization error.

The second model(maxbilstm) refers to [3]. We allowed the recurrent layer to produce sequence of output and applied global max pooling on the produced sequences. The rest of the structure is identical to the first model.

The third model(mixbilstm) refers to [4]. In the pooling layer, we concatenated the results of global max pool and global average pool and applied dropout of 0.5 probability to represent the stochastic pooling process as keras does not provide the implementation.

The fourth model(mixconv) was inspired by [3]. Instead of regarding the recurrent layer as a 'convolution' layer, we added a real convolution layer with 64 filters of kernel size three between the recurrent layer and mixed pooling layer. The convolution layer has the potential to better capture closer words(up to trigrams) and filter noise from words placed far from each other.

All deep learning models were trained with maximum of 10 epochs and batch size of 256 and validation loss as early stopping criterion. All models finished training before reaching the epoch bound

Labels	Models				
	bilstm	maxbilsm	mixbilstm	mixconv	LR
toxic	0.9798	0.9809	0.9801	0.9818	0.9496
severe toxic	0.9911	0.9910	0.9908	0.9897	0.9385
obscene	0.9930	0.9935	0.9935	0.9931	0.9560
threat	0.9725	0.9784	0.9621	0.9705	0.9544
insult	0.9865	0.9876	0.9870	0.9868	0.9436
identity hate	0.9799	0.9850	0.9813	0.9833	0.9311
average	0.9838	0.9861	0.9825	0.9842	0.9455

Fig.2. ROC AUC for predictions. Row max are highlighted.

Results and Analysis

Models were evaluated with area under the Receiver Operator Curve(rocauc) for each class. The result in Figure 2 shows that deep learning models outperformed logistic regression with bag of word features for all labels with large margin. 'maxbilstm' model received the best performance in most categories and has the highest average rocauc.

For overall toxicity, recurrent convolutional structure with actual convolution layer and mixed pooling(mixconv) surpasses maxbilstm and mixbilstm, suggesting that the convolution layer, that creates phrase level representation, better capture the characteristic of overall toxicity. However, for specific toxicity categories that may depend more on individual words, the convolution layer that blurs the individual effect of words does not help improve.

The mixed pooling layer was not able to improve the classification result. We speculated that the implementation was a simplified version of \cite{b4} given the keras framework and cannot achieve the effect described in the paper.

We also submitted the prediction the official test set for evaluation and were placed around 55 percentile.

Conclusions and Future Works

This project further proves the advantage of deep learning method in text classification and compared popular deep learning structures in the context of toxic comment identification. The bidirectional lstm model with max pooling has aggregated advantage whereas other models each leads performance in only one or few categories.

The result and ranking does not generalize to all text classification problems as different domains have unique linguistic features that requires various structures to capture.

To further improve the result, future work could focus on 1) a precise implementation of stochastic pooling; 2) exploration of different combinations of structures; 3) better preprocessing on the text and adding extracted features.

Contact

Tianyu Yang
Georgetown University
Washington DC, U.S.
Email: ty233@georgetown.edu

Wen Cui
Georgetown University
Washington DC, U.S
Email: wc692@Georgetown.edu

References

1. Mikolov, Tomáš / Karafiát, Martin / Burget, Lukáš / Černocký, Jan / Khudanpur, Sanjeev (2010): "Recurrent neural network based language model", In INTERSPEECH-2010, 1045-1048.
2. Sundermeyer, Martin / Schlüter, Ralf / Ney, Hermann (2012): "LSTM neural networks for language modeling", In INTERSPEECH-2012, 194-197.
3. LAI, S.; XU, L.; LIU, K.; ZHAO, J.. Recurrent Convolutional Neural Networks for Text Classification. AAAI Conference on Artificial Intelligence, North America, feb. 2015
4. Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The Journal of Machine Learning Research 15.1 (2014): 1929-1958.
5. Yu D., Wang H., Chen P., Wei Z. (2014) Mixed Pooling for Convolutional Neural Networks. In: Miao D., Pedrycz W., Ślęzak D., Peters G., Hu Q., Wang R. (eds) Rough Sets and Knowledge Technology. RSKT 2014. Lecture Notes in Computer Science, vol 8818. Springer, Cham