11747 Assignment 1

Jiayuan Xu
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
jiayuanx@andrew.cmu.edu

1 Initial Interest Survey

Our team consists of three people, including Dahua Gan (dgan), Bei Zhou, and George Xu (jiayuanx). We are discussing potential direction of the team project, with current attention on Common Sense Reasoning and Fake News Detection (...).

2 Text Classifier Implementation

2.1 Pre-processing

During pre-processing, I cleaned the input text using regex scripts based on the one used in [1], tokenized the input string using nltk TreebankWordTokenizer, then converted to lower cases.

2.2 Model Architectures

I implemented three types of models: LSTM, Dilated CNN, and a combination of Dilated CNN and CNN for text classification. Within each class of architecture with various building blocks such as batch normalization and residual connections. All of my model architectures are made of the following architecture. It started with an Embedding layer that was trained from scratch. Each embedded input tensor would be passed through a sequential encoding model. The result was then maxpooled across time to produce a fixed length encoded vector, which was pushed through a linear layer then softmax-ed to obtain predicted probabilities for each class.

The majority of the experiment housed in the sequential encoding section. I started out with LSTM modules innate to the PyTorch library. After a few experimentations, the LSTM classifier was capped at 77% validation accuracy.

The second encoder implemented is a variation of the Dilated CNN architecture. I referenced the work by Bai et al. [2] on Temporal Convectional Networks, inplemented a three-layer DCNN/TCN with exponential dilation of [1, 2, 4]. This encoder led to a slight performance boost to validation accuracy of 80%, while still below the CNN benchmark of 81.8% [1]. Later by introducing residual connection across blocks, the model was finally able match the benchmark by reaching 82.1%.

Interested in the result, I further introduced a combination of the Dilated CNN and the CNN block used in the CNN benchmark. In detail, I separately passed the word embeddings through the aforementioned DCNN and three 1D convolutions in parallel. The output vectors are then maxpooled across time and concatenated as a single vector before entering the linear layer. This only slightly boosted the classification performance, once reaching 83.4%. However, with this architecture the validation accuracy is now consistently above 82%.

I trained eight models, with previously mentioned variations, during the last attempt to boost the performance. To obtain the final results, I generated predictions using all of the models and obtained their majority vote, hitting 84.1% on dev set.

Preprint. Under review.

2.3 Results

The key results are listed below.

Table 1: Appeal Prediction

Model	Accuracy
LSTM	77.0%
Dilated CNN (DCNN)	82.1%
Dilated CNN + CNN (DDCNN)	82.4%
Ensemble	84.1%

2.4 Discussion

After quite some experimentations, my single model performances hovered around 82%. From my experience, CNN-based architectures out-performs simple LSTM. Since the task is text classification, I suspect the lift came from CNN's strength in capturing local features, such as key terms relating to each topic. The idea of introducing Dilation into the functionality is to obtain better long-term fusion of word embedding. However the result failed add much value to the released CNN-alone baseline (81.8%), which further recognizing the hypothesis that what the model did was to locate important terms in the input text.

References

[1] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, 2014.

[2] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.