CSCI 699 Spring 2019
Machine Learning for Knowledge Extraction & Reasoning
Homework 2*

Chi Zhang

USC ID: 6099-4134-05

Department of Computer Science

University of Southern California

March 17, 2019

1 Code Repository

`https://github.com/vermouth1992/CSCI699ml4know/tree/master/hw2`

2 Development Pipeline

We started from an open source implementation of CNN for relation extraction [4] from `https://github.com/ShomyLiu/pytorch-pcnn` and try various ideas to improve the results. For model architecture, we tried CNN with multi-sized window kernel [2] and RNN with attention [5]. We also tried CNN with rank loss [1]. Some ideas are inspired by a great blog of various models for relation extraction in `http://shomy.top/2018/02/28/relation-extraction/`.

2.1 Features

We consider lexical features as described in [4]: entity1, entity2, and the left and right token of the two entities. We don't use hypernyms in WordNet since it is from extra source. For sentence level feature, we consider word embedding and position features. We use pre-trained word embedding from Glove [3] with 50 dimension to initialize our embedding layer. The position feature measures the distance of each word to the two entities. The distance index will be passed to an embedding layer as well.

We use open source nltk tokenizer (`https://www.nltk.org`) for tokenization of raw sentences.

2.2 Model Architecture

For CNN-based model, we consider PCNN as described in [2] as shown in Figure 1. Compared with single-sized window kernel in [4], multi-sized kernel is able to capture multiple n-gram features.

For RNN-based model, we consider standard BiLSTM with output attention in [5] and the architecture is shown in Figure 2. In this model, we only use sentence feature with word embedding and position embedding. The attention layer takes a weight sum of the output of BiLSTM layer. Assume $H = [h_1, h_2, \cdots, h_T]$, then the weighted sum is calculated as

$$M = \tanh(H) \tag{1}$$

$$\alpha = softmax(w^T M) \tag{2}$$

*Instructor: Xiang Ren

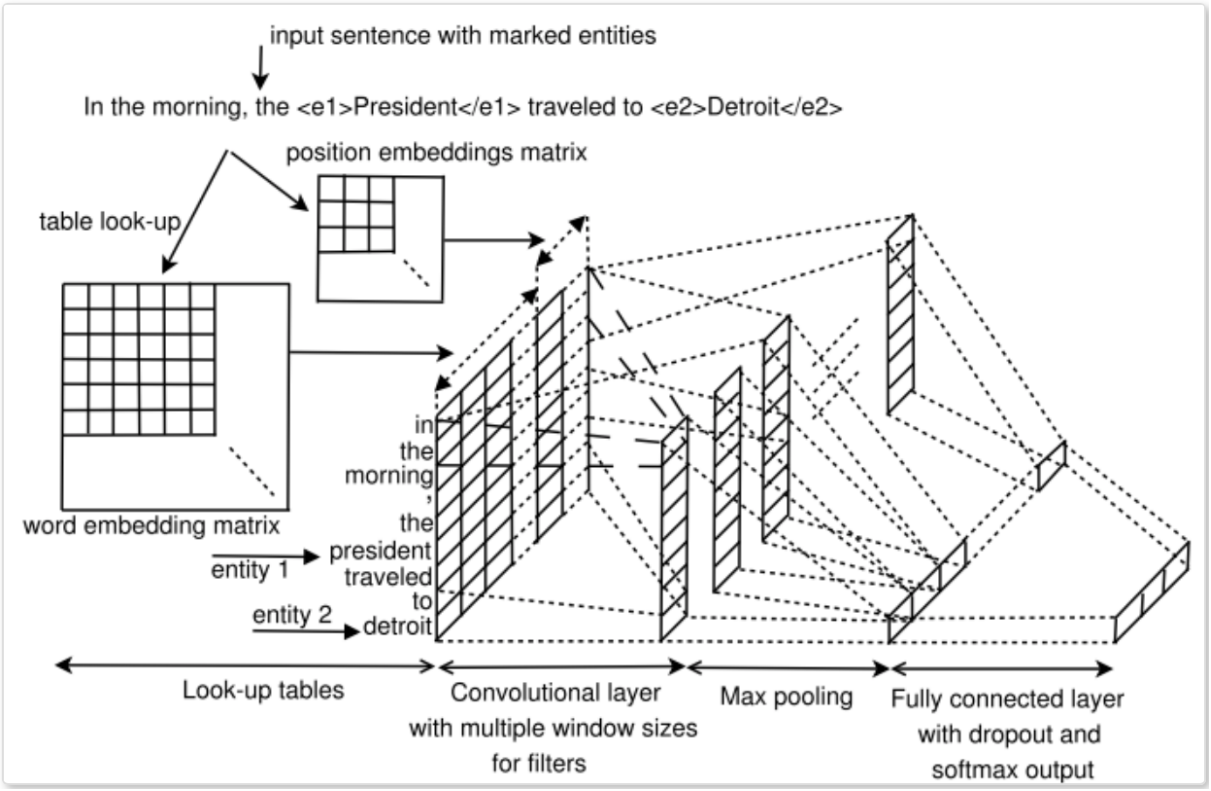

Figure 1: CNN architecture for relation extraction with multi-sized window [2]

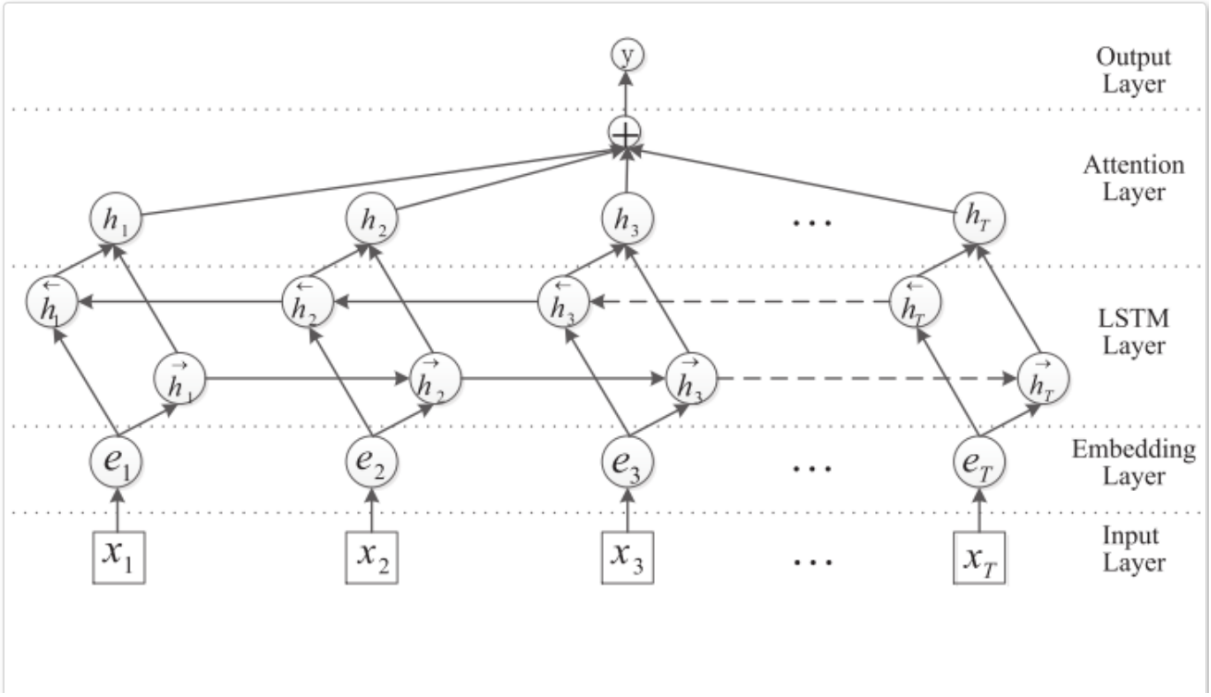

Figure 2: RNN architecture for relation extraction with attention [5]

Table 1: Performance of CNN model with various window-size

Window size	F1 score
3	80.75%
2, 3, 4, 5	82.01%
2, 3, 4, 5, 6, 7, 8	82.25%
2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	82.17%

$$r = H\alpha^T \tag{3}$$

where α is the attention weights.

2.3 Loss Function

There is one problem using raw cross entropy loss. The "Other" class is treated the same as other classes. This is problematic because "Other" class contains more redundant distribution than normal classes. In [1], the author proposes rank loss defined as

$$L = \log(1 + \exp(\gamma(m^+ - s_\theta(x)_{y^+}))) + \log(1 + \exp(\gamma(m^- + s_\theta(x)_{y^-}))) \tag{4}$$

where s is the final score for class $y+$ and $y-$. The negative label used is the one with maximum score. m is a margin to separate positive and negative scores. We treat "Other" labels separately. During training, we set $s_\theta(x)_{y^+}$ to be zero directly. During testing, if all the score are less than zero, we classify it as "Other".

3 Performance Analysis

During experiments, we randomly choose 800 examples for validation. All the models are trained in 100 epoch and pick the one with greatest validation F1 score.

3.1 Compare various window-size for CNN

We show the F1 score with various window-size in Table 1. By using multi-size window size, the F1 performance increases around 1.5%. However, keep increasing the window size will cause the model to overfitting. In the following experiments, we use window size of $[2, 3, 4, 5, 6, 7, 8]$.

3.2 The effect of rank loss for CNN-based model

We train the model with rank loss [1] and the multi-sized window is $[2, 3, 4, 5, 6, 7, 8]$. The final F1 score is 82.89%, which increases by 0.64%.

3.3 RNN model with attention

Finally, we train RNN model with attention. The validation F1 score is 82.25%, which is compared to CNN-based model. However, we didn't use lexical features in RNN-based model.

References

[1] C. N. dos Santos, B. Xiang, and B. Zhou. Classifying relations by ranking with convolutional neural networks. *CoRR*, abs/1504.06580, 2015.

[2] T. H. Nguyen and R. Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48. Association for Computational Linguistics, 2015.

[3] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

[4] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344. Dublin City University and Association for Computational Linguistics, 2014.

[5] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, and B. Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *ACL*, 2016.