

Add Name Entity Recognizer to Convolutional Neural Network

Guangyu Lin
UTEID: gl8429
glin@utexas.edu

Ge Gao
UTEID: gg24984
gegao1118@utexas.edu

Huihuang Zheng
UTEID: hz4674
huihuang@utexas.edu

Abstract

TODO

1. Introduction

TODO

2. Problem Definition and Algorithm

2.1. Task Definition

TODO

2.2. Algorithm Definition

TODO

3. Experiment

3.1. Methodology

3.1.1 Dataset

Our method was evaluated by comparing accuracy in a question classification task named TREC [6] with previous researchers' results. The task involves classifying a question into k question types. They defined a two-layered taxonomy, which represents a natural semantic classification for typical answers in the TREC task. The hierarchy contains 6 coarse classes (ABBREVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION and NUMERIC VALUE) and 50 fine classes. Each coarse class contains a non-overlapping set of fine classes. We choose this dataset because of two reasons. The fact that Name Entity Recognizer can classify person, location may help us in this task. Second, in Kim's paper [4] and Li's paper [6], they also used this database and reported previous accuracy so it will be easy for us to compare. The TREC dataset has 5500 annotated questions for training and 500 questions as the test set.

3.1.2 Setup

We used *word2vec* trained model from Mikolov [7] on 100 billion words of Google News and from TODO: Guangyu please add the reference for word2vec. We built our framework using NER from Stanford NER software [2] TODO: Guangyu, am I correct?

and deep learning tool Theano [1]. Since Kim's work[4] has Github published code using Theano (https://github.com/yoonkim/CNN_sentence), our method which combines CNN and NER was extended from their code.

We compared our method with Kim's method of using CNN and previous best methods without using CNN in the TREC dataset. By comparing CNN and our CNN + NER method, we can evaluate the contribution of NER features in this task. In addition, because Kim's method didn't outperform previous methods without CNN, we need to compare previous methods.

3.1.3 Evaluation

In this paper, we use same metric as previous researchers. We count the number of correctly classified questions by two standards. Top-1 accuracy and top-5 accuracy. We define

$$I_{i,j} = \begin{cases} 1 & \text{Correct label of the } i\text{-th question is} \\ & \text{output within rank } j \\ 0 & \text{otherwise} \end{cases}$$

Here, classifiers will give j labels for the i -th question. If the correct label for question i is among those j labels, $I_{i,j}$ will be set to 1. Top 1 accuracy is defined as $j = 1$, which is a usual precision definition. Top 5 accuracy is defined as $j = 5$, which allows multi-labels for classifiers. Formally, if we have n test samples, top- k accuracy is defined as

$$P_k = \frac{1}{n} \sum_{i=1}^n I_{i,k}$$

Method	Top-1 Accuracy	Top-5 Accuracy
Hierarchical classifier[5]	91.0	98.8
SVM _S [9]	95.0	-
CNN [4]	91.8	100.0
CNN + NER (ours)	91.8	100.0

Table 1. Top-1 and top-5 accuracy for TREC coarse question types.

Method	Top-1 accuracy	Top-5 accuracy
Hierarchical classifier[5]	84.20	95.00
SVM _S [9]	90.8	-
CNN [4]	80.2	90.60
CNN + NER (ours)	81.0	90.60

Table 2. Top-1 and top-5 accuracy for TREC coarse question types.

3.2. Experimental Results

We reimplemented Kim’s CNN method as baseline. We train Kim’s CNN and our CNN + NER in 2000 epochs. Then, we record the best accuracy in the 2000 epochs. We compared with two methods without using CNN: 1, Hierarchical classifier by Li and Roth[5] uses features unigram, POS, head chunks, NE, and semantic relations. 2, SVM_S uses ni-bi-trigrams, wh word, head word, POS, parser, hyponyms, and 60 hand-coded rules as features from Silva et al [9], which is the state-of-the-art in the best of our knowledge. Our test result for coarse classes was shown in table 1. Our test result for fine classes was shown in table 2

3.3. Discussion

In table 1, we can see our CNN + NER model doesn’t improve accuracy from Kim’s only CNN model. However, when it is in a more complexed situation, the fine task in table 2, we can see our model can slightly improved the top-1 accuracy (81.0 vs 80.2 in only CNN model).

Comparing to non-neural approaches methods, which needs manually engineered features. CNN’s approach is easier to train. Our method has higher top-1 and top-5 accuracy than the Hierarchical classifier method in coarse task. However in fine task, Li and Roth’s method outperforms ours. According to the fact that we add NER’s feature to CNN and improved the accuracy of fine task, we think that demonstrates the human defined features are helpful in fine types classification.

Our CNN + NER model has not got comparable accuracy to the work of Silva et al. [9]. However, their classifier involves a large number of hand-code resources. For example, they have 60 hand-coded rules. CNN’s training is simpler because we don’t need those manually engineered features.

4. Related Work

5. Future Work

This time we tried add NER features to CNN and improved CNN question classifier slightly, we think if we add more features, the CNN classifier can be more accurate.

6. Conclusion

In this work, we add Name Entity Recognizer (NER) as one dimension number to *word2vec* expression. The modified expression can be used for training a CNN for NLP tasks involving name entity. Our experiment shows that our CNN + NER model can improve the accuracy of question type classification task compared to only CNN method. Although our method still can not get comparable performance of the state-of-the-art work, the state-of-the-art work needs a large number of hand-code resources. Our method is easier to train.

References

- [1] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. J. Goodfellow, A. Bergeron, N. Bouchard, and Y. Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.
- [2] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005.
- [3] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*, 2014.
- [4] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.
- [5] X. Li and D. Roth. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.
- [6] X. Li and D. Roth. Learning question classifiers: the role of semantic information. *Natural Language Engineering*, 12(3):229–249, 2006.
- [7] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [8] D. Roth, G. K. Kao, X. Li, R. Nagarajan, V. Punyakanok, N. Rizzolo, W.-t. Yih, C. O. Alm, and L. G. Moran. Learning components for a question-answering system. In *TREC*, 2001.
- [9] J. Silva, L. Coheur, A. C. Mendes, and A. Wichert. From symbolic to sub-symbolic information in question classification. *Artificial Intelligence Review*, 35(2):137–154, 2011.
- [10] E. M. Voorhees and D. Tice. Overview of the trec-9 question answering track. In *TREC*, 2000.