Enriching Pre-trained Language Model with Entity Information for Relation Classification

Shanchan Wu
Alibaba Group (U.S.) Inc., Sunnyvale, CA
shanchan.wu@alibaba-inc.com

Yifan He
Alibaba Group (U.S.) Inc., Sunnyvale, CA
y.he@alibaba-inc.com

Abstract

Relation classification is an important NLP task to extract relations between entities. The state-of-the-art methods for relation classification are primarily based on Convolutional or Recurrent Neural Networks. Recently, the pre-trained BERT model achieves very successful results in many NLP classification / sequence labeling tasks. Relation classification differs from those tasks in that it relies on information of both the sentence and the two target entities. In this paper, we propose a model that both leverages the pre-trained BERT language model and incorporates information from the target entities to tackle the relation classification task. We locate the target entities and transfer the information through the pre-trained architecture and incorporate the corresponding encoding of the two entities. We achieve significant improvement over the state-of-the-art method on the SemEval-2010 task 8 relational dataset.

1 Introduction

The task of relation classification is to predict semantic relations between pairs of nominals. Given a sequence of text (usually a sentence) s and a pair of nominals e_1 and e_2, the objective is to identify the relation between e_1 and e_2 (Hendrickx et al., 2010). It is an important NLP task which is normally used as an intermediate step in variety of NLP applications. The following example shows the Component-Whole relation between the nominals "kitchen" and "house": "The [kitchen]$_{e1}$ is the last renovated part of the [house]$_{e1}$."

Recently, deep neural networks have applied to relation classification (Socher et al., 2012; Zeng et al., 2014; Yu et al., 2014; dos Santos et al., 2015; Shen and Huang, 2016; Lee et al., 2019). These methods usually use some features derived from lexical resources such as Word-Net or NLP tools such as dependency parsers and named entity recognizers (NER).

Language model pre-training has been shown to be effective for improving many natural language processing tasks (Dai and Le, 2015; Peters et al., 2017; Radford et al., 2018; Ruder and Howard, 2018; Devlin et al., 2018). The pretrained model BERT proposed by (Devlin et al., 2018) has especially significant impact. It has been applied to multiple NLP tasks and obtains new state-of-the-art results on eleven tasks. The tasks that BERT has been applied to are typically modeled as classification problems and sequence labeling problems. It has also been applied to the SQuAD question answering (Rajpurkar et al., 2016) problem, in which the objective is to find the starting point and ending point of an answer span.

As far as we know, the pretrained BERT model (Devlin et al., 2018) has not been applied to relation classification, which relies not only on the information of the whole sentence but also on the information of the specific target entities. In this paper, we apply the pretrained BERT model for relation classification. We insert special tokens before and after the target entities before feeding the text to BERT for fine-tuning, in order to identify the locations of the two target entities and transfer the information into the BERT model. We then locate the positions of the two target entities in the output embedding from BERT model. We use their embeddings as well as the sentence encoding (embedding of the special first token in the setting of BERT) as the input to a multi-layer neural network for classification. By this way, it captures both the semantics of the sentence and the two target entities to better fit the relation classification task.

Our contributions are as follows: (1) We put forward an innovative approach to incorporate

entity-level information into the pretrained language model for relation classification. (2) We achieve the new state-of-the-art for the relation classification task.

2 Related Work

There has been some work with deep learning methods for relation classification, such as (Socher et al., 2012; Zeng et al., 2014; Yu et al., 2014; dos Santos et al., 2015)

MVRNN model (Socher et al., 2012) applies a recursive neural network (RNN) to relation classification. They assign a matrix-vector representation to every node in a parse tree and compute the representation for the complete sentence from bottom up according to the syntactic structure of the parse tree. (Zeng et al., 2014) propose a CNN model by incorporating both word embeddings and position features as input. Then they concatenate lexical features and the output from CNN into a single vector and feed them into a softmax layer for prediction. (Yu et al., 2014) propose a Factor-based Compositional Embedding Model (FCM) by constructing sentence-level and substructure embeddings from word embeddings, through dependency trees and named entities. (Santos et al., 2015) tackle the relation classification task by ranking with a convolutional neural network named CR-CNN. Their loss function is based on pairwise ranking. In our work, we take advantage of a pre-trained language model for the relation classification task, without relying on CNN or RNN architecutures. (Shen and Huang, 2016) utilize a CNN encoder in conjunction with a sentence representation that weights the words by attention between the target entities and the words in the sentence to perform relation classification. (Wang et al., 2016) propose a convolutional neural network architecture with two levels of attention in order to catch the patterns in heterogeneous contexts to classify relations. (Lee et al., 2019) develop an end-to-end recurrent neural model which incorporates an entity-aware attention mechanism with a latent entity typing for relation classification.

There are some related work on the relation extraction based on distant supervision, for example, (Mintz et al., 2009; Hoffmann et al., 2011; Lin et al., 2016; Ji et al., 2017; Wu et al., 2019). The difference between relation classification on regular data and on distantly supervised data is that the latter may contain a large number of noisy labels. In this paper, we focus on the regular relation classification problem, without noisy labels.

3 Methodology

3.1 Pre-trained Model BERT

The pre-trained BERT model (Devlin et al., 2018) is a multi-layer bidirectional Transformer encoder (Vaswani et al., 2017).

The design of input representation of BERT is to be able to represent both a single text sentence and a pair of text sentences in one token sequence. The input representation of each token is constructed by the summation of the corresponding token, segment and position embeddings.

'[CLS]' is appended to the beginning of each sequence as the first token of the sequence. The final hidden state from the Transformer output corresponding to the first token is used as the sentence representation for classification tasks. In case there are two sentences in a task, '[SEP]' is used to separate the two sentences.

BERT pre-trains the model parameters by using a pre-training objective: the masked language model (MLM), which randomly masks some of the tokens from the input, and set the optimization objective to predict the original vocabulary id of the masked word according to its context. Unlike left-to-right language model pre-training, the MLM objective can help a state output to utilize both the left and the right context, which allows a pre-training system to apply a deep bidirectional Transformer. Besides the masked language model, BERT also trains a "next sentence prediction" task that jointly pre-trains text-pair representations.

3.2 Model Architecture

Figure 1 shows the architecture of our approach.

For a sentence s with two target entities e_1 and e_2, to make the BERT module capture the location information of the two entities, at both the beginning and end of the first entity, we insert a special token '$', and at both the beginning and end of the second entity, we insert a special token '#'. We also add '[CLS]' to the beginning of each sentence.

For example, after insertion of the special separate tokens, for a sentence with target entities "kitchen" and "house" will become to:

"[CLS] The $ kitchen $ is the last renovated part of the # house # . "

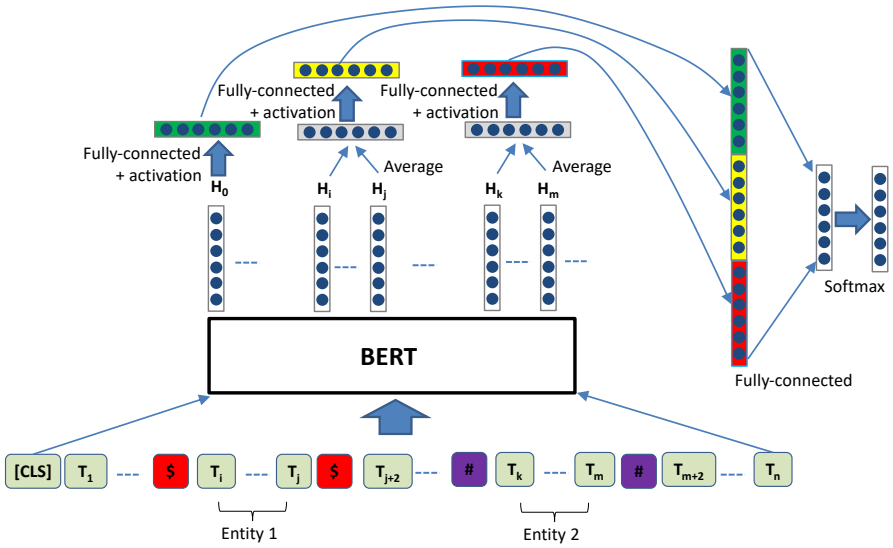

Figure 1: The model architecture.

Given a sentence s with entity e_1 and e_2, suppose its final hidden state output from BERT module is H. Suppose vectors H_i to H_j are the final hidden state vectors from BERT for entity e_1, and H_k to H_m are the final hidden state vectors from BERT for entity e_2. We apply the average operation to get a vector representation for each of the two target entities. Then after an activation operation (i.e. tanh), we add a fully connected layer to each of the two vectors, and the output for e_1 and e_2 are H_1' and H_2' respectively. This process can be mathematically formalized as Equation (1).

$$H_1' = W_1 \left[tanh \left(\frac{1}{j-i+1} \sum_{t=i}^{j} H_t \right) \right] + b_1$$
$$H_2' = W_2 \left[tanh \left(\frac{1}{m-k+1} \sum_{t=k}^{m} H_t \right) \right] + b_2 \quad (1)$$

We make W_1 and W_2, b_1 and b_2 share the same parameters. In other words, we set $W_1 = W_2$, $b_1 = b_2$. For the final hidden state vector of the first token (i.e. '[CLS]'), we also add an activation operation and a fully connected layer, which is formally expressed as:

$$H_0' = W_0 \left(tanh(H_0) \right) + b_0 \quad (2)$$

Matrices W_0, W_1, W_2 have the same dimensions, i.e. $W_0 \in R^{d \times d}$, $W_1 \in R^{d \times d}$, $W_2 \in R^{d \times d}$, where d is the hidden state size from BERT.

We concatenate H_0', H_1', H_2' and then add a fully connected layer and a softmax layer, which can be expressed as following:

$$h'' = W_3 \left[concat \left(H_0', H_1', H_2' \right) \right] + b_3$$
$$p = softmax(h'') \quad (3)$$

where $W_3 \in R^{L \times 3d}$ (L is the number of relation types), and p is the probability output. In Equations (1),(2),(3), b_0, b_1, b_2, b_3 are bias vectors.

We use cross entropy as the loss function. We apply dropout before each fully connected layer during training. We call our approach as R-BERT.

4 Experiments

4.1 Dataset and Evaluation Metric

We use the SemEval-2010 Task 8 dataset in our experiments. The dataset contains nine semantic relation types and one artificial relation type *Other*, which means that the relation does not belong to any of the nine relation types. The nine relation types are *Cause-Effect, Component-Whole, Content-Container, Entity-Destination, Entity-Origin, Instrument-Agency, Member-Collection, Message-Topic* and *Product-Producer*. The dataset contains 10,717 sentences, with each containing two nominals e1 and e2, and the corresponding relation type in the sentence. The relation is directional, which means that Component-Whole(e1, e2) is different from Component-Whole(e2, e1). The dataset has already been partitioned into 8,000 training instances and 2,717 test instances. We evaluate our

solution by using the SemEval-2010 Task 8 official scorer script. It computes the macro-averaged F1-scores for the nine actual relations (excluding Other) and considers directionality.

4.2 Parameter Settings

Table shows the major parameters used in our experiments.

Table 1: Parameter settings.

Batch size	16
Max sentence length	128
Adam learning rate	2e-5
Number of epochs	5
Dropout rate	0.1

We add dropout before each add-on layer. For the pre-trained BERT model, we use the uncased basic model. For the parameters of the pre-trained BERT model, please refer to (Devlin et al., 2018) for details.

4.3 Comparison with other Methods

We compare our method, R-BERT, against results by multiple methods recently published for the SemEval-2010 Task 8 dataset, including SVM, RNN, MVRNN, CNN+Softmax, FCM, CR-CNN, Attention-CNN, Entity Attention Bi-LSTM. The SVM method by (Rink and Harabagiu, 2010) uses a rich feature set in a traditional way, which was the best result during the SemEval-2010 task 8 competition. Details of all other methods are briefly reviewed in Section 2.

Table 2 reports the results. We can see that R-BERT significantly beats all the baseline methods. The MACRO F1 value of R-BERT is 89.25, which is much better than the previous best solution on this dataset.

Table 2: Comparison with results in the literature.

Method	F1
SVM (Rink and Harabagiu, 2010)	82.2
RNN (Socher et al., 2012)	77.6
MVRNN (Socher et al., 2012)	82.4
CNN+Softmax (Zeng et al., 2014)	82.7
FCM (Yu et al., 2014)	83.0
CR-CNN (Santos et al., 2015)	84.1
Attention CNN (Shen and Huang, 2016)	85.9
Att-Pooling-CNN (Wang et al., 2016)	88.0
Entity Attention Bi-LSTM (Lee et al., 2019)	85.2
R-BERT	**89.25**

4.4 Ablation Studies

4.4.1 Effect of Model Components

We have demonstrated the strong empirical results based on the proposed approach. We further want to understand the specific contributions by the components besides the pre-trained BERT component. For this purpose, we create three more configurations.

The first configuration is to discard the special separate tokens (i.e. '$' and '#') around the two entities in the sentence and discard the hidden vector output of the two entities from concatenating with the hidden vector output of the sentence. In other words, we add '[CLS]' at the beginning of the sentence and feed the sentence with the two entities into the BERT module, and use the first output vector for classification. We label this method as **BERT-NO-SEP-NO-ENT**.

The second configuration is to discard the special separate tokens (i.e. '$' and '#') around the two entities in the sentence, but keep the hidden vector output of the two entities in concatenation for classification. We label this method as **BERT-NO-SEP**.

The third configuration is to discard the hidden vector output of the two entities from concatenation for classification, but keep the special separate tokens. We label this method as **BERT-NO-ENT**.

Table 3 reports the results of the ablation study with the above three configurations. We observe that the three methods all perform worse than R-BERT. Of the methods, BERT-NO-SEP-NO-ENT performs worst, with its F1 8.16 absolute points worse than R-BERT. This ablation study demonstrates that both the special separate tokens and the hidden entity vectors make important contributions to our approach.

In relation classification, the relation label is de-

pendent on both the semantics of the sentence and the two target entities. BERT without special separate tokens cannot locate the target entities and lose this key information. The reason why the special separate tokens help to improve the accuracy is that they identify the locations of the two target entities and transfer the information into the BERT model, which make the BERT output contain the location information of the two entities. On the other hand, incorporating the output of the target entity vectors further enriches the information and helps to make more accurate prediction.

Table 3: Comparison of the BERT based methods with different components.

Method	F1
R-BERT-NO-SEP-NO-ENT	81.09
R-BERT-NO-SEP	87.98
R-BERT-NO-ENT	87.99
R-BERT	**89.25**

5 Conclusions

In this paper, we develop an approach for relation classification by enriching the pre-trained BERT model with entity information. We add special separate tokens to each target entity pair and utilize the sentence vector as well as target entity representations for classification. We conduct experiments on the SemEval-2010 benchmark dataset and our results significantly outperform the state-of-the-art methods. One possible future work is to extend the model to apply to distant supervision.

References

Andrew M Dai and Quoc V Le. 2015. Semi-supervised Sequence Learning. In *Advances in Neural Information Processing Systems 28*. 3079–3087.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805

Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying Relations by Ranking with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*. 626–634.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó. Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 Task 8: Multi-way Classification of Semantic Relations Between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation (SemEval '10)*. 33–38.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke S. Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations. In *The 49th Annual Meeting of the Association for Computational Linguistics, ACL 2011*. 541–550.

Guoliang Ji, Kang Liu, Shizhu He, and Jun Zhao. 2017. Distant Supervision for Relation Extraction with Sentence-Level Attention and Entity Descriptions. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017*. 3060–3066.

Joohong Lee, Sangwoo Seo, and Yong Suk Choi. 2019. Semantic Relation Classification via Bidirectional LSTM Networks with Entity-aware Attention using Latent Entity Typing. *CoRR* (2019).

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural Relation Extraction with Selective Attention over Instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant Supervision for Relation Extraction Without Labeled Data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL '09)*. 1003–1011.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *CoRR* abs/1705.00108 (2017). arXiv:1705.00108

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving lan-

guage understanding with unsupervised learning. *Technical report, OpenAI* (2018).

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2383–2392.

Bryan Rink and Sanda Harabagiu. 2010. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. 256–259.

Sebastian Ruder and Jeremy Howard. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. 328–339.

Cicero Nogueira Dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying Relations by Ranking with Convolutional Neural Networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, (ACL) 2015*.

Yatian Shen and Xuanjing Huang. 2016. Attention-based Convolutional Neural Network for Semantic Relation Extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 2526–2536.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. Semantic Compositionality Through Recursive Matrix-vector Spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL '12)*. Stroudsburg, PA, USA, 1201–1211.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). 5998–6008.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation Classification via Multi-Level Attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Shanchan Wu, Fan Kai, and Qiong Zhang. 2019. Improving Distantly Supervised Relation Extraction with Neural Noise Converter and Conditional Optimal Selector. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, 2019*.

Mo Yu, Matthew R. Gormley, and Mark Dredze. 2014. Factor-based compositional embedding models. In *In NIPS Workshop on Learning Semantics*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification via Convolutional Deep Neural Network. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, 2014*. 2335–2344.