RG_PA at SemEval-2021 Task 1: A Contextual Attention-based Model with RoBERTa for Lexical Complexity Prediction

Gang Rao[*], **Maochang Li**[*], **Xiaolong Hou, Lianxin Jiang, Yang Mo, Jianping Shen**
Ping An Life Insurance, Lt
ShenZhen, China
{raogang532, limaochang389, houxiaolong430, jianglianxin769,
moyang853, shenjianping324}@pingan.com.cn

Abstract

In this paper we propose a contextual attention-based model with two-stage fine-tune training using RoBERTa. First, we perform the first-stage fine-tune on corpus with RoBERTa, so that the model can learn some prior domain knowledge. Then we get the contextual embedding of context words based on the token-level embedding with the fine-tuned model. And we use Kfold cross-validation to get K models and ensemble them to get the final result. Finally, we attain the 2nd place in the final evaluation phase of sub-task 2 with pearson correlation of 0.8575.

1 Introduction

LCP is an augmented version of Complex Word Identification(CWI) (Shardlow et al., 2020), predict complexity score for each target word in a sentence. The dataset is a multi-domain English dataset annotated with a 5-point Likert scale (1-5). The annotation model in CompLex addresses complexity as a continuum instead of a binary feature. Previous studies of CWI treat the task as a binary classification, predict a complexity label (complex vs. non-complex) for a set of target words in a sentence.

In this paper, we exploratory data analysis(EDA) for the dataset, and found that the distribution of the single task and multi task dataset are very inconformity, so should bulid two models for every dataset great than one model for merge two task dataset.

Several key technologies as follows:

- Train a RoBERTa based fine-tune corpus classifier. It use the data of all single and multi with train, trial and test dataset as train data, no dev and test, and only train 1 epoch, which enable the RoBERTa model ahead of time learning the domain knowledge.
- At each layer, calculate target vector and context tokens embedding attention, the layer context vector is average the context tokens embedding with soft alignment.
- Weighted the RoBERTa last 12 layers context vector and target vector. They use the same weights, and it's sum equals to 1.
- The degeneration of gradual unfreezing (Howard and Ruder, 2018). At first epoch freeze the pretrained model parameter only learning the head layers parameter, then unfreeze all model parameter.
- Multi-Sample Dropout at last layer (Inoue, 2019)

2 Background

Previous approaches to CWI typically refer to binary identification of complex words, two shared tasks on CWI topic have been organized so far. SemEval-2016 Task 11 (Paetzold and Specia, 2016) and BEA workshop 2018 (Yimam et al., 2018). The two tasks approache a number of different model to classification, ranging from traditional machine learning classifiers such as support vector machines (SVM), decision trees, random forest, and maximum entropy classifiers to deep learning classifiers, such as recurrent neural networks. A wide range of features were used such as word embeddings, word and character n-grams, word frequency, Zipfian frequency distribution, word length, morphological, syntactic, semantic, and psycholinguistic.

[*] These authors contributed equally to this work This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http:// creativecommons.org/licenses/by/4.0/

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 623–626
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

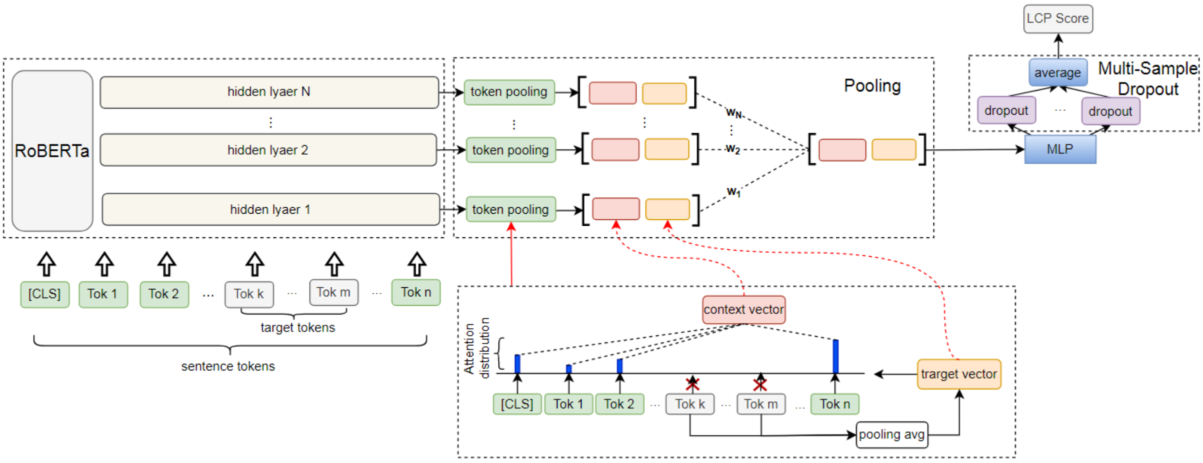

Figure 1: Attention Based Context Representation for LCP

BERT is a new language representation model, and stands for Bidirectional Encoder Representations from Transformers (Devlin et al., 2018). Since BERT appear, fine-tuned pre-trained model with just one additional output layer to create state-of-the-art models for a wide range of tasks. A number of Transformers series models are proposed, such as GPT-2, RoBERTa (Liu et al., 2019), XLM, DistilBert, XLNet. In this papaer we focus on use RoBERTa to slove the tasks.

3 System overview

We propose a RoBERTa with attention based model to solve the LCP task, Figure 1 outlines our proposed model framework. First, use Byte-Pair Encoding (BPE) to tokenize the input sentent, which is an effective subword technique to relieve the Out-of-Vocabulary (OOV) problem. For every RoBERTa hidden layer, we apply token pooling that is average the target words tokens embedding as the target vector. Then we calculate the attention between target vector and context tokens vector which are sentence tokens masked the target tokens, After that, context tokens embedding multiply attention weight as the context vector, then concatenate the context vector and target vector. Second, pooling the RoBERTa last 12 layers context vector and target vector, Finally, connect the MLP layer to predict the LCP Score.

3.1 Pooling

The input sentenct is tokenized to n tokens $t_i, i = 1, 2, ..., n$, and the target tokens index are $l_t = \{k, k+1, ..., m\}$, the context tokens index are $l_c = \{1, ..., k-1, m+1, ..., n\}$, which exclude the target tokens. E_i^j denotes the i_{th} token embedding of hidden layer j, T^j denotes the target vector of the hidden layer j.

$$T^j = \frac{1}{m-k+1} \sum_{i \in l_t} E_i^j \tag{1}$$

The attention weight between context tokens embdding and target vector of layer j is compute by.

$$\alpha_i^j = softmax(s(E_i^j, T^j)) = \frac{exp(s_i^j)}{\sum_i exp(s_i^j)}, i \in l_c \tag{2}$$

where

$$s_i^j = s(E_i^j, T^j) = (E_i^j)^{\mathrm{T}} T^j \tag{3}$$

After that, we compute the weighted summation for c^j

$$c^j = \sum_{i \in l_c} \alpha_i^j E_i^j \tag{4}$$

Finally, calculate the pooling target vetor $\bar{t}$, and context vector $\bar{c}$, they are weighted last N layers target vector and context vector of RoBERTa, the weight $w_1, w_2, ..., w_N$ is the model parameters and equals to 1.

$$\bar{t} = \sum_j^N w_j t^j \tag{5}$$

$$\bar{c} = \sum_j^N w_j c^j \tag{6}$$

$$\sum_{j=1}^N w_j = 1 \tag{7}$$

Quantile	Single		Multi	
	train	trial	train	trial
Q1	0.2115	0.2143	0.3026	0.3090
Q2	0.2794	0.2667	0.4091	0.4219
Q3	0.3750	0.3594	0.5294	0.5140

Table 1: Single and Multi DataSet Quantile

4 Experiments

4.1 Data

Table 1 shows the single and multi dataset quantile. From the table we found that the single and multi words Complexit Score distribution are very different. Single is easy to understand but multi words are difficult. So we build different model to adjust the dataset.

We use 5-fold cross validation, first generate a new feature score_bin which is binning the LCP score by quantile. Because the dev dataset commonly used to search the optimal hyper parameters, in this experiment we only use dev dataset to found the best epoch, in order to prevent overfitting by early stop, but we found that pretrained model only train 5-6 epochs could be convergence on the task dataset, so not need deliberately generate the dev dataset, only let dev dataset as same as test dataset. Train and test dataset are splited use stratified KFold by the features domain corpus and score_bin.

4.2 Corpus information

The dataset give the sentences domain corpus, but how to use this information? At first, we build the multi-task learning. The auxiliary task is the corpus classification which use the last 12 layers average CLS token embedding. But the auxiliary task not improve the LCP task, and the accuracy of corpus classifier is quite low. It's not conform to the actual, because of the sentences corpus come from Bible Europarl and Biomedical, and they are very easy to distinguish.

Since that, we build a corpus classification model separately which is a RoBERTa fine-tune model (Sun et al., 2019). Benefit from the dataset are easy to classify, the model only need to train 1 epoch, and could get 0.99 accuracy. We merge the single and multi trial, train and test dataset as new train dataset, this can let the model see all data include test dataset. After train, the RoBERTa learning the domain knowledge, and in advance learning part of the test dataset.

Then, export the RoBERTa model as the pre-trained model of LCP task.

4.3 Single LCP Task

First merge the single train and trial dataset, then process stratified 5-fold, compare the origin pre-trained model(RoBERTa-large) and fine-tune by the corpus classification(pre-RoBERTa-large).

For train, we use the Mean squared error(MSE) loss function and adam optimizer (Kingma and Ba, 2014). At first epoch we freeze the RoBERTa parameters, only traininge the head layers. Apply learning rate linear schedule with warmup, $lr = 2e-5$, $warmup\_steps = 200$, and use early stop.

Table 2 shows the single task result, the metric is Pearson correlation (R). The **fold-x** column is the metric of CV model evaluate on the fold-x dataset. The **mean** column is the average of the fold-* column. Pre-trained corpus classification with fine-tune RoBERTa-large is a little outperformance than origin RoBERTa-large. The single model result is the average of all 5-fold models's predict result for single task test data, and **model result** column is the metric of the the model result. The task final result is the average of all models result, and **final result** column is the metric of the the final result. The two model can achieve 0.7586 and 0.7618, but use simple average ensemble could get 0.7629. It's quite effective.

4.4 Multi LCP Task

The merged dataset of multi train and trial only have 1616 examples, In single task, the pre-RoBERTa-large is outperformance than origin RoBERTa-large. In order to augment the multi task examples, Fisrt use the data which merge all sigle and multi train trial dataset, use 5-fold cross validation, splited data use stratified KFold as same as single task. Then use pre-RoBERTa-large train the LCP task. After that, inference the vector $\overline{h} = [\overline{c}, \overline{t}]$ for all merge data, the final $\overline{h}$ is average of all 5-fold models. Finally, use the vector to calculate cosine similarity of the multi dataset with single dataset, then recall single examples add to the multi train example with threshold. Here we use $sim\_threshold = 0.75$, and recall 2707 single examples.

Then split dataset and train strategy are as same as singe task. The results are in Table 3. gen-RoBERTa-large is the origin RoBERTa model with Data Augmentation, pre-gen-RoBERTa-large is the RoBERTa model fine-tune by the corpus with Data

model	the result of 5-fold cross validateion						LCP single task	
	fold-1	fold-2	fold-3	fold-4	fold-5	mean	model result	**final result**
RoBERTa-large	0.7528	0.7723	0.7777	0.7854	0.7696	0.7716	0.7586	**0.7629**
pre-RoBERTa-large	0.7636	0.7725	0.7707	0.7849	0.7719	0.7727	0.7618	

Table 2: Single Task Result

model	the result of 5-fold cross validateion						LCP multi task	
	fold-1	fold-2	fold-3	fold-4	fold-5	mean	model result	**final result**
RoBERTa-large	0.7531	0.7681	0.7829	0.7692	0.7492	0.7645	0.8310	**0.8575**
pre-RoBERTa-large	**0.7862**	**0.7952**	0.7624	0.7352	0.7656	0.7689	0.8332	
gen-RoBERTa-large	0.7713	0.7517	**0.7845**	0.7550	**0.7793**	0.7684	0.8325	
pre-gen-RoBERTa-large	0.7614	0.7646	0.7616	**0.7920**	0.7791	**0.7717**	0.8355	

Table 3: Multi Task Result

Augmentation. Results shows model fine-tune by the corpus classification are outperformance than origin model, The final result 0.8575 is fusioned by average of the four models cv results, and rank the 2nd in test phrase.

5 Conclusion

This paper presents a method to predicting lexical complexity, which apply RoBERTa-large as the backbone language model. First fine-tune backbone model for corpus classification. Then bulid model with attention based context representation. make vector recall for multi task data augmentation. Finally, we carry out a multi-model average ensemble strategy to enhance the model performance. In the future, we will exploit better model for text representation, and utilizing data augmentation for all task.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Hiroshi Inoue. 2019. Multi-sample dropout for accelerated training and better generalization. *arXiv preprint arXiv:1905.09788*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Gustavo Paetzold and Lucia Specia. 2016. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. Complex: A new corpus for lexical complexity predicition from likert scale data. *arXiv preprint arXiv:2003.07008*.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo H Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. *In Proceedings of BEA*.