Empirical Studies for Answering KorQuAD-OPEN

Jinyoung Sung
School of Computing
KAIST
jysung710@kaist.ac.kr

Jaewon Lee
School of Computing
KAIST
jwlee3746@kaist.ac.kr

Jaejin Lee
School of Computing
KAIST
fovjfozk@gmail.com

Abstract

Machine question answering is a challenging task in natural language processing. Some pre-trained models such as BERT gained attention due to their great performance recently. In this project, we were provided with KorQuAD-OPEN, a challenging dataset for Korean question answering, and baseline model BERT. Through several trials and errors, we have succeeded in tuning the pre-trained model for this task. We show our results of these experiments, and analyze the results for further research of Korean question answering.

1 Introduction

Machine question answering is one of the oldest, yet still challenging task in natural language processing task. The goal of the project is to produce a model to predict the answers according to the given question. For the project, we mainly focus on **finding a perfect match for the KorQuAD-OPEN dataset and fine tuning the model**. Recently, networks of question answering models tends to become larger to achieve state-of-the-art performance. Our best model has achieved F1 score of 76 on the dev set, and 45.48 on the test set.

2 Related Works

Pre-trained Language Model Bidirectional Encoder Representations from Transformers (BERT)(Devlin et al., 2018) model released by Google showed great improvement in question answering task. Although it has been shown to be powerful in SQuAD(Rajpurkar et al., 2016) challenge, we could not improve the performance of BERT any further on the Korean question answering dataset(Lim et al., 2019).

There are 4 publicly available pre-trained models for Korean dataset; KoBERT, KoELECTRA, HanBERT, and KorBERT. KorBERT from ETRI is said to show good overall performance for this task, but we could not test this model due to model size and license issue. We try to maximize the performance on the base of KoBERT and KoELECTRA for this project.

3 Dataset

Standford Question Answering Dataset(SQuAD) is a representative reading comprehension dataset for question answering task. Similarly KorQuAD, which is built by LG CNS, consists of context, question and answer. KorQuAD-OPEN modifies KorQuAD by using retrieved snippets from Naver instead of using Wikipedia. There are 7954 questions for train set and 66367 questions for dev set and approximately 65% of the dataset is answerable.

4 Approach

In this section, we discuss our approach for Korean questioning answering task or we plan to implement. We use BERT as our baseline model, which is provided by NAVER corp., and modify the KoELECTRA pre-trained model from HuggingFace.

4.1 Baseline

BERT-Base, Multilingual Cased pre-trained model which has 102 languages, 12-layer, 768-hidden, 12-heads, 110M parameters is used.

4.2 Our Model

ELECTRA(Clark et al., 2020) learns by looking at the token from the generator to determine whether it is ”real” token or ”fake” token in the discriminator (Replaced Token Detection). This method has the advantage of being able to learn about all input token, and has shown better performance compared to BERT. Learning Appearance is shown in Figure 1.

KoELECTRA is an ELECTRA model that has been pre-trained with 14GB of Korean text (96M texts, 2.6B tokens). **KoELECTRA-Base**

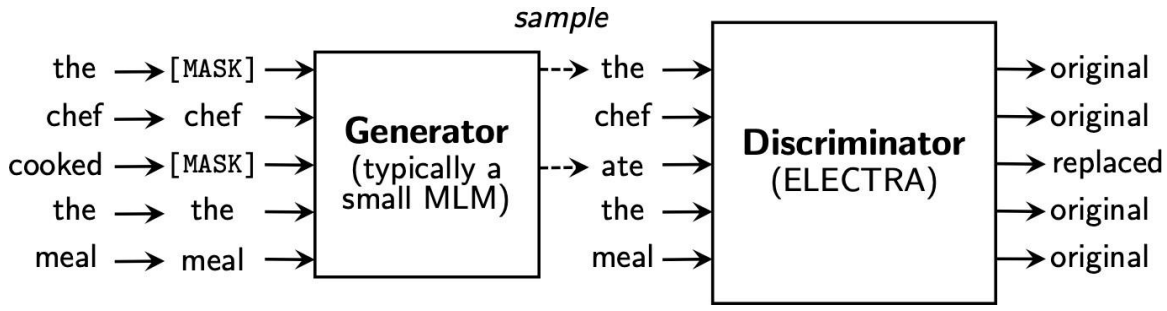

Figure 1: Learning Appearance of ELECTRA

was used as the main model, and the smaller model **KoELECTRA-Small** was used as the ensemble model.

4.3 Ensemble

For the most of the experiments, we have found that models showed the best test score after running 1 or 2 epoch, while **F1** is still increasing. It seems doubtless that models are overfitting to the train set, since the dataset is relatively small. The best way to overcome this problem is adding more Korean comprehension data. However, this project was limited to using this specific dataset, thus we seek to improve generalization performance by combining smaller models. First ensemble model is to add logits of two different small models, **KoELECTRA-small-v2** and **DistilBERT**. Another strategy is to get votes from several **KoELECTRA-small-v2** models. We expect the models to jointly learn and reach to a better generalization result.

5 Experiments

5.1 Evaluation method

We use **F1**-score as a performance measurement. **F1** is a function of Precision and Recall, which is

$$\textbf{F1} = \frac{2 \times precision \times recall}{precision + recall}$$

in formula. We use character-level **F1** since we cannot apply the bags of tokens for Korean language. **EM** metric explains how much of the predictions matches the ground truch answers exactly. We focus on **F1**-score, since it is a balanced measurement between Precision and Recall, and it is hard to predict the exact answer in this task.

5.2 Normalizing text

We test if quality of the text corpus affects the performance of pre-trained model. Training code from the baseline model regard the name '¾Kø' and the name with word spacing '¾ Kø' different. Also there were Chinese letters in the data, dropping the training efficiency. We try to ignore these spacing, and unnecessary letters by normalizing the text during training.

5.3 Hyperparameter tuning

We try to find the best model by tuning the hyperparameters, since there was a lot of volatility in **F1**-score during training. We study if lowering learning rate or increasing batch size have effects on performance.

Batch size While Masters and Luschi (2018) and Keskar et al. (2016) claims reducing batch size might have positive effects on generalization performance, we test bigger batch size since the model with small batch size takes too much time to converge. We test $batch\_size = 6, 8, 16$.

Learning rate We optimize learning rate to stabilize the training process. We test $learning\_rate = 2e-5, 5e-6$.

6 Results

6.1 Pre-trained model

Model	Dev	Test
Bert-multilingual (Baseline)	74.4	36.0
KoELECTRA-small-v2	73	36.5
KoELECTRA-base-v2	77.8	36.3
KoELECTRA-base-v2 (KorQuAD)	74.6	**41.9**

Table 1: **F1**-score of pre-trained models. KoELECTRA-base-v2 model pre-trained on KorQuAD dataset showed the best test score.

According to Table 1, KoELECTRA model performs better than BERT model. Analysis of the

reasons why KoELECTRA outperformed Bert-multilingual-cased shows the following.

First, KoELECTRA learned 14GB of Korean text (96M texts, 2.6B tokens) which is a larger amount of Korean text data than Bert-multilingual-cased. Also, while Bert only contained Wikipedia data, KoELECTRA was more optimized and learned in KorQuAD task because it included well-preprocessed news data, crawled data from portals and also Wikipedia data.

Also, there is a difference in pre-processing. Unlike Bert, KoELECTRA normalized text by eliminating Chinese characters and some special characters, used the Korean Sentence Separator (KSS), and removed noise sentence such as author name, date, and head of articles.

6.2 Normalizing text

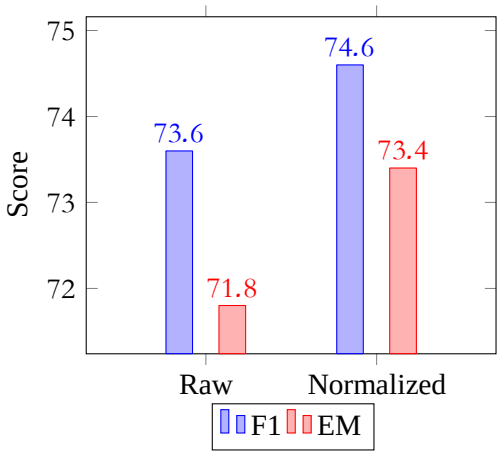

Figure 2: Normalization in text yields better performance. This implies that more amount of qualified data is required for this task.

Figure 2 shows the **F1**-score before and after normalizing the text. After normalizing the text, we could achieve the best performance with KoELECTRA. It shows the quality of the dataset has impact on question answering model. Thus refining the dataset thoroughly by removing Chinese letters, some watermarks, or any other unnecessary parts instead of simply ignoring letters might show even better results.

6.3 Hyperparameter tuning

Learning rate (*batch size* = 6)	**Dev**	**Test**
$lr = 2e-5$	74.4	41.9
$lr = 5e-6$	73.5	**45.4**
$lr = 2e-6$	77.1	36.8
Batch size ($lr = 5e-6$)	**Dev**	**Test**
$batch\_size = 6$	73.3	45.2
$batch\_size = 8$	77.1	36.8
$batch\_size = 16$	72.9	**45.4**

Table 2: **F1** result of hyperparameter tuning.

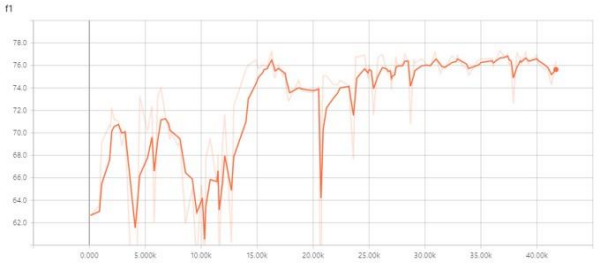

Figure 3: Learning progress **before** hyperparameter tuning

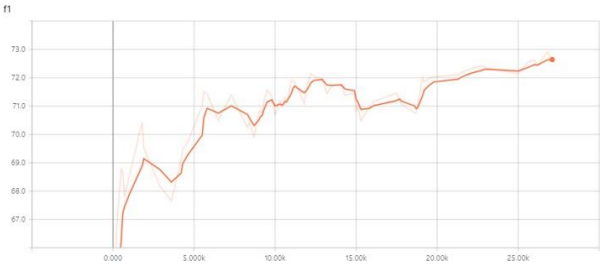

Figure 4: Learning progress **after** hyperparameter tuning

See the figure 3. In order to reduce divergent behaviors, the learning rate was reduced to the extent that it did not reduce the efficiency of learning. Also, the batch size was increased to the extent that no out of memory problems occurred so that learning could be done quickly and stably. According to the Table 2, it seems that the model with $batch\_size = 16$ and $learning\_rate = 5e-6$ is the optimal parameter. There are significant improvements of test F1-score and stabilized learning progress in the figure 4 at the optimal parameter.

By combining the best approaches discussed in previous sections, our model achieves **45.48** in test score.

6.4 Ensemble

We plan for applying ensemble methods to improve generalization performance by combining smaller models. First ensemble model is to mean logits of two different small models, **KoELECTRA-small-v2** and **DistilBERT**. Another strategy is to get votes from several **KoELECTRA-small-v2** models. We write the code and tried to run the model in NSML, but failed to confirm the result due to problems with GPU resources. We will try to apply this method in further research.

7 Conclusion

In this project, we have focused on searching a perfect model for KorQuAD-OPEN dataset. Our empirical studies in Section 6 show that **KoELECTRA** is a better model than **BERT** for Korean question answering, and the Korean dataset should be more refined and larger. Our model achieved **F1 score of 77.8 on Dev Set and 45.48 on the test set**. That was the 8th position among 16 teams on leaderboard.

For future work, we could consider pre-training the data or utilizing more Korean comprehension data from KorQuAD 1.0 to prevent overfitting and improve overall performance. We could also perform ensemble learning with smaller models to accelerate training process achieve better generalization performance.

References

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. 2016. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. Korquad1.0: Korean qa dataset for machine reading comprehension.

Dominic Masters and Carlo Luschi. 2018. Revisiting small batch training for deep neural networks. *CoRR*, abs/1804.07612.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250.