

---

# Practice of Improvement KoChatGPT by preprocessing and data augmentation

---

Donggyu Kim  
Aiffel  
Seoul, South Korea  
crlotwhite@gmail.com

## Abstract

*ChatGPT* brings unprecedented changes, but there are still many shortcomings in Korean. One of the Korean language models, *KoChatGPT*, is more specialized model for Korean than *ChatGPT* and it is open source. However, it is not known whether the model is actually the same as *ChatGPT*, and it is not showing good results because the data is not sufficiently refined. As a researcher studying deep learning and natural language processing, I conducted a small project to improve the model to improve it and contribute to the development of Korean natural language processing. To improve the model, I performed data preprocessing as one of the easiest ways. among preprocessing methodology, the length-based nomalize and data augmentation was proceeded. To detail augmentation, a clustering model based *AutoEncoder* was designed. At last, the model make the result that *BLEU* score increase significantly. Moreover, I had succeeded in improving the performance of the model to a readable level of the results of the previous model in terms of the results generated. Furthermore, it was demonstrated that this technique is sufficiently helpful for model improvement by significantly improving performance through such data preprocessing and data augmentation.

## 1 Introduction

The *ChatGPT* has brought about an unprecedented change in the world. The previous *Stable Diffusion* was enough without it, but *ChatGPT*'s vast versatility has attracted attention from all industries around the world. In fact, even *Microsoft*, the holding company of *OpenAI*, laid off 30% of its CX (Customer Experience) team, resulting in many people losing their jobs. In the case of Korea, natural language processing is quite difficult due to the nature of Korean, and it has a very passive culture in sharing datasets for this purpose. As a result, there is a problem that the answer accuracy of *ChatGPT* is lower than when the question is asked in English, and many people do not benefit from this technology because various language models have not been released. Fortunately, through a language model called *KoChatGPT*, I can relatively easily implement and experience these big techs. However, the researcher who produced the model also mentioned that it was necessary to verify the quality of the data because the dataset was quickly collected through web crawling. In fact, when learning in a personal environment, the quality was actually not good. I decided to do a project to improve this. There are many ways to improve the model. Models can be improved through various methods, such as additional refinement of datasets, exploration of additional datasets to replace datasets with human feedback, and tuning the decoder's hyperparameters for better generation capabilities. Among other things, I decided to perform data cleaning and preprocessing, which can make a big improvement with relatively little effort. In addition, we will use data augmentation to augment the insufficient number of data to get the most out of deep learning.

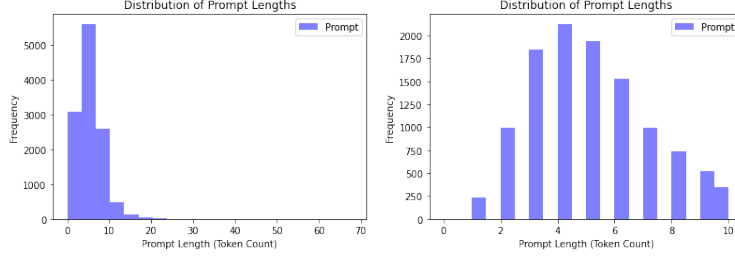


Figure 1: Prompt's length distribution diagrams

## 2 Method

There are various methods that can be considered to improve the performance of *KoChatGPT*. It can be improved by introducing additional new data or by replacing some of the models used in the training process with new models. However, I decided to use the data augmentation. In general, data augmentation are known to help improve the performance of deep learning models. In particular, when used for generative models such as GAN and Diffusion models, its performance can be improved significantly. [1] Although *KoChatGPT* is not a model that handles images, it is expected to demonstrate high performance as a generative model that generates text. Also, It was chosen as the fastest way to produce results due to the lack of time given.

### 2.1 Experiment environment

The experiment environment used in this process was as follows.

- AMD EPYC 7252 8-Core Processor
- NVIDIA A40
- NVME SSD 32GB
- PyTorch 1.13.1
- CUDA 11.6
- cudnn8-runtime

For the convenience of installation, cloud services and docker was used to conducted experiments.

### 2.2 Data Augmentation

It can be seen that the *KoChatGPT* model creator has refined it to some extent, but It was additionally conducted **EDA (Exploratory Data Analysis)** for more efficient processing. This model was mainly used for the **prompt** item as it is important to infer appropriate answers to various prompts.

**Figure 1** visualize the number of tokens a prompt has. Although not shown in the diagram on the left, there seems to be occasional data with a prompt length of more than 20. Of course, a sufficiently long prompt will help to generate an appropriate output, but in this case, the effect is insignificant, so it is erased for more efficient learning. Therefore, as shown in the diagram on the right, data with 11 or more tokens were removed to make the length distribution of prompts close to the normal distribution.

### 2.3 Clustering based on Unsupervised Learning

The next is the domain. In fact, there is no domain-related data in the dataset. In other words, it requires a lot of extra effort to analyze it. To distinguish domain, the cluster model the data with unsupervised learning was used which base on *AutoEncoder* model. Instead of simple machine learning algorithms such as a *KMeans*, the model were designed to be serially connected with *AutoEncoder* and *KMeans* for more effective clustering result, and more layers were stacked on the *AutoEncoder*. During the training model, The hyperparameter modified is `n_clusters` as 7. There was no particular reason, but when adjusted by 1, a balanced distribution was created.

| Layer (type)         | Output Shape  | Param # |
|----------------------|---------------|---------|
| input_1 (InputLayer) | (None, 25943) | 0       |
| dense (Dense)        | (None, 128)   | 3320832 |
| dense_1 (Dense)      | (None, 64)    | 8256    |
| dense_2 (Dense)      | (None, 2)     | 130     |
| dense_3 (Dense)      | (None, 64)    | 192     |
| dense_4 (Dense)      | (None, 128)   | 8320    |
| dense_5 (Dense)      | (None, 25943) | 3346647 |

Table 1: AutoEncoder Model Structure

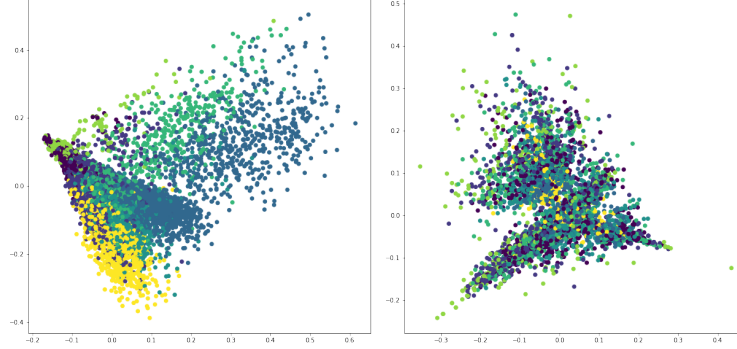


Figure 2: Domain scatter diagrams

The number of total parameters was 6,684,377.

- Total params: 6,684,377
- Trainable params: 6,684,377
- Non-trainable params: 0

## 2.4 KoEDA

A data augmentation was used on the length-based normalized and domain-classified data in previous precess. The **EDA (Easy Data Augmentation)** was used as the method used for data augmentation, which is a technique that can easily perform frequently used text data augmentation techniques such as Random Insertion, Random Swap, Random Deletion, and Synonym Replace, and there was a well-made tool which called *KoEDA*. The set hyperparameters are **a** and **p**, which mean the ratio of words in sentences changed by each augmentation technique and the number of augmented sentences generated per original sentence, respectively. Since the dataset consist of Korean text, instead of Random Insertion and Random Swap, Synonym Replace using *WordNet* and Random Deletion was used. For this purpose, 0.9 was applied as a **p** parameter, and 0.3 was given as an **a** parameter value for Synonym Replace and Random Deletion.

## 3 Result

### 3.1 Analysis Scatter Diagram

As a result of data clustering based on unsupervised learning, two types of scatter diagrams were created. According to **Figure 2**, it was found that the data were distributed similarly to a certain level. Even if the hyperparameter, `n_clusters`, was changed or not, this shape had not changed.

### 3.2 Analysis Histogram

In addition, by visualizing the number of items by class through a histogram, the following conclusions can be inferred. It is definitely biased for some data, and the data augmentation should be made

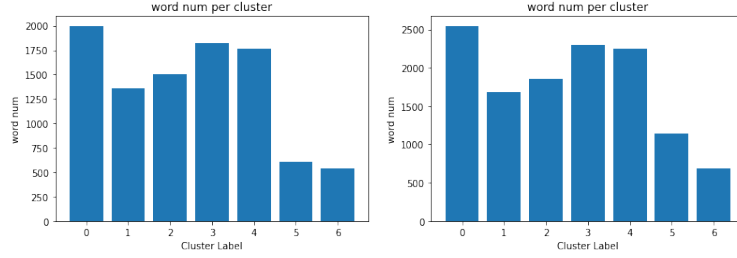


Figure 3: The number of class' items (before/after)

```

Input: 불고기용 고기 한우예요? \
Output: 불고기는 일반적으로 돼지고기로 만들어진 음식입니다. 주로 생선으로 만들어지며, 돼지고기도 일
반적으로 쇠고기의 재료로 사용됩니다. 兪) (일본어: )
五,, -, --, ---, ----, -----\ -- -\

```

Figure 4: Example's input and output (Before)

for certain classes that are deficient. In the experiment, based on the histogram data, I have tried to normalize the data using data augmentation only for the deficient domain. According to **Figure 3**, It can be see that the number of classes 5 and 6 slightly increased as the augmentation ratio was too low. Nevertheless, If There is enough time to increase the ratio, it could be formed a balanced dataset for each domain.

### 3.3 BLEU and ROUGE

After all training, I compared the performance improvement of the model using **BLEU** and **ROUGE** as evaluation metrics. **BLEU** is primarily used to evaluate the accuracy of machine translation, and **ROUGE** is an evaluation metric primarily used to evaluate how well the generated summary captured the original text. These two metrics were used to evaluate whether the generated text was natural, and the results are shown below.

**Before improvement** BLUE Score: 8.917777499809274e-233  
 ROUGE-1: Precision = 1.0, Recall = 0.05, F1 Score = 0.09523809523809523  
 ROUGE-2: Precision = 0.0, Recall = 0.0, F1 Score = 0.0  
 ROUGE-L: Precision = 1.0, Recall = 0.05, F1 Score = 0.09523809523809523  
 ROUGE-Lsum: Precision = 1.0, Recall = 0.05, F1 Score = 0.09523809523809523

**After improvement** BLUE Score: 7.507453527867446e-232  
 ROUGE-1: Precision = 0.42857142857142855, Recall = 0.225, F1 Score = 0.2950819672131147  
 ROUGE-2: Precision = 0.25, Recall = 0.1282051282051282, F1 Score = 0.1694915254237288  
 ROUGE-L: Precision = 0.42857142857142855, Recall = 0.225, F1 Score = 0.2950819672131147  
 ROUGE-Lsum: Precision = 0.42857142857142855, Recall = 0.225, F1 Score = 0.2950819672131147

Overall, the effect of the improved model was noticeable. Older models did not properly generate readable data. In other words, it was difficult to understand meaningful content or context. However, with the improved model, we are able to obtain much more sophisticated and understandable results.

The improved model gives much better output even though it uses the same prompts. As a result, the evaluation index is significantly improved when compared to the previous model. This contributed significantly to improving the performance and accuracy of the model.

In addition, this improved model also helped increase the reliability of data generation. Previously, we had problems generating some incomplete data, but now we provide much more complete and accurate data, which has improved the efficiency of our work and the quality of our results.

```

Input Prompt: ### Instruction(명령어):
시카고 오헤어 국제공항은 어디에 있어

### Response(응답):
Generated Response: ### Instruction(명령어):
시카고 오헤어 국제공항은 어디에 있어

### Response(응답):
시카고 오헤어는 미국 시카고에 있습니다. 『시카고 오헤어 국제공항』 ("Asian Inc.or.kr):『시카고 오헤어 국제공항
(Timua de Passeta Inc.)은 미국 시카고로 불리며, 현재는 미국에서 가장 많이 사용되는 국제공항 중 하나입니다. 현재는 미국과 캐나다
국경을 가로지르는 국제공항으로 사용되는 국제공항입니다.『시카고 오헤어 국제공항(Timua Oreport to Hour):『시카고 오헤어 국
제공항은 주로 국제공항에 위치하고 있으며, 이 국제공항은 시카고 지역의 미국 전역에 위치하고 있습니다. 이 국제공항은 국제공항으로
불리기도 합니다.』:應題:應題:『시카고 오헤어 국제공항에 탑승하실 때는 항공기 탑승이나 항공편 이용하실 때 체크인을 받으시고, 탑승할 때에는 보안
검색을 잘 하시면 됩니다.』: : : : :應提 : : )

```

Figure 5: Example’s input and output (After)

## 4 Conclusion

In this research, various methods were explored to find ways to improve *KoChatGPT*. The way to improve the model using one of these methods was attempted, and several metrics demonstrated that this improvement was real.

In particular, in the case of the *BLEU* score, it clearly showed the degree of improvement according to the change in the number. The *BLEU* score is a widely used metric for evaluating the quality of machine translation, with higher scores indicating better quality translations. The study confirmed a significant improvement in these scores.

Although the effect of data augmentation is still debated, we believe that this experiment has clearly improved it. Data augmentation is a way to generate new data by transforming existing training data, which helps improve the generalization ability of the model. This experiment showed that this method can produce meaningful results in text generation models.

In conclusion, this study demonstrates that data augmentation can play an important role in text generation models. This improves the performance of the model and enables it to better respond to emerging problems. However, it should be kept in mind that proper data augmentation strategies and training procedures are required for these improvements to occur. This research suggests such a direction, and it is expected that it will be an important step in improving the model.

## References

- [1] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” *arXiv preprint arXiv:1712.04621*, 2017.