MAXLENGTH ADAPTATION FOR ENHANCED KOGPT MODEL PERFORMANCE

A PREPRINT

Hong SeoI*
Department of Natural Language Processing
Modu Lab online 5th Research
Seoul, South Korea
hongchaenui@gmail.com

September 20, 2023

ABSTRACT

This research aims to improve the performance of chatGPT, a deep learning model in the field of modern natural language processing. The focus of the study is to adjust the maximum length of the data and apply it to model training. We attempted to minimize data padding by analyzing the length distribution of the data and removing data over a certain length. After learning the model, we evaluated the model's performance using the BLEU indicator and compared the results before and after data processing. As a result, the model's performance improved slightly when the maximum data length was limited, but some problems still existed, and data augmentation technology and additional data were needed to solve them.

Keywords chatGPT · data processing · BLEU

1 Introduction

Deep learning models in modern natural language processing are developing very quickly and powerfully. The beginning of this popularity was the emergence of chatGPT, released by OpenAI in November 2022. ChatGPT is a large-scale language model based on GPT-3.5 that allows natural and informative conversations with users. ChatGPT consists of three models: Supervised Fine Tuning (SFT), Reward Model (RM), and Proximal Policy Optimization (PPO). The SFT model pre-trains ChatGPT with large-scale text and code data. The RM model learns a model that evaluates chatGPT's response based on user feedback. The PPO model reinforces chatGPT using the rewards of the RM model. The goal of this study is to propose a method for learning by adjusting the maxlength of chatGPT by limiting the data length through data preprocessing, and to expect the performance of chatGPT to be improved by this method.

2 Methods

koGPT is the Korean version of chatGPT and is a model learned with large-scale Korean language and code data. koGPT is learned in three steps: Supervised Fine Tuning (SFT), Reward Model (RM), and Proximal Policy Optimization (PPO). At each stage, each model is trained using a different dataset, and through the interaction of these three models, chatGPT can create a chatbot that can communicate with users. There are many ways to improve the performance of koGPT. You can add more datasets, use data augmentation methods, or optimize the model's hyperparameters through hyperparameter search. In addition, various instruction tuning and prompt techniques can be applied. In this paper, we will measure the performance of koGPT using a much simpler method than the methods presented previously. This is how to adjust max length.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

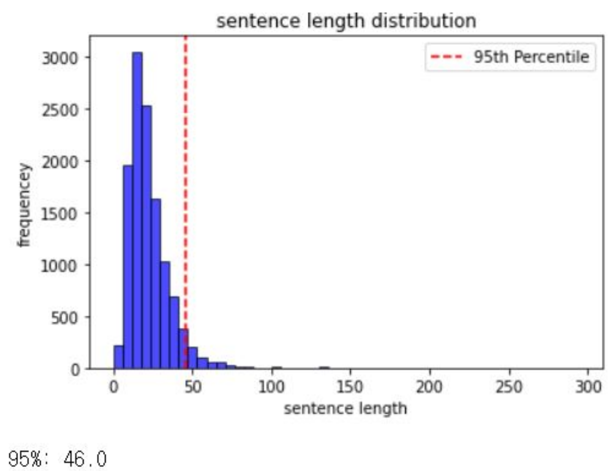
2.1 Experiment environment

The experiment environment used in this process is as follows.

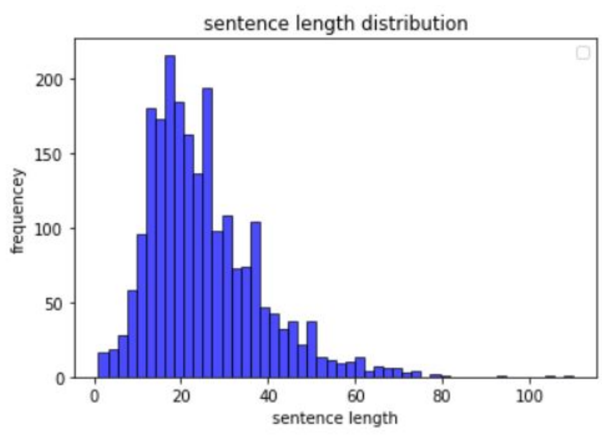
- Torch version:1.12.1
- Cuda version: 11.3
- transformers version: 4.28.0

2.1.1 Data processing

In order to run text data on an artificial intelligence model, the length of each sample of text data must be set to be the same. To keep the length the same, padding is usually applied, but this padding increases the memory usage of the model and reduces model performance. Therefore, in this process, we will aim to improve the performance of the model by checking the length distribution of the data that accounts for 95% of the total dataset distribution in the distribution of the dataset for the SFT and RM models and removing data with a length exceeding a certain length.



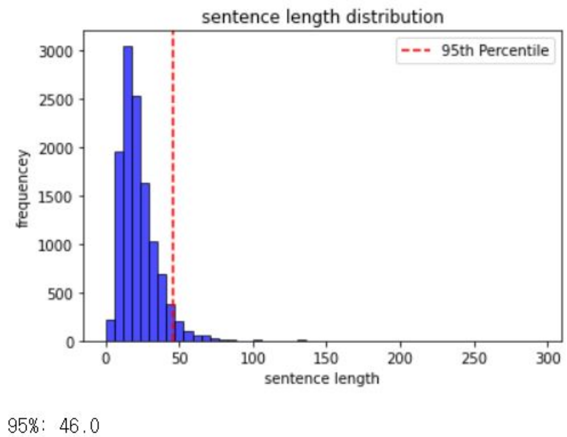
(a) SFT Dataset distribution before preprocessing



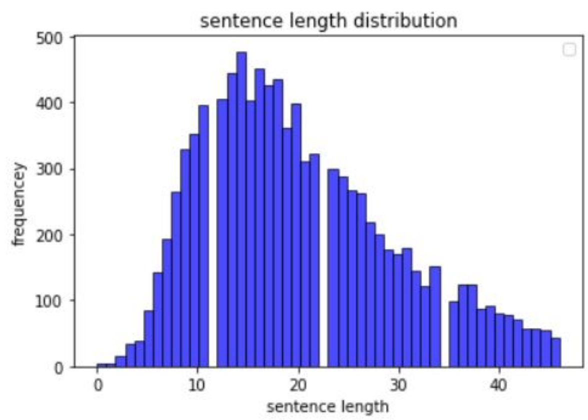
(b) SFT Dataset distribution before preprocessing

Figure 1: SFT Dataset distribution

In the case of the STF dataset, the number of samples of the original data was 12,000, but as a result of only 95% remaining, the number of data samples decreased to 11,435. However, the skewness in the distribution of the data decreased and the asymmetry decreased.



(a) RF Dataset distribution before preprocessing



(b) RF Dataset distribution after preprocessing

Figure 2: RF Dataset distribution

In the case of the rm dataset, the number of samples in the original data was 10220. As a result of preprocessing so that only 95% remained, 9739 data samples remained. Additionally, as a result of preprocessing, the skewness in the distribution was reduced and the asymmetry was reduced.

By preprocessing the data in this way to reduce skewness, the statistical assumptions of the data were satisfied and the data was standardized to make comparison easier.

3 Result

After completing model learning, we attempted to evaluate the model's performance more objectively through the BLEU indicator. The difference in results before and after preprocessing the data is as follows.

3.1 Before improvement

- BLEU score1: 0.0015264903927896083
- BLEU score2: 0
- BLEU score3: 0.03059604336516 66
- BLEU score4: 0.009629943614188135

3.2 After improvement

- BLEU score1: 0

- BLEU score2: 0
- BLEU score3: 0.03986357128268015
- BLEU score4: 0.013679192123121896

After data processing, the BLEU score overall improved. In particular, scores 3 and 4 improved by approximately 0.009 and 0.004. However, the answer to the sentence "Where is Chicago O'Hare International Airport?" used in score3 achieved a relatively high BLEU score, but there is a problem in that the answer is incorrect information. The model answers the question, "Chicago O'Hare International Airport is located in Lake Oakley, St. Arie, United States." However, the actual Chicago O'Hare International Airport is located in Chicago, Illinois.

4 Conclusion

In this study, we attempted to examine whether adjusting the maximum length of data could contribute to improving chatGPT model performance. As a result of limiting the maximum data length, the performance of the model measured through BLEU score slightly increased even though the number of data samples was reduced. However, there was a problem that the hallucination problem still occurred. In order to solve this problem, it is considered essential to conduct further training using data augmentation technology or additional data.

Additionally, in addition to this data-centric approach, performance can be improved by applying model-related techniques. First, instruction tuning and prompting techniques can be applied. Additionally, it may be possible to introduce LoRA or reward ranking algorithms that enable more efficient calculation performance.

There are many different ways to improve model performance. Therefore, what is important is the effort and passion of deep learning engineers who can constantly try various techniques.