COSE474-2023F: Final Project Report
"Better image captioning performance by using LLM"

Hyewon Ryu
Department of Industrial
Management and Engineering
2020170845

1. Introduction

The task of image captioning is a complex task that requires the model to not only understand the image well but also be able to generate appropriate text. Various models that process images are already showing high performance, and with the development of large language models, text generation is also showing high-quality results. In this project, I divided the existing image captioning model into two section, image and text and I tried to show how the use of LLMs in the text generation process can have a positive impact on the overall model.

1.1. Motivation

Nowadays, LLMs are not limited to certain tasks. If appropriate instructions are given, LLMs show high performance not only for general question-answering tasks but also for a variety of previously unseen tasks. Tk-INSTRUCT is a model that pre-trained T5 using large benchmarks of 1,616 NLP tasks, and is a model that produces appropriate output based on task instructions even if an untrained task was given. In this project, based on the performance of the Tk-INSTRUCT, prompt-tuning was performed to do augmentation on text data, and I wanted to check how much this could improve performance in the image-captioning task.

1.2. Problem definition

The image-captioning model is divided into image part and text part. However, since the two parts are not learned independently, it can be expected that the performance improvement in one part will affect the other. Therefore, I want to construct a model structure so that the performance improvement in the text part can result in the image part and the overall capability improvement.

1.3. Contributions

The main contribution of this project is that it presents a pipeline that performs text augmentation through LLMs and reflects it in image-captioning task. In the same way that the image-captioning task is divided into the image part and the text part, the contribution of this project is also divided into the following two parts.

Text augmentation There are variety of ways to perform text augmentation, however, in this project I utilized the LLM, which is based on T5 model, following the trend of studies. I compared several instruction prompts to be delivered to the LLM model and confirmed whether performance improved as a result of actual model training process.

Train image-captioning model In this project, an additional learning process is performed using CLIP, a representative model for image captioning task, as the baseline. At this time, since the dataset is newly constructed through text data augmentation, I propose a method to efficiently reflect this in learning.

2. Methods

2.1. Text augmentation

In the text augmentation process, image information is not used, only captions are used. Tk-INSTRUCT, the LLM used in this project, receives prompt instructions for creating additional captions and original caption data from the train dataset. The prompt instruction consists of a definition and example of the task. Experiments were conducted by dividing the example into whether it included only positives or both positives and negatives. Here, positive and negative examples indicate whether the output is appropriate or inappropriate, respectively. Since the quality of data generated through data augmentation can greatly affect the final model performance, it is also possible to consider not only performing prompt-tuning on the pretrained LLM but also obtaining more accurate augmentation through fine-tuning.

2.2. Single-step training

As new text data is created as a result of text data augmentation, the dataset consists of image, original text, and generated text. Since there are two text data in one image, the same image must be learned twice. However, this causes

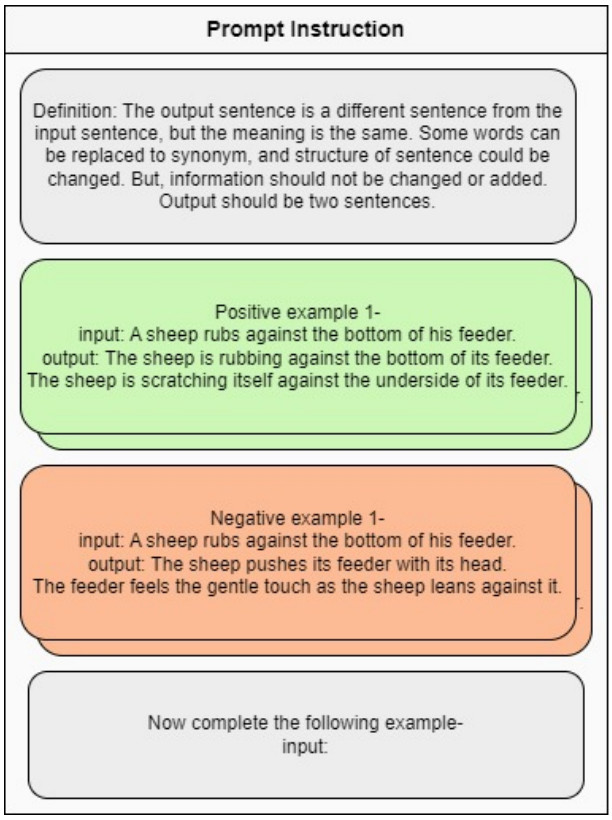

Figure 1. Example of prompt.

Algorithm 1 Single-step training

```
for num_epoch do
    model.train()
    tr_loss = 0
    step = 0
    for batch in trainloader do
        step+ = 1
        optimizer.zero_grad()
        images = batch['image']
        text1 = batch['origin_text']
        text2 = batch['aug_text']
        text1 = clip.tokenize(text1)
        text2 = clip.tokenize(text2)
        logits_img1, logits_txt1 = clip(images, texts1)
        logits_img2, logits_txt2 = clip(images, texts2)

        ground_truth = arange(len(images))
        calculate loss of img1, img2, txt1, txt2
        total_loss = mean of losses

        total_loss.backward()
        tr_loss+ = total_loss.item()
        optimizer.step()
    end for
    tr_loss/ = step
end for
```

a problem of learning efficiency. Therefore, the loss is calculated for each text and integrated based on a certain weight to determine the final loss for the image.
As a result, it was possible to avoid excessive redundancy of image data and increased calculation and to reflect the original text and augmented text according to weight. This process makes it possible to properly control the errors inherent in the augmented data.

3. Experiments

3.1. Dataset

This project uses the MSCOCO dataset. The MSCOCO-2017 dataset consist of 118K/5K training and validation images and also contains a natural language description for each image. In this project, I reduced the size of the dataset. Only 300 image-text pairs are used as training dataset, and 50 image-text pairs are used as the test dataset.

3.2. Expriment setting

I fine-tuned only the captioning model, CLIP, through Adam optimizer, and kept the learning rate constantly at 5e-5. Additionally, the batch size was set to 30. All my experiments were conducted using colab's single Tesla V100 GPU. Epoch number was set to 10. All implementation codes are written in pytorch.
Implementaion codes: *https://github.com/hyewwn/COSE474*

3.3. Results

3.3.1. Text augmentation

As a result of text augmentation performed by applying only prompt tuning to Tk-INSTRUCT, it was confirmed that, contrary to the original intention, the same text as the input text was output. This can be presumed to be due to the fact that the model did not fully understand the task definition and sample given in the prompt or the performance of the used model was insufficient. It seems that this problem can be solved by fine-tuning the Tk-INSTRUCT model, but I was not able to try it during the course of this project.

3.3.2. Single-step training

As mentioned earlier, experiments are divided into two types according to prompt. Experiment 1 is given only a positive example, and Experiment 2 is given both a positive example and a negative example. It is difficult to say that this learning result is appropriate because adequate output was not obtained from the text augmentation earlier. However, in terms of the validity of the model structure, it was confirmed

that the learning was going smoothly through figure2 and figure3.

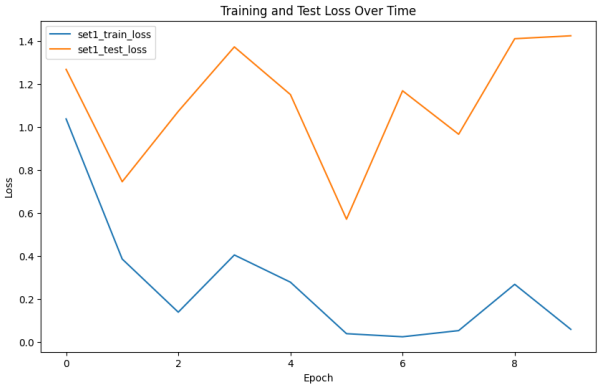

Figure 2. Training and test loss in experiment setting 1.

Figure 3. Training and test loss in experiment setting 2.

Also, I calculated the BLEU score to quantitatively measure the performance of the fine-tuned CLIP. The BLEU score is a method of measuring translation performance by comparing the similarity between machine translation results and human translation results. It can be used regardless of language and has the advantage of fast calculation speed. The larger the value, the better the performance. However, in this project, the calculated bleu score cannot be trusted due to problems with the previous augmentation. However, if only the previous problem is solved, it is significant that a pipeline has been created that can calculate the final performance using the BLEU score.

4. Conclusion

n conclusion, this project aimed to enhance image captioning performance by incorporating Large Language Models (LLMs), specifically Tk-INSTRUCT, for text augmentation. The motivation stemmed from the versatility of LLMs in various natural language processing tasks. The project focused on dividing the image captioning model into image and text components, exploring the impact of LLMs on text generation, and subsequently improving the overall model performance.

The motivation behind utilizing Tk-INSTRUCT for text augmentation was to leverage its ability to perform prompt-tuning on T5-based models for diverse NLP tasks. However, the experiments revealed challenges in achieving the desired results from text augmentation, possibly due to insufficient model understanding of task instructions or limitations in the Tk-INSTRUCT model. Further exploration, particularly through fine-tuning Tk-INSTRUCT, could potentially address these issues.

The single-step training approach was introduced to efficiently integrate the augmented text data into the image captioning model. Despite challenges in text augmentation, the project demonstrated the feasibility of the proposed model structure, showing smooth learning through experimental results.

While the results from text augmentation were not as expected, the project laid the groundwork for future improvements. The proposed pipeline offers a systematic approach to incorporating LLMs for text augmentation in image captioning tasks. The BLEU score, although affected by text augmentation challenges, holds potential for quantitatively measuring performance once these issues are addressed.

Limitation and Future Work

The biggest problem of this project is, of course, the failure in the text augmentation process. Therefore, future work should primarily focus on refining the text augmentation process, potentially through fine-tuning LLMs, to unlock the full potential of this pipeline and improve overall image captioning performance. Excluding the above problems, this project has the following limitations.

1. Trained by too small dataset and epochs
2. The method of increasing training data is not efficient.
3. Data obtained through augmentation may increase errors.

Therefore, future research is needed to increase the efficiency of the model through more efficient learning methods and adjust it to enable learning from more data.

References

Fan, L., Krishnan, D., Isola, P., Katabi, D., & Tian, Y. (2023). Improving CLIP Training with Language Rewrites.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PMLR.

Li, C., Xu, H., Tian, J., Wang, W., Yan, M., Bi, B., ... & Si, L. (2022). mplug: Effective and efficient vision-language learning by cross-modal skip-connections.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics (pp. 311-318).

Wang, Y., Mishra, S., Alipoormolabashi, P., Kordi, Y., Mirzaei, A., Arunkumar, A., ... & Khashabi, D. (2022). Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. arXiv preprint arXiv:2204.07705.