COMP9444 Project Summary

Image Caption Generation Model Optimization

9444 Pro Max

Z5412156 Siyuan Wu

Z5364549 Ruowen Ma

Z5391150 Haonan Peng

Z5431671 Huangsheng Shi

Z5496029 Yinru Sun

I. Introduction

The task of image caption creation often involves the generation of descriptive captions for images. This activity commonly relies on the use of computer vision and natural language processing (NLP) models. In the industrial sector, the integration of image captioning models has the potential to enhance the autonomy of robots by enabling them to assess and interact with their environment, thus facilitating the execution of essential tasks [1]. In various domains, such as social media and e-commerce, where an ongoing stream of images frequently occurs, the utilization of image caption generators can prove beneficial. These generators automatically provide descriptions of the essential attributes portrayed in uploaded images, thereby enhancing user accessibility. Furthermore, this technology can reduce the burden on platform administrators to manually verify compliance with regulatory guidelines concerning image content.

The primary objective of this project is to enhance the generalization capability of CLIP and obtain improved transfer learning outcomes. This will be accomplished through the use of suitable data preparation techniques, the addition of a new unfrozen layer, and the incorporation of CBAM into the existing model.

II. Literature Review

Discovered the LSTM-CNN model and CLIP model in the literature. The LSTM-CNN model combines LSTM networks and CNNs, initially used for time series and video analysis. It's suitable for tasks with sequential data and spatial-temporal dependencies. CLIP, designed for understanding images and text together, is better suited for image-related tasks like classification and object detection. And through the paper, we learned that CBAM can reduce overfitting during the training process.

III.Methods

In the process of text preprocessing, we employed several widely utilized data cleaning approaches in the field of natural language processing (NLP). These techniques encompassed the removal of punctuation, the conversion of all words to lowercase, the elimination of string-like numerical values, and the removal of dangling characters.For image preprocessing, we applied random horizontal-vertical flip, normalizing pixel values, and CLAHE to enhance image contrast, similar to the preprocessing method mentioned in here[3].

Our project employs the base CLIP model with various image models (RN50, RN101, RN50x4, ViT-B/32) and a GPT2 text model to explore how CNNs and Transformers process visual data for image captioning. We evaluate these models using ROUGE, BLEU, METEOR, validation loss, and perplexity to establish benchmarks and understand each model's capabilities. Enhancements include image processing techniques and integrating CBAM for feature emphasis. The final layers of CLIP are unfrozen for training adaptability, aiming for efficient and high-performing image captioning solutions.

IV. Experimental Setup and Dataset Exploration

The objective of our experimental study is to demonstrate the potential improvement in the generalization capability of CLIP. Therefore we aim to have our proposed technique to be applied to a dataset that CLIP did not have pre-trained on. With the condition mentioned above, COCO2017 presents itself as a viable choice for this particular undertaking. The original COCO dataset has 591,000 training images and captions, and 25,000 validation images and captions, with each image provided with five captions. The dataset has a total of 80 distinct classes. The model was trained using the AdamW optimizer, employing a learning rate of 2e-5 and incorporating a warm-up phase consisting of 5000 steps. The loss function employed in our study is cross entropy. The training procedure occurs on Colab's T4 GPU with a RAM capacity of 12G. Due to constraints imposed by the service provider, training processes exceeding a duration of 6 hours may experience interruptions, resulting in the loss of progress. Consequently, we choose to randomly select 1/5 of the data from both the training and validation datasets, and thereafter train on the reduced dataset. The model was initially trained using a batch size of 40 for 30 epochs. However, when a downscaled dataset was used, the occurrence of overfitting increased significantly due to the repetitive training of the same data. Hence, considering the validation loss performance, we have made the decision to reduce the number of epochs to 10 for the final outcome. Presented in the following table are the hyperparameters employed in the model that was utilized.

Model	*Learning Rate*	*Embedding dimension*	*Input resolution*	*Blocks*	*width*	*GPT2 Text Model*		
						layers	width	heads
RN50	2×10^{-5}	1024	224	(3,4,6,3)	2048	12	768	12
RN50 + CBAM	2×10^{-5}	512	512	(3,4,6,3)	2048	12	768	12
RN101	2×10^{-5}	512	224	(3,4,23,3)	2048	12	768	12
RN50x4	2×10^{-5}	640	288	(4,6,10,6)	2560	12	768	12

Table 4.1 CLIP-ResNet hyperparameters

Model	*Learning Rate*	*Embedding dimension*	*Input resolution*	*layers*	*width*	*heads*	*GPT2 Text Model*		
							layers	width	heads
RN50	2×10^{-5}	512	224	12	768	12	12	768	12

Table 4.2 CLIP-ViT hyperparameters

V. Results

Compare four different models in the clip using the same data. We found that for small data sets, the processing results of ViT and the structure of RN101 are better. But for complex data and calculations, RN101 performs relatively well. During the training process, all four models have varying degrees of overfitting. The main reason is that our training data is too small, and

the parameters of epochs do not match the training data well. Then we added some training images and reduced the epochs. Below are the averages of each indicator in the model.

	Bleu Score	Rouge Score	Val Loss	Perplexity	Meteor Score
ViT/B-32	0.000746048	0.091640075	2.853108743	17.6042345	0.055397284
RN50	0.000582041	0.089609143	2.749951068	14.16060801	0.05440529
RN50x4	0.000548616	0.088949979	2.644293264	15.84881778	0.055097486
RN101	0.00068006	0.089776471	2.667882754	14.50194454	0.054949094

Table. Average of each indicator

To enhance the model's performance, we utilized the CBAM method and unfroze layers. CBAM combines spatial and channel attention mechanisms, improving the CNN's attention mechanism by highlighting important spatial locations and channels in input feature maps. This integration enhances the CNN's representational power, potentially leading to improved performance in tasks like image classification, object detection, and segmentation. Unfreezing layers allows the model to focus on the specific task by gradually leveraging general features from lower layers and updating higher layers' representations accordingly. Due to computing resource constraints, we opted for the lightweight RN50 to implement these techniques. The results of combining RN50 with CBAM and unfrozen layers are presented below.

	Bleu Score	Rouge Score	Val Loss	Perplexity	Meteor Score
RN50	0.000582041	0.089609143	2.749951068	14.16060801	0.05440529
RN50 CBAM	0.000562042	0.104865633	3.484697704	32.77507668	0.062642405

Table. RN50 indicator after using CBAM

As we can see from the table, Rouge score and meteor score are all higher than the original training scores. Although the overall performance is not effective enough, we can see the improvement on Rouge score and meteor score after applying CBAM and unfrozen layers. However, the stable but not improving validation loss and perplexity might suggest that CBAM is indeed preventing the model from overfitting by not allowing it to fit too closely to the training data.

VI. Conclusions

Key Strengths:

- CBAM: Significantly enhances feature representation, focusing on vital image regions for richer contextual captions, which prevents overfitting by not allowing it to fit too closely to the training data.
- Transfer Learning with RN50: Leverages extensive pre-training on diverse datasets, crucial for understanding varied visual content.

Key Weaknesses:

- Data Constraints: Limited dataset scope leads to underfitting and reduced model generalization.
- Overfitting: Small dataset size risks the model memorizing data patterns rather than learning them.

Key Limitation:

- Computational Demands: Training large models like CLIP with CBAM is resource-intensive and may not be feasible for all setups.

Recommendations for Future Work:

- Advanced Data Augmentation: To create a more diverse dataset for better model training.
- DAT in ViT: To potentially improve feature extraction and attention mechanisms, addressing overfitting.
- Switching to BLIP: For improved image-text alignment and nuanced language context understanding in captions.
- Ensemble Techniques: Combining various models to leverage their strengths and improve accuracy and robustness.

References

[1] Sathe, S., Shinde, S., Chorge, S., Thakare, S., Kulkarni, L. (2022). Overview of Image Caption Generators and Its Applications. In: Bhalla, S., Bedekar, M., Phalnikar, R., Sirsikar, S. (eds) Proceeding of International Conference on Computational Science and Applications . Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-19-0863-7_8

[2] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021, January 5). *Learning transferable visual models from Natural Language Supervision*. CLIP: Connecting text and images. https://arxiv.org/pdf/2103.00020v1.pdf

[3] Desai, K., & Johnson, J. (2021). Virtex: Learning visual representations from textual annotations. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr46437.2021.01101

[4]Woo, S., Park, J., Lee, J., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. *ArXiv*. /abs/1807.06521