Image captioning

Ivan Yakunin, Leonid Popov, Vadim Lanko

May 2023

Abstract

In this project we propose a solution to the image captioning problem. The model used in our approach was BLIP. **COCO 2014** dataset was used for experiments. The model was compared with the baseline and existing solutions.
`https://github.com/Moonrise6377/stable_diffusion`.

1 Introduction

Image captioning, an automated process of generating descriptive text that describes the content of an image, presents a complex challenge with wide-ranging applications across numerous research domains. Some examples of possible applications are: Accessibility for Visually Impaired, Social Media Analysis, Visual Question Answering, Image Description for Media and Advertising, Automatic Image Captioning Systems (photo-sharing platforms, etc.).

In recent years, there has been remarkable progress in the field of image captioning, largely due to the groundbreaking advancements in deep learning techniques. The emergence of deep learning has paved the way for innovative approaches that leverage large-scale image datasets and sophisticated neural architectures, enabling the creation of highly effective and contextually relevant image captions. These advancements have not only improved the quality and accuracy of image captioning but have also opened up new paths for multimodal research, where the fusion of visual and textual information leads to enhanced understanding and interpretation of visual content.

In our project we try to use one of the latest approaches that is aimed at solving the image captioning problem (BLIP) and test its performance on a well-known dataset (COCO).

1.1 Team

The project was made by Ivan Yakunin, Leonid Popov and Vadim Lanko.

Ivan Yakunin prepared technical documentation (section 3, readme), conclusion (section 7), designed the repository.

Leonid Popov prepared section 5 of the report, made ResNet & LSTM model.

Vadim Lanko prepared sections 1, 2, 4 and 6 of the report, made BLIP evaluation.

2 Related Work

Existing approaches can be divided in two large groups: common solutions and deep learning-based techniques. Here is a brief review on these approaches.

2.1 Common Solutions

Image captioning is presented as a sequence-to-sequence problem in machine vision. One of the essential sequence-to-sequence problems is machine translation. One of the most common solutions to the image captioning problem is inspired by machine translation. In this class of methods, known as the "encoder-decoder" methods, the input image is mapped to an intermediate representation of the image contents. It is then converted to a sequence of words that make up the caption of the image. To detect objects in the image and obtain a rich representation, CNNs are commonly used in the encoding stage. The final layer serves as a feature vector. In the decoding stage, RNNs are often used as they are good at representing human language and text. After the image and its corresponding caption in the dataset are mapped into a common space, the correspondence between the two representations is learned and new captions are generated for new images. Despite significant results, these methods usually give general and vague captions for images and do not describe image contents appropriately since all information is compressed into a single vector. This causes problems with learning the information at the beginning of the sequence and the deeper relations between image contents.

In such approaches the image is fed to some CNN (ResNet, etc.) and its output can be passed, for example, to a RNN, LSTM or GRU (Gated Recurrent Unit)[Cho et al., 2014].

In recent works from 2018 to 2022, these methods are rarely used. Methods based on the attention mechanism and graphs have been used more frequently.

2.2 Deep Learning-Based Solutions

Attention-based methods in image captioning are inspired by human attention patterns, where models are trained to focus on important image features. These methods use a sequential decoder with a context vector and generate a weighted sum of feature vectors to guide the decoding process. Soft attention assigns weights to relevant image features, while hard attention uses stochastic sampling [Xu et al., 2015]. Other variations include multi-head attention, stacked attention, and attention on attention (AoA), which improve the quality and relevance of attended information. The main limitation is the lack of identifying the object relations within the image.

Graph-based methods in image captioning use graphs to represent the relations between elements, enhancing caption quality [You et al., 2016]. Two types of graph extraction are commonly used: scene graph extraction from images and textual data. Scene graphs connect objects, attributes, and relationships. CNNs extract visual features, and GCNs encode scene graphs. LSTM decoders generate captions. These methods improve caption generation by leveraging spatial and semantic relations.

Convolution-based methods have also gained popularity in vision-language tasks. [Aneja et al., 2018] introduced a convolutional model with masked convolutions instead of recurrent functions. [Wang and Chan, 2018] proposed a CNN-based framework with modules for vision, language, attention, and prediction. Hierarchical attention was utilized to capture different levels of concepts. Convolution-based models offer more caption diversity and perform well in classification tasks without vanishing gradients. However, further improvements are needed in terms of performance based on evaluation metrics.

Transformer-based methods have gained attention as alternatives to RNNs and LSTMs in vision-language tasks [Vaswani et al., 2017]. Some of such models are: the "Object Relation Transformer" (incorporating geometric attention); combined Transformer-like encoder with an LSTM decoder; "ETA-Transformer" framework (improving EnTangled Attention (ETA) to combine semantic and visual information); the CaPtion TransformeR (CPTR) with a Transformer encoder for sequentialized raw images; PureT (a Transformer-based architecture using SwinTransformer); ViTCAP (a Vision Transformer-based model with a lightweight Concept Token Network (CTN)); COS-Net (a Transformer-style encoder-decoder structure using a CLIP model for cross-modal retrieval); the SP module for spatial awareness and the SR module for scale-wise reinforcement. Transformer-based methods offer improved flexibility and address limitations of RNNs and LSTMs in vision-language tasks.

Vision Language Pre-Training Methods for Image Captioning [Hu et al., 2022] aim to reduce reliance on fully-supervised learning. BLIP (Bootstrapping Language-Image Pre-Training) is a widely used model that provides a shared representation for images and text. BLIP effectively utilizes the noisy web data by bootstrapping the captions, where a captioner generates synthetic captions and a filter removes the noisy ones. BLIP is a new VLP framework which enables a wider range of downstream tasks than existing methods. It introduces two contributions from the model and data perspective, respectively: Multimodal mixture of Encoder-Decoder (MED), Captioning and Filtering (CapFilt). BLIP achieves state-of-the-art performance on a wide range of vision-language tasks, including image-text retrieval, image captioning, visual question answering, visual reasoning, and visual dialog.

It is worth mentioning that it is possible to combine the methods mentioned above in order to improve the flaws of each of such methods. Further, we'll focus on the utilization of the **BLIP** model and testing its performance on **COCO** dataset.

3 Model Description

3.1 Model choice

We chose this model because experimenting with models and their results highlighted the following advantages of the BLIP model:

1. Multimodal Understanding: BLIP combines language and image pre-training, allowing it to understand and generate text in the context of visual information. By leveraging both textual and visual cues, BLIP achieves a deeper understanding of multimodal data, enabling tasks such as image captioning, visual question answering, and text-to-image synthesis.

2. Enhanced Cross-Modal Transfer: BLIP's pre-training on a large-scale multimodal dataset enables it to learn powerful representations that facilitate transfer learning between different modalities. This means that knowledge gained from one modality (e.g., text) can be effectively transferred and applied to another modality (e.g., images), leading to improved performance on various multimodal tasks.

3. Robustness to Noisy Inputs: BLIP's joint language-image training helps enhance the model's robustness to noisy or incomplete inputs. By learning to associate visual and textual information, the model can compensate for missing or ambiguous data, improving its ability to handle real-world scenarios with imperfect or incomplete multimodal inputs.

In summary, the advantages of the BLIP model lie in its multimodal understanding, enhanced cross-modal transfer capabilities, and robustness to noisy inputs, making it a valuable tool for multimodal NLP tasks and applications.

3.2 Processing structure

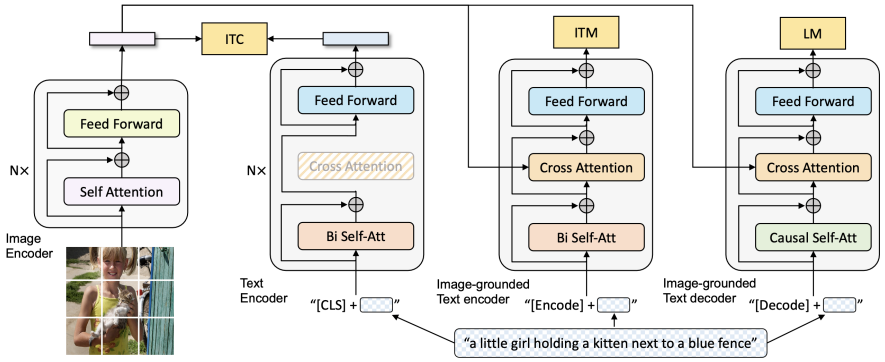

Figure 1: Pre-training model architecture and objectives of BLIP

What could be better with such capabilities as understanding and generative description, a multimodal encoder-decoder (MED) is used, which can be used in one of three ways: a unimodal encoder, an image-bound text encoder, and an image-bound text decoder. We jointly optimize three objectives during pre-training, with two goals based on understanding and one goal based on generation. Image and Text Contrast Loss (ITC) activates unimodal encoding. Loss of image-to-text correspondence (ITM) activates the image-bound text encoder. ITM is a binary classification problem in which the model uses an ITM head (linear layer) to predict whether an image-text pair is positive (coincident) or negative (incomparable), given their multimodal feature. The loss of Language modeling (LM) activates an image-based text decoder, which is designed to generate text descriptions from an image. It optimizes the cross entropy loss, which trains the model to maximize the probability of text appearance by autoregression. Image-grounded text decoder replaces the bi-directional self-attention layers with causal self-attention layers, and shares the same cross-attention layers and feed forward networks as the encoder. The decoder is trained with a language modeling (LM) loss to generate captions given images. In order to perform efficient pre-training while leveraging multi-task learning, the text encoder and text decoder share all parameters except for the SA layers. The reason is that the differences between the encoding and decoding tasks are best captured by the SA layers. In particular, the encoder employs bi-directional self-attention to build representations for the current input tokens, while the decoder employs causal self-attention to predict next tokens. On the other hand, the embedding layers, CA layers and FFN function similarly between encoding and decoding tasks, therefore sharing these layers can improve training efficiency while benefiting from multi-task learning

3.3 CapFilt

New to VLP models in BLIP is the use of CapFilt. Alternative texts often inaccurately describe the content of the image, and high-quality image-text pairs are available in smaller quantities than collected from the Internet. Therefore, the authors of the model proposed a new CapFilt method that allows generating captions to noisy images and filtering noisy image-text pairs. The filter is configured individually after initialization using the COCO dataset. In particular, CapFilt decodes the text and then compares the text to match the image. It is finetuned with the ITC and ITM objectives to learn whether a text matches an image. Synthetic image descriptions are generated for the collected image-text pairs (Ts). The filter removes noisy texts in both the original web texts (Tw) and the synthetic texts (Ts), where a text is considered to be noisy if the ITM head predicts it as unmatched to the image. As a result, it is possible to clear the data from noise, having received, although not accurate, but more appropriate descriptions of the images. Next, the final dataset is assembled from filtered image-text pairs, which are combined with human-annotated descriptions. And the model is already being trained on this data.

When only the captioner or the filter is applied to the dataset with 14M im-

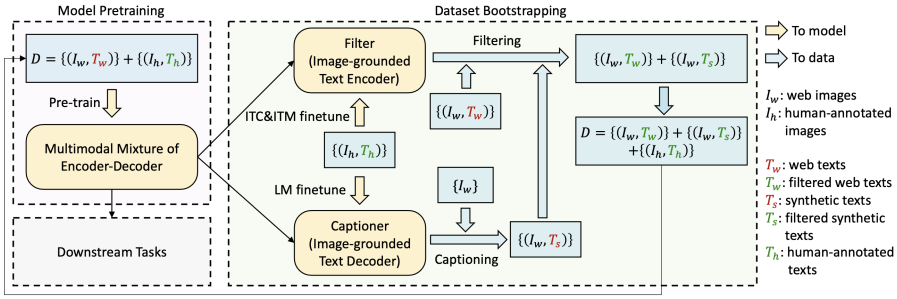

Figure 2: Learning framework of BLIP

Figure 3: Example of CapFilt on web and syntetic text

ages, performance improvement can be observed. When applied together, their effects compliment each other, leading to substantial improvements compared to using the original noisy web texts. CapFilt can further boost performance with a larger dataset and a larger vision backbone, which verifies its scalability in both the data size and the model size. During CapFilt, the captioner and the filter are end-to-end finetuned individually on COCO. In the study the effect if the captioner and filter share parameters in the same way as pre-training. The performance on the downstream tasks decreases, which we mainly attribute to confirmation bias. Due to parameter sharing, noisy captions produced by the captioner are less likely to be filtered out by the filter.

3.4 ResNet & LSTM model

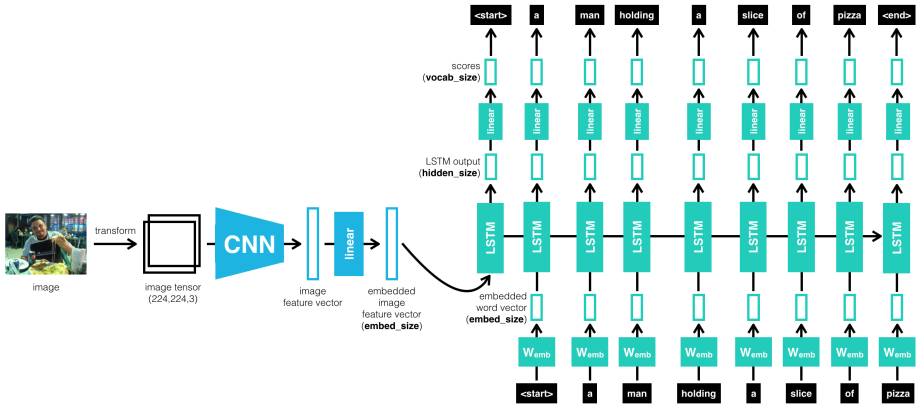

Figure 4: ResNet & LSTM scheme

The diagram illustrates the image captioning approach, showcasing the EncoderCNN and DecoderRNN components. The EncoderCNN encodes the essential information from an input image into a fixed-size feature vector. This vector is then fed into the DecoderRNN, represented by unfolded LSTM cells at different time steps. The output words are generated one by one, with each word being used as input for the subsequent time step, until the complete caption is generated. The arrow connecting the LSTM cells represents the hidden state information, which serves as the network's memory and is also fed back to the LSTM at each time step.

The architecture consists of a CNN encoder and an RNN decoder. The CNN encoder utilizes a pre-trained ResNet model trained on ImageNet, a convolutional neural network with skip connections. This architecture has proven effective in image recognition tasks due to its ability to capture residual differences before and after convolution using identity blocks. By dropping the last fully connected layer and replacing it with a trainable fully connected layer, the CNN encoder transforms the final feature map into a more suitable encoding for the RNN decoder.

RNNs, particularly LSTMs, have demonstrated their usefulness in language-related tasks, thanks to their ability to model sequential data. LSTMs incorporate both long-term and short-term memory information within the network. For the captioning task, an LSTM-based decoder is employed, following the concept of sequence-to-sequence models used in translation. The initial input for the RNN decoder consists of the concatenation of the CNN encoder output with the "START" token. A fully connected layer is applied to the hidden states at each time step to generate a softmax probability distribution over the entire vocabulary. The word with the highest probability is selected as the generated word for that time step. This process continues until either a maximum caption length is reached or the "STOP" token is generated, indicating the end of the

sentence.

LSTM Decoder In the project, all inputs are passed to the LSTM as a sequence. The sequence format includes a feature vector extracted from the input image, followed by the start word, next words, and so on.

Embedding Dimension To ensure consistency in size, the LSTM expects each input in the sequence to have the same dimensions. Therefore, both the feature vector and each word are embedded into a common size known as the embed_size.

4 Dataset

In the project we use COCO 2014 dataset. More information about it can be found in the research paper [Lin et al., 2014]. The dataset is available for download on the official website.

The COCO (Common Objects in Context) dataset is a widely used benchmark dataset for object detection, segmentation and captioning tasks. We will focus on the image captioning part of this dataset.

The dataset contains five human-generated captions for every image in train, val and test sets. The table provides a brief statistics of the images distribution over these parts.

	Train	Valid	Test
Images	82783	40504	40775

Table 1: Statistics of the COCO 2014 dataset.

Examples of images with the corresponding captions are provided below.

A man with three cows on road next to trees.
A man holding a stick walks with three cows down a street.
A man runs with his 3 cattle along a street
A herder in white walking on a road with his cattle.
A man walking besides cows down a road.

(a) 1

Multiple people playing with a soccer ball on the beach.
People are playing volleyball on the beach beside the ocean.
A group of boys playing soccer on the beach.
A group of people are playing soccer on a beach.
the men play soccer on the beach with no shoes

(b) 2

A man riding a skateboard down a set of steps.
A person in midair as he skateboards up several steps.
A MAN IS JUMPING ON THE SKATE BOARD
This dextrous fellow rides his skateboard on stairs.
A young man turns on his skateboard mid-air over the concrete steps.

(c) 3

A doughnut sitting on top of a brown napkin.
A donut on a napkin with a fork on a table
A table with a beverage and a donut with a fork next to it.
A doughnut sitting on a napkin with a fork and a cup of coffee.
A cake donut in front of a cup of coffee.

(d) 4

Figure 5: Examples from COCO dataset

5 Experiments

5.1 Metrics

BLEU-4 was used as a metric to evaluate the performance of each model. The formula is provided below.

$$BLEU = \min\left(1, \frac{L_{hypothesis}}{L_{reference}}\right) \cdot \left(\prod_{i=1}^{n} Precision_i\right)^{\frac{1}{n}}$$

$$Precision_i = \frac{i - grams_{clipped}}{i - grams_{total}}$$

5.2 Experiment Setup

ResNet & LSTM model During the training of our neural network model, we employed the cross entropy loss function.

Learing rate	0.001
Batch size	128
Vocabulary threshold	5
Embedding size	256
Hidden size	512
Number of epochs	3

Table 2: Parameters

Data transformations First, we resize the images, scaling the shorter edge to a size of 256 pixels. This step helps standardize the image dimensions for further processing.

Next, we randomly crop a region of size 224x224 from the resized image. This random cropping allows us to extract different regions of interest, capturing diverse visual information.

To enhance data augmentation, we incorporate a random horizontal flipping operation. With a 50% probability, we horizontally flip the image, which helps introduce variety and reduce potential biases.

Lastly, we perform normalization on the image. Specifically, we apply mean and standard deviation values of (0.485, 0.456, 0.406) and (0.229, 0.224, 0.225) respectively. This normalization aligns the image data with the pre-trained model's expectations, improving compatibility and overall performance.

The training process for our model took approximately 5 hours to complete three epochs using an A100 graphics card.

BLIP model In our work, we incorporated the use of the "Blip Pretrained Large" model.

Both models were evaluated on the val set.

5.3 Baselines

The ResNet & LSTM model was used as a baseline to be compared with. Some of the previous solutions and their scores are mentioned in the next section.

6 Results

In this section we present the achieved results. We provide the BLEU score for our baseline model and BLIP. These scores are compared to some prevoius art scores on the same dataset below in the table 3.

Model	BLEU-4 score
BLIP	0.36
ResNet & LSTM	0.21
CapDec[Nukrai et al., 2022]	0.26
ClipCap[Mokady, 2021]	0.32
VLKD[Dai et al., 2022]	0.17

Table 3: Score comparison

The achieved score for BLIP model is quite high compared to other existing approaches. It is important to note that the model was not fine-tuned on the COCO dataset before evaluating. It implies that BLIP can work very well even out-of-the-box.

The sample output of the BLIP model and the baseline could be found below in Tab. 4.

there are many cows that are walking down the road together.

Table 4: Output sample

The comparison of captions of BLIP and the baseline model for corresponding pictures is demonstrated below 6.

(a) ResNet & LSTM: "a herd of elephants walking along a lush green field"
BLIP:"there are many cows that are walking down the road together"

(b) ResNet & LSTM: "a man with a tie and a suit jacket."
BLIP:"they are dressed up as soldiers and a woman in a costume"

(c) ResNet & LSTM: "a person is skiing down a snowy mountain."
BLIP:"skiers on a snowy slope with trees in the background"

(d) ResNet & LSTM: "a woman holding a tennis racket on a tennis court"
BLIP:"woman playing tennis on a tennis court with a tennis racket"

Figure 6: Prediction examples

The captions generated by BLIP are more meaningful. They catch more details and describe objects on the pictures better.

7 Conclusion

In this project, we applied new knowledge gained during the study of methods and models in NLP. We used a pre-trained BLIP model to predict image captions. We utilized the COCO dataset for qualitative data and employed the BLEU metric for evaluation.

The baseline approach involved a CNN encoder, which employed a pre-trained ResNet model trained on ImageNet, and an LSTM-based decoder for generating captions. The pre-trained model was locally fine-tuned on the COCO dataset, and the results were showcased, including several image captions.

The findings suggest that the pre-trained BLIP model exhibited high efficiency in generating image captions, especially when the data was cleaned from noise using the CapFilt technique. The utilization of the qualitative data from the COCO dataset and the incorporation of well-established evaluation metrics further validate the effectiveness of the proposed method.

References

[Aneja et al., 2018] Aneja, J., Deshpande, A., and Schwing, A. G. (2018). Convolutional image captioning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5561–5570.

[Cho et al., 2014] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

[Dai et al., 2022] Dai, W., Hou, L., Shang, L., Jiang, X., Liu, Q., and Fung, P. (2022). Enabling multimodal generation on CLIP via vision-language knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2383–2395, Dublin, Ireland. Association for Computational Linguistics.

[Hu et al., 2022] Hu, X., Gan, Z., Wang, J., Yang, Z., Liu, Z., Lu, Y., and Wang, L. (2022). Scaling up vision-language pretraining for image captioning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17959–17968.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham. Springer International Publishing.

[Mokady, 2021] Mokady, R. (2021). Clipcap: Clip prefix for image captioning. *ArXiv*, abs/2111.09734.

[Nukrai et al., 2022] Nukrai, D., Mokady, R., and Globerson, A. (2022). Text-only training for image captioning using noise-injected clip. pages 4084–4092. Publisher Copyright: © 2022 Association for Computational Linguistics.; 2022 Findings of the Association for Computational Linguistics: EMNLP 2022 ; Conference date: 07-12-2022 Through 11-12-2022.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Wang and Chan, 2018] Wang, Q. and Chan, A. B. (2018). Cnn+cnn: Convolutional decoders for image captioning. *ArXiv*, abs/1805.09019.

[Xu et al., 2015] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2048–2057, Lille, France. PMLR.

[You et al., 2016] You, Q., Jin, H., Wang, Z., Fang, C., and Luo, J. (2016). Image captioning with semantic attention. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4651–4659.