Shahid Beheshti University
Data Science (2023-2024)

Final report template

Sina Tayebi
Department of Computer Science
Shahid Beheshti University
si.tayebi@mail.sbu.ac.ir

Ariana Aghamohammadi
a.aghamohammadi@mail.sbu.ac.ir
Department of Computer Science
Shahid Beheshti University

Afshin Jalili
a.jalili@mail.sbu.ac.ir
Department of Computer Science
Shahid Beheshti University

Abstract

Image Captioning (IC) has achieved astonishing developments by incorporating various techniques into the CNN-RNN encoder-decoder architecture. However, since CNN and RNN do not share the basic network component, such a heterogeneous pipeline is hard to be trained end-to-end where the visual encoder will not learn anything from the caption supervision. This drawback inspires the researchers to develop a homogeneous architecture that facilitates end-to-end training, for which Transformer is the perfect one that has proven its huge potential in both vision and language domains and thus can be used as the basic component of the visual encoder and language decoder in an IC pipeline. Meantime, self-supervised learning releases the power of the Transformer architecture that a pre-trained large-scale one can be generalized to various tasks including IC. The success of these large scale models seems to weaken the importance of the single IC task. However, we demonstrate that IC still has its specific significance in this age by analyzing the connections between IC with some popular self-supervised learning paradigms.

1 Introduction

In this context, the overarching goal is to address the challenges associated with the traditional CNN-RNN encoder-decoder architecture in Image Captioning (IC). The conventional heterogeneity of CNN and RNN components within this pipeline impedes seamless end-to-end training, particularly as the visual encoder struggles to glean valuable insights from caption supervision. Recognizing this limitation, researchers have pivoted towards a homogeneous architecture, leveraging the versatile Transformer model as a foundational component for both the visual encoder and language decoder in the IC pipeline. The Transformer's demonstrated efficacy in both vision and language domains makes it an ideal choice for enabling end-to-end training and fostering improved information flow between the visual and language components. Furthermore, the incorporation of self-supervised learning amplifies the potential of the Transformer architecture. Large-scale pre-trained Transformers

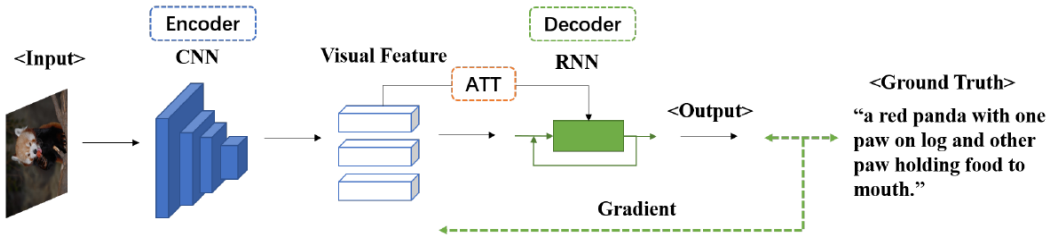

Figure 1: The heterogeneous encoder-decoder architecture for IC, where the visual encoder is a CNN and the language decoder is an RNN. Since such a model is hard to be trained end-to-end, the gradient can not be backpropagated to the visual encoder.

exhibit remarkable generalization capabilities across various tasks, including IC. Despite the success of such broad-spectrum models, there is a nuanced exploration of the continued significance of IC in the era of these expansive architectures. This investigation seeks to unravel the specific relevance of IC by scrutinizing its connections with popular self-supervised learning paradigms. By doing so, the study aims to shed light on the distinctive challenges and opportunities that persist within the realm of Image Captioning, emphasizing its enduring importance in the face of evolving self-supervised learning approaches.

2 Proposed method

2.1 heterogeneous architecture

The first method of our work is using a heterogeneous architecture with a pre-trained CNN-based visual encoder and RNN-based language decode.In this architecture, the visual encoder extracts the visual features and these features are input into the language decoder for captioning. Since the visual encoder (CNN) and the language decoder (RNN) do not share the same structure, this architecture is considered heterogeneous. The major problem of such heterogeneous architecture is that the whole model is hard to be trained end-to-end. This is mainly because CNN and RNN do not share the basic network component, then the optimization strategies, e.g. The optimizer or the learning rate of the encoder and decoder are hard to be unified. To remedy this, researchers divide the training of the visual encoder and the language decoder. Specifically, they pre-train a visual encoder and then fix it. When the model is trained by the caption supervisors, the parameters of the visual encoder will not be updated. As a result, the gradients are not backpropagated from the word-level supervision to the pixel-level input, as shown in Figure 1 that the gradient (the green dash line) does not transmit to the CNN. This means that these heterogeneous architectures are not really end-to-end trained. Thus, the visual encoder fails to learn high-level semantic knowledge from the caption supervisions and the extracted visual features have determined the upper bound of the generated captions' quality, where the gap between vision and language domains is still huge.[reference to the survey]

2.1.1 network architecture

We used a pre-trained resnet model (resnet50) as visual encoder and a LSTM with two hidden layers with 512 neurons as language decoder. the encoder that we provide to use the pre-trained ResNet-50 architecture (with the final fully-connected layer removed) to extract features from a batch of pre-processed images. The output is then flattened to a vector, before being passed through a Linear layer to transform the feature vector to have the same size as the word embedding[Figure 2].

2.1.2 Data transformation and normalization

It covered resizing, random cropping, random horizontal flip, converting to tensor, followed by normalization. This is a standard image transformer that we used for the current task. These transformations are what is needed for the ResNet50 since it is a pre-trained model we need to make sure the images are in the expected size and normal form. Also we normalized captions with converting them to lowercase and removing the stopwords. Also we tokenize them to feed into our LSTM language decoder.

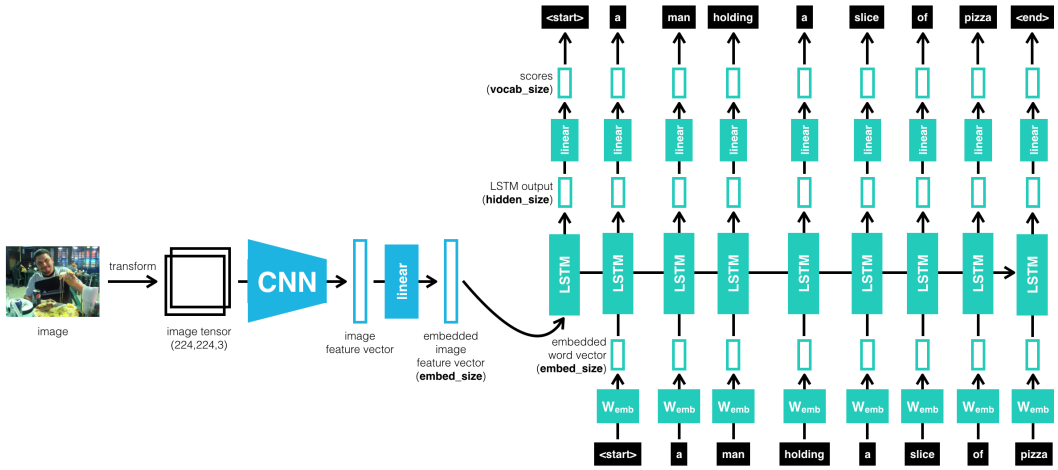

Figure 2: network architecture

2.1.3 Training

We consulted the two papers listed above "Show and Tell: A Neural Image Caption Generator" and "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention" [reference] to gain a better understanding on the overall process and framework to implement an Image Caption. The overall approach includes an Encoder and a Decoder. The Encoder leveraged a pre-trained CNN architecture like ResNet-50 to help extract features from the images. The goal of the encoder is to encode the content of the image into a smaller feature vector. Which later gets passed to the Decoder (RNN) network. The Decoder which is our LSTM that we use to generate the captions for an image. We pass the encoder embedding as input to the LSTM to learn. The vocabulary in the training pool are pretty much all the unique words in our data set, with the added <start> and <end> tokens to indicate start and end of a sentence. We used a vecob-hreshold to 4. The number of words was better than 5, and we did not want to increase the vocab words to include every rare word that may have only been used once. A balance would be key, so we went with 4. for the batch-size we used 10 for our training. With enough resources, we could use greater values for batch size like 32. embed-size is used both in embedded image feature vector and word embedding. In this case we used 256 and a hidden-size of 512. For the trainable parameters, we made all the weights in the decoder trainable, which at every iteration they get updated. For the encoder, we used to only update the weights in the embedding layer. we selected Adam optimizer, instead of SGD. From past experiences, we have seen it perform/converge faster. When we first started the training we used the wrong learning rate of 0.001 which was too high and there was no improvement and had to end it after 1 epoch. We updated the Learning Rate to 0.0005 and noticed significant improvement in the overall learning.

2.2 Homogeneous architecture

In order to overcome the problem of end-to-end training with our heterogeneous network, we used a transformer based network architecture, which is we used a transformer-based model as visual encoder and also a transformer-based language model as decoder. Since both encoder and decoder are transformer-based, we can have end-to-end training and the gradient could be back propagated from language decoder in word-level to visual encoder in pixel-level. As sketched in [Figure 3], a straightforward homogeneous architecture can be configured as follows: the visual encoder is set as a pre-trained vision Transformer **[Liu et al., 2021b]** and the language decoder is set as a classic Transformer **[Vaswani et al., 2017]**. Since now the architecture is homogeneous that the optimization strategies of both the encoder and decoder can be unified, the whole model can be end-to-end trained, i.e., as shown in [Figure 3], the gradient (the purple dash line) can be backpropagated from the word-level supervision to the visual Transformer. In this way, the visual encoder can learn high-level semantic knowledge from the language supervisions, while the encoder of the previous heterogeneous architecture can not. Given this homogeneous prototype, researchers can further refine it for generating better captions. To help the readers get some preliminary ideas.**[reference to survey]**

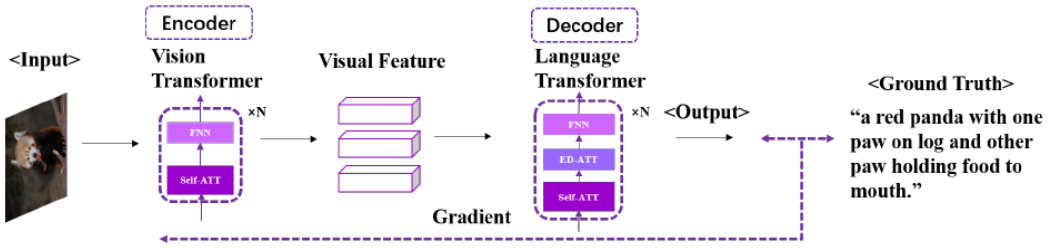

Figure 3: A Transformer-based homogeneous encoder-decoder architecture for IC, where the visual encoder and language decoder are both Transformer networks. Since this homogeneous architecture facilitates the end-to-end training, the gradient can be backpropagated to the visual encoder.

2.2.1 Network architecture

We connected two transformer-based model as a encoder-decoder transformer network. For visual encoder, we use VIT(visual transformer) and for the language decoder, we use GPT-2.

2.2.2 Encoder architecture

The Visual Transformer (ViT) is a neural network architecture that applies the transformer model, originally designed for natural language processing, to image data. It represents an image as a sequence of fixed-size non-overlapping patches, each linearly embedded and flattened. Positional embeddings are added to maintain spatial relationships between patches. The core of ViT is the transformer encoder, with self-attention mechanisms capturing non-local dependencies. The final output is generated by passing the representation of the [CLS] token through fully connected layers. ViT is pre-trained on a large dataset using tasks like predicting the order of shuffled patches and can be fine-tuned for specific computer vision tasks.**[refrence to VIT]**

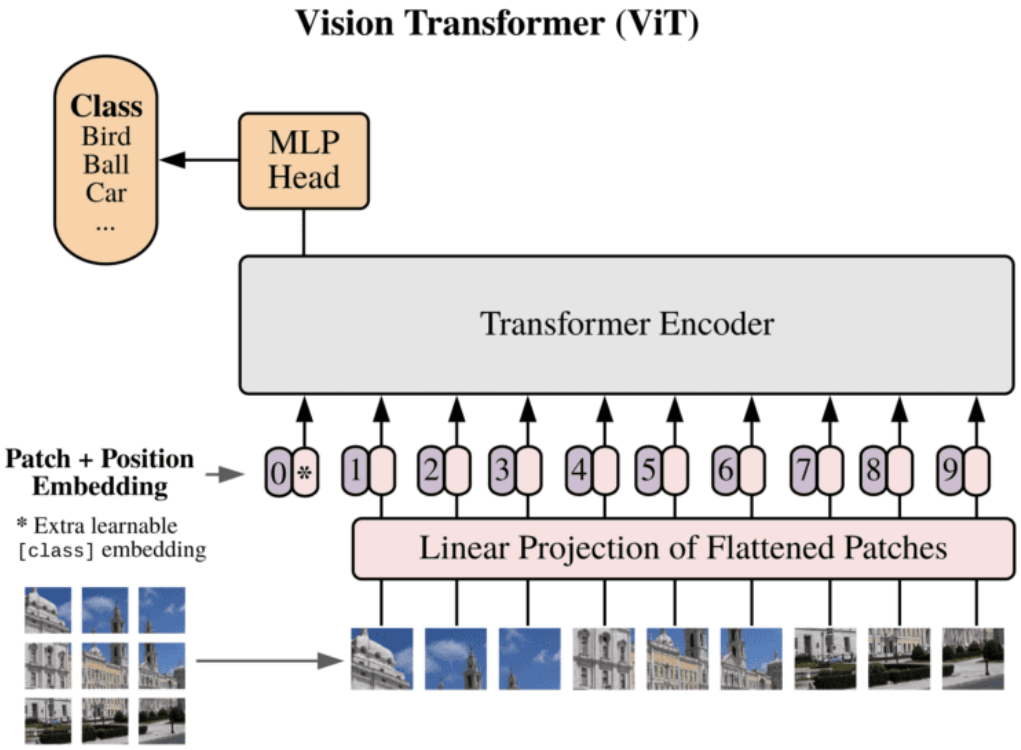

Figure 4: Visual transformer architecture

2.2.3 Decoder architecture

The *GPT-2* (Generative Pre-trained Transformer 2) network architecture is a powerful language model developed by *OpenAI*. It belongs to the transformer architecture family and is characterized by its attention mechanisms**[refrence to attention is all you need]**, enabling it to capture contextual relationships in sequential data efficiently. GPT-2 employs a decoder-only transformer architecture with a massive number of parameters, reaching up to 1.5 billion or more in larger versions. It operates on a pre-training and fine-tuning paradigm, where it is first pre-trained on a large corpus of text using unsupervised learning. The model's key innovation lies in its ability to generate coherent and contextually relevant text across a wide range of tasks, such as text completion, translation, and question-answering.

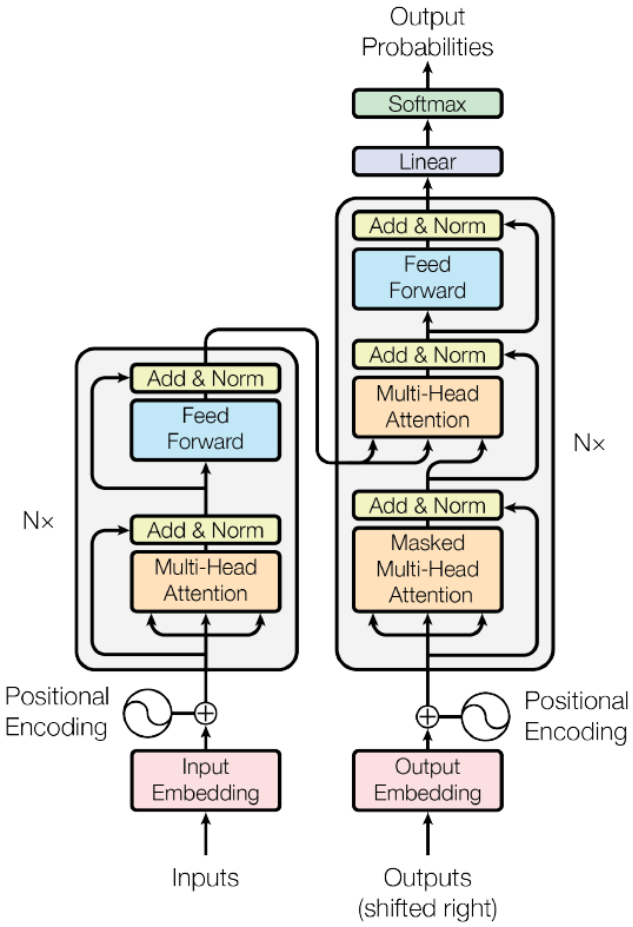

Figure 5: GPT-2 architecture

3 Results

We used two famous datasets for training and fine tuning the models. The COCO (Common Objects in Context) dataset is a widely used and comprehensive benchmark in computer vision, specifically designed for *object recognition* and *image captioning* tasks. Developed by Microsoft, the dataset contains a diverse collection of images, each annotated with object instances, key points, and elaborate captions. The images in the COCO dataset depict complex real-world scenes with multiple objects, diverse backgrounds, and various contextual relationships. This richness makes COCO particularly valuable for evaluating the performance of computer vision models in understanding and interpreting intricate visual scenes.**[see the figure][reference to the paper]** The Flickr8k dataset is a widely utilized benchmark dataset in the field of natural language processing and computer vision, specifically

designed for image captioning tasks. Comprising 8,000 images sourced from the photo-sharing platform Flickr, the dataset is accompanied by descriptive captions for each image. Each image in the Flickr8k dataset is associated with multiple human-generated sentences that succinctly describe its content. This dataset has played a crucial role in advancing research on image captioning algorithms, allowing researchers and practitioners to develop and evaluate models that can automatically generate coherent and contextually relevant textual descriptions for a given image. The variety of scenes and objects captured in the images, coupled with the diversity of the associated captions, makes Flickr8k a valuable resource for training and testing image captioning models, contributing to the development of sophisticated multimodal AI systems.**[refrence to flickr8k dataset]**

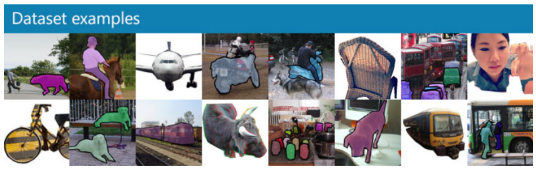

Figure 6: dataset examples

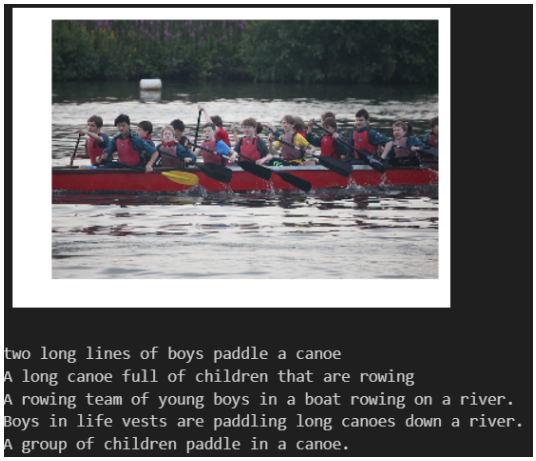

Figure 7: dataset examples

3.1 Heterogeneous architecture

we trained our network on COCO dataset with 15000 samples which are randomly chosen from *train2014* of the COCO dataset.We trained the network for 5 epochs and the benchmarks are mentioned in [the table].

Benchmarks		
Train loss	Test loss	Learning rate
1.7341	**3.6542**	0.0005
2.9631	4.0251	0.001
3.5412	9.6758	0.01

Table 1: CNN-RNN network result

3.2 Homogeneous architecture

We used flickr8k dataset for training the network. We trained the model one epoch on all images and captions in dataset and benchmarks are mentioned.

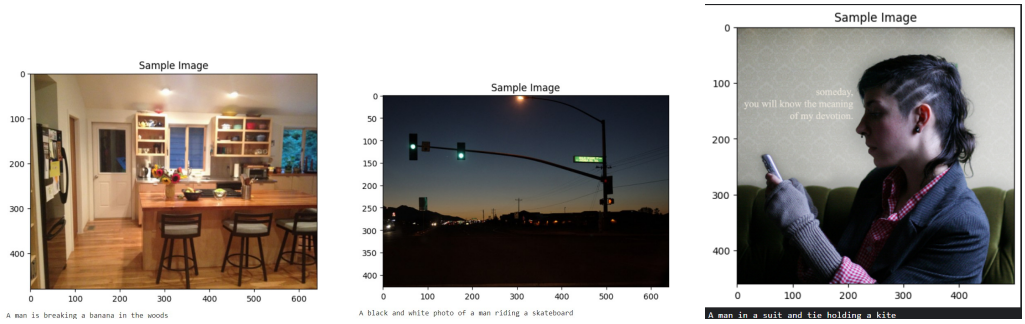

Figure 8: example outputs for CNN-RNN network

Benchmarks				
Train loss	Validation loss	Rouge2 Precision	Rouge2 Recall	Rouge2 Fmeasure
3.149400	2.938128	0.011200	0.132000	0.020500

Table 2: VIT-GPT2 network benchmarks

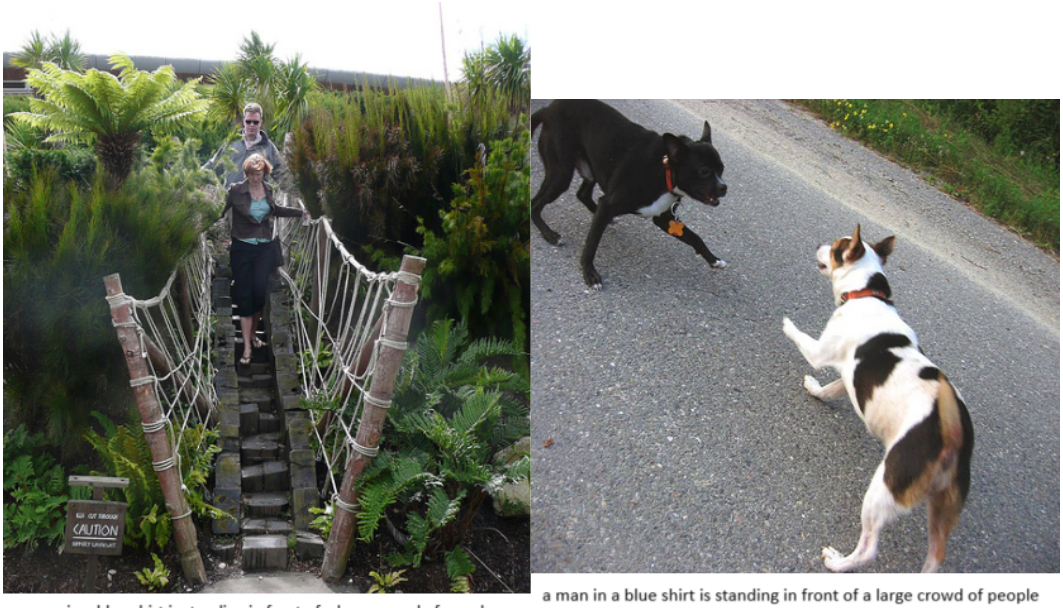

Figure 9: example outputs for VIT-GPT network

4 Discussion

The comparative analysis of two architectures, heterogeneous and homogeneous, for image captioning reveals distinct performance characteristics. The heterogeneous architecture faces challenges in generating suitable captions for images even after 5 epochs and with increased data. However, it demonstrates potential for improvement with additional computational resources, eventually converging to better loss and generating enhanced captions. Notably, this improvement is predominantly observed in the embedding layer of the visual encoder, as the gradient struggles to backpropagate effectively to the visual encoder. Contrastingly, the homogeneous architecture exhibits advantages in the image captioning task. With more computational resources and extended training epochs, it tends to outperform the heterogeneous counterpart. This superiority is attributed to the homogeneity of the network, allowing for end-to-end training. In this unified architecture, both the encoder and decoder are seamlessly integrated, enabling efficient backpropagation of gradients throughout the entire model. This cohesiveness contributes to the homogeneous architecture's ability to leverage additional computational resources for more substantial improvements in performance. Moreover, the

homogeneous architecture benefits from the inherent strength of its encoder and decoder components. Both the visual encoder and language decoder in the homogeneous architecture are more powerful compared to their counterparts in the heterogeneous architecture. This increased power enables the homogeneous model to capture and understand intricate relationships within the data, resulting in superior image captioning performance. In summary, the homogeneous architecture, with its unified design and potent components, emerges as a more promising candidate for image captioning tasks, particularly when provided with ample computational resources and extended training periods.

References

5 Further experiments

5.1 Persian image captioning

In order to improve our idea, we tested our architecture on Persian language. we used VIT as visual transformer for encoder and Pars-Bert language model as decoder.

5.1.1 Network architecture

We saw the architecture of VIT in [3-2-2]. for decoder we used ParsBERT model. ParsBERT is a state-of-the-art pre-trained language model designed for natural language processing tasks in Persian (Farsi), the official language of Iran. Developed based on the BERT (Bidirectional Encoder Representations from Transformers) architecture, ParsBERT is trained on a large corpus of Persian text, enabling it to capture bidirectional contextual information and semantic relationships within the language. The model has demonstrated exceptional performance across various downstream tasks such as sentiment analysis, named entity recognition, and text classification. ParsBERT's effectiveness lies in its ability to understand the intricacies of Persian language structure, making it a valuable resource for researchers and practitioners working on Persian language processing applications.

5.1.2 Decoder architecture

ParsBERT, an advanced pre-trained language model for Persian, is built upon the BERT architecture, a transformer-based model. The architecture involves a multi-layer bidirectional transformer encoder, where each layer has self-attention mechanisms, allowing the model to consider contextual information from both directions within a given sequence. ParsBERT incorporates a token embedding layer to represent input tokens, a position embedding layer to capture the sequential relationships, and segment embeddings for distinguishing between different segments in the input. The model is pre-trained on a large corpus of Persian text using masked language modeling, where it learns to predict missing words within sentences.

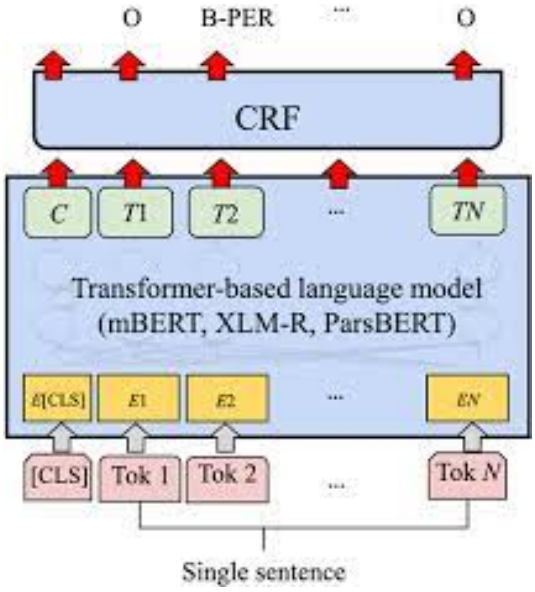

Figure 10: ParsBERT model general architecture

5.2 Result

5.2.1 Dataset

We use flickr8k-persian dataset which is translated version of original flickr8k dataset that we used in previous sections for training English image captioning model. We train the network with 4 epochs and the benchmarks are mentioned. As we can see in the result, the model converges to better loss and it also could generate better captions with more computational resources and extended epochs.

Benchmarks		
Step	Training loss	Validation loss
3500	3.8275	3.363113
17500	2.9009	2.969944
31500	2.6305	2.892444

Table 3: VIT-ParsBERT network benchmarks

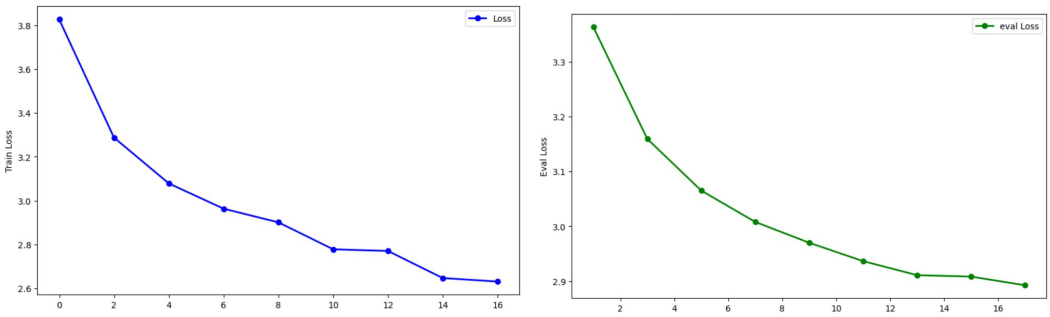

Figure 11: Train and eval loss for VIT-ParsBERT

Figure 12: example outputs for VIT-ParsBERT network