**Machine Learning for statistical NLP: Advanced LT2326**

**Visual Question Answering with Image Captions**

**Eleni Fysikoudi**

**University of Gothenburg**

**Abstract**

Natural language processing and computer vision are prevalent machine learning tasks that are thoroughly researched. Their interdisciplinary field of multimodality is a current topic that explore how to effectively integrate and interpret information across different data types (e.g., text, images, and audio) in a unified manner. There are many real-life applications that involve this integration. In this project, I endeavored to delve into the concept of multimodality, aiming to deepen my understanding of how different data types interact within machine learning models. To achieve this, I focused on developing and fine-tuning existing models to see how they could be adapted for multimodal tasks. Even though my models were not successful, they bestowed me with experience working with more complicating and state-of-the-art machine learning challenges.

1. **Introduction**

Large language models nowadays can successfully perform a vast variety of machine learning and real-life tasks. For that reason, the focus the last couple of years has been on improving them and molding them to be used in more and more tasks. Nevertheless, the repercussions of their extensive use have also been divulged and discussed extensively. Their impact on the environment has come to focus and there are many researchers who are endeavoring to develop small models that manage to perform the same tasks with less computational cost. My initial naïve idea was to follow on those footsteps and create model architectures that deal with the task of visual question answering. My motivation for this project was to not only to develop a sustainable application but also one that has a cause. That is to be used by visually impaired people in their every day life. My goal was creating an open-ended question answering model by leveraging image captions to provide additional context to prosperously generate open-ended answers. Thus, I endeavored to break it down to two tasks. First, create an image captioning model and secondly integrating it in the bigger VQA model. I will try to explain my reasoning and the steps I took to generate these models. I was ambitious enough to think I could succeed in doing this without pretrained models. However, I resolved to using them because I overestimated what I can succeed with the less complicating models. Yet, my results were also far from impressive.

1. **Background**

Image captioning and visual question answering are both two well established machine learning tasks. Both tasks involve both NLP and CV.

* 1. **Image Captioning**

Image captioning has been studied in quite detail and quite successfully. There have been different approaches to this task but I believe that the less computationally expensive solution is creating a single model that takes images as input and generates captions. That’s what my project also tried to achieve. For this part of the project, I opted for the Microsoft COCO captions dataset that can be accessed from HuggingFace. “The MS COCO caption dataset contains human generated captions for images contained in the Microsoft Common Objects in COntext (COCO) dataset” (Xinlei Chen, Et al.) Specifically, for each image there are 4 descriptions some more detailed than others as it is natural. People perceive images differently and consider which of their details are important enough to be mentioned. The captions are already lowercased and tokenized for the most part. Thus, the dataset can be regarded as very well-organized, clean and not noisy.

* 1. **VQA**

VQA is a considerably challenging machine learning task that can be characterized as yet to be solved to a satisfactory level. There have been disparate approaches for how to deal with it. My idea, not a novel one, was as mentioned before to be combined with image captioning for further addition of contextual information. The idea was to endeavor to create a model that takes as input the image, the caption and the question and returns an open-ended answer. For this second part of the project, the dataset utilized was also taken from HuggingFace which provided a subset of the official one. The reason why the subset from HuggingFace was chosen rather than the original is that figuring out how to use the VizWiz API seemed like a challenge to me in comparison to how easily accessible was the one used. The dataset is called VizWiz and is part of an ongoing competition that started in 2023. The dataset differs from the rest of the VQA datasets in that it was developed by blind people taking photos with their mobile phone and asking questions. Some implications of the way it was construed, are the fact that some images have bad quality, the questions are sometimes unrelated to the images because of errors, quite often unanswerable and finally they are more conversationally styled. These unique features increment the difficulty of an already challenging task. The already existent algorithms struggle with it so new more complex ones should be developed.

1. **Methodology**
   1. **Image Captioning**

To begin with the preprocessing of the data, there were two methods that were adopted and experimented with. The first method was creating a dataset with all 4 of the captions tokenized with padding to the max length of 100 tokens. The second method involved removing all but the very first caption for each image with padding to max length of 30 tokens. In both methods Bert Tokenizer is utilized. As far as the images are concerned, in both experiments they were initially converted to colored ones as some were black and white and then resized to dimensions of 224 x 224 and loaded to the dataset like tensors. The goal was to make them smaller to enable the model to learn as much as possible but without shrinking it to the point where information is lost. I thought that 224 is a good medium that would work well with the model.

Regarding the model architecture, the model consisted of two essential parts a CNN and an RNN, in particular an LSTM. To be more specific, the CNN part of the model has 3 convolutional layers with 32, 64, 128 out features respectively without padding, followed by batch normalization and a rectified linear unit (ReLu) activation function as well as max pooling to decrease dimensionality. It’s then being flattened to a one-dimension vector and passed through a fully connected linear state. There is also a dropout function before the linear layer. The linear layer with the features of each image is then passed to the LSTM alongside the embedding learned with a simple Pytorch embedding layer of each vocabulary token. During training, the captions and the embedding layer are concatenated along the feature dimension to form a combined input of shape batch size, sequence length and CNN output size plus embedding size. The LSTM then processes this concatenated input to generate sequential outputs, which are mapped through a fully connected layer to predict the next word in the caption. In the testing, since we don’t provide the captions instead we feed the start token to the LSTM to generate the caption based on the image features and the vocabulary. The LSTM has 3 layers as far as depth and hidden size of 512 and learning rate of 0.003, the embedding size is 256 and the loss used is cross entropy loss since generation is the goal.

* 1. **Results of IG**

Unfortunately, the model that was created did not perform as expected. To begin with, the computational cost was far bigger than expected as the model took a lot of ours to train. In both experiments (all captions vs one caption), after training for about 100 epochs the generated descriptions consisted of a lot of repeated words or words unrelated to the images. To evaluate, I first used images from the training just to see the progress of the model. Two techniques were used for the prediction of the next word. Initially, the most probable next word was chosen however that was usually words such as “a”, “and”, “of” which are function words or sometimes the word “man” which I suspect is a common word in the dataset. The second technique was k-most probable and specifically the 5 most probable predictions and one was chosen randomly. The outcomes were a bit better as verbs appeared and the correlation of image and description was more accurate. Nevertheless, it still was not as good as expected. I proceeded to test it with the VizWiz dataset although I already knew that it will not be able to create informative descriptions of the images. I tested around 10 images but the captions were completely unrelated to the images. In conclusion, it was clear that another solution should be found for creating image captions for the VizWiz dataset.

* 1. **VQA**

The preprocessing of this dataset involved to start removing a lot of the columns that I did not want to use. Particularly, the columns kept was the image, the question and the first possible answer out of the many that were available. What was removed, is the id and name file of the image, the answer type, the answerable column and then in this particular made dataset someone was reporting the results of some pretrained models they used which were redundant for me. The next step was adding image captions to each image. Since the model I created was not successful I resorted in using a pretrained transformer from HuggingFace specifically vit-gpt2-image-captioning. As expected, the descriptions of the images are quite good. However, not always accurate and very often unrelated to the images. Especially, when it generated captions for the blurry and noisy images, as GPT usually does, there was just made-up captions completely wrong. Furthermore, a lot of the questions were related to OCR like what brand of chips is this and the caption was something along the lines of a bag of chips so not much extra information. As far as the model architecture is concerned, as already discussed my intention was to construe new models. However, at this point in the project and considering the difficulty I gave up on that plan I just desired to have a worthy output. Therefore, the VQA model was comprised of a pipeline of pretrained models each with an imperative function. The first part of the model, is a vision transformer (ViT) base model which aims to extract features from the images which are then decreased in dimensions by being passed through a fully connected linear layer. Secondly, the BERT model is fine tuned and used to encode the input question, and caption into an embedding. Both are created separately and passed again through a linear layer to be correctly aligned to the same dimensions. Then, the embeddings and image features are fused together through concatenation and passed through a linear layer so they can be reduced from a 1536 to a 768 dimension vector used as input. The final component is GPT-2 model which is used as a decoder aiming to leverage the multimodal features for generating an answer. Since GPT-2 expects either input ids or already made embeddings I try to adapt the fused vector by repeating it along the sequence length dimension to match the token count in the answer sequence. This results in a tensor of shape batch size, sequence length, 768.

* 1. **Results VQA**

The model did not perform as expected. To begin with, the model seemed to be running fast which suggests that it was perhaps not very computationally expensive. The results I got while trying to evaluate the answers were mostly generations of full-stops, newlines and some random words but without having much sense. There was effort in adjusting the hyperparameters when training the model for more epochs and different learning rate but the loss was not changing. Thus, the model was not learning anything so the issue is not inadequate training.

1. **Discussion**

Taking into consideration the results of the models, someone could conclude that there might be an error in the logic of making the models, especially the VQA. Looking back now, I would like to try in the future some changes and experiments. As far as the image captioning model, I would like to try reading some more literature to find out more about how I can update the model. That is because I endeavored to implement the most straightforward method of creating descriptions and yet I did not really succeed. For the VQA model, there are many ideas as to how to make it better or find out what went wrong exactly. Perhaps, the first thing I would like to do is train and test this model with a different and cleaner dataset. I think that could provide an insight to whether the model is on its own culpable for the results or the data plays a big role in that as well. Similarly, I would want to clean the data by removing the unanswerable images just to test the same hypothesis. Nevertheless, without having done any of these experiments, there are some things that I believe play a role in the deficiency of the model. That is in the architecture of the model a few things could have been done differently. A different approach, one that is maybe more beginner or intermediate friendly is to do the visual question answering as a multiclassification task rather than open generation. The reason why I did not opt for that was that I wanted to manage out of vocabulary images and questions and avoid the restriction to the top-500 answers for instance. However, that would have worked well at least as far as yes and no questions are concerned. Moreover, I believe that a big issue behind the results of this model is the utilization of two unrelated models that is BERT and GPT. I use BERT to create the embeddings and then feed them to GPT. Nevertheless, GPT possibly makes embeddings completely disparately so it is like feeding it a foreign language that it does not understand. Therefore, using and fine tuning only GPT could bring better results. The problem with the current application which might be why the model is not performing as expected is

the way of making the embeddings. That is reshaping and repeating the combined features based on the length of the answer input sequence to get the expected input for GPT. I currently do not have an idea of how else to form the BERT embedding into GPT ones other than what aforementioned (not using BERT). Another idea is adjusting the input of the model. That is, a possible alternative would be fusing only the image features and captions while providing the question as a prompt. Maybe that could give good results even with still having both BERT and GPT (probably not).

1. **Conclusion**

To summarize, the goal of the project was to experiment and learn more about multimodal models by creating some. Although the efforts did not bloom into a beautiful flower and the goal of a successful model was not met, the experience and knowledge that I acquired is enough of a reward for now. It was foreseen that the output of the models would not be revolutionary but the results were disappointing since the models did not generate any output at all other than gibberish. There are some ideas of how to work on this further that I would like to implement in the near future.

**References**

Adedokun, O. R., & Odetunmibi, O. A. (2021). Caption generation of images using CNN and LSTM. International Journal of Innovative Research in Engineering and Management. Retrieved from <https://ijirem.org/DOC/1-caption-generation-of-images-using-cnn-and-lstm.pdf>

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. arXiv preprint arXiv:1504.00325. <https://arxiv.org/pdf/1504.00325>

Aneja, J., Deshpande, A., & Schwing, A. G. (2018). Convolutional image captioning. arXiv preprint arXiv:1802.08218. <https://arxiv.org/pdf/1802.08218>

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., ... & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. arXiv preprint arXiv:1505.00468. <https://arxiv.org/pdf/1505.00468>