NLP with Deep Learning
Project Report - Mini Project 5

Name: Poojith Reddy Maligireddy
UID: u1405749

Note:

In my initial project proposal, I outlined my intention to employ RNN with LSTM for machine translation task. However, after transformers class, I have decided to focus on utilizing transformers for the same machine translation task as they are considered as state-of-the-art models. The dataset I will be working with remains same Multi30K. Also, I would like to explore TensorFlow for this project, as we have used PyTorch in our previous mini projects. This will provide me with a good understanding of TensorFlow, as I only have a limited experience with it.

What problem did you work on? Why is it interesting and relevant to the class?:

In this project, I worked on the machine translation task using transformers. Machine translation is a NLP task that involves translating text from one language to another language (English to German in this project). Seq-to-Seq also has other applications such as text summarization, POS tagging and so on. Machine translation is a fundamental and challenging problem in natural language processing (NLP), and employing transformers for this task is particularly interesting due to their state-of-the-art performance and efficiency in handling sequential data. Transformers have revolutionized the field of NLP by introducing attention mechanisms, enabling models to capture long-range dependencies in language sequences. This makes them well-suited for tasks like machine translation, where context and understanding of the entire input sequence are crucial for generating accurate translations.

The class aims to equip students with the knowledge and skills to apply deep learning techniques to natural language understanding and generation tasks. Machine translation with transformers serves as a practical application of the concepts covered in the course, allowing for a hands-on exploration of the latest advancements in deep learning for NLP. This project is relevant to the class as it applies deep learning techniques (transformers) to a real-world NLP problem (translating text from one language to another (English to German in this project)).

What are the important ideas you explored?:

To begin with, the first idea is the transformer architecture, it is used as a base for many state-of-the-art models. This architecture uses self-attention mechanism to capture relationship among words in the sentence and also the main idea is the parallelization for efficient modelling of long-range dependencies. The attention mechanism is used to weigh the importance of different words in a sequence while processing each word and so it allows model to focus on relevant information, which in turn improves the ability to capture context.

The next important idea is the Multi-Head Attention, transformers use multiple attention heads in parallel, allowing the model to capture different aspects of the relationships between words. This multi-head attention mechanism enhances the model's ability to learn diverse patterns. The next one is layer normalization and residual connections, they are used contribute to the stability of training transformer models by normalizing intermediate layer outputs and so there is a smooth flow of information through residual connections.

Finally, the idea of using pre-trained fine-tuning models, as these models are trained on large language corpora, fine-tuning these models for specific translation tasks often leads to improved performance with less data.

What ideas from the class did you use?:

The first idea is the Encoder-Decoder Structure, where the encoder processes the input sequence and the decoder generates the output sequence one word at a time. This enhances the model’s capacity for sequence-to-sequence tasks like translation. Also, it uses positional encoding to provide information about the positions of words in a sequence to enable the model to consider the order of the words as transformers don’t understand the order of input sequence. Following that is the approach to train a Set-to-Seq model directly on parallel corpora.

Also, evaluating the machine translation system using BLEU score metric, eliminating the need for manual evaluation. However, it's important to note that BLEU scores have limitations—they rely on tokenization and are strongly influenced by specific reference translations. Consequently, well-translated sentences might receive lower scores.

Finally, effective tokenization and vocabulary are very important for machine translation. subword tokenization is used to handle languages and improve the model's generalization.

What did you learn?:

By working on this machine translation project, I gained a thorough understanding of how the transformer's encoder-decoder architecture works. I learned how to preprocess text, prepare inputs for the encoder, generate text using the decoder, and compare it with the original. I got a clear understanding of the entire pipeline, from using the encoder-decoder architecture to evaluating the model using the BLEU score metric.

After completing this project, I now feel confident in creating full pipelines for other tasks or projects using Hugging Face and experimenting with them. I also learned how to leverage state-of-the-art models and fine-tune them for specific tasks, making the most of advanced technologies in natural language processing.

By working on this project, I not only engaged with state-of-the-art technology but also gained practical insights into how transformers can be effectively applied to real-world NLP tasks.

Results and comparisons to baselines:

In this machine translation project, I have used Multi30K, a bilingual parallel text dataset (English-German), which contains 29000 training examples, 1014 validation examples and 1000 test examples and below is the link about it.

https://aclanthology.org/W16-3210.pdf

I used the model named "Helsinki-NLP/opus-mt-en-de" from Hugging Face for this machine translation task (English to German) and evaluated the model's translation performance using BLEU score metric from nltk library which ranges from 0 to 1(Higher the better).

After downloading the above pre-trained model, I fine-tuned this on this machine translation task with the above dataset. I followed the same procedure as the previous mini-project where I had trained for max 10 epochs and took the best model which has high validation BLEU score metric and calculated the test BLEU score with that best model.

After hyper-parameter tuning, I found out the best learning rate as 1e-4.
Below are the results:

The highest validation set BLEU score is 0.71.
The test set BLEU score is 0.69.

Looking at this metric, I would say that the scores mentioned above are quite satisfactory. Therefore, we can conclude that the model is performing very well when compared to baseline models (which gives almost zero BLEU score), even though we only trained it for ten epochs. This highlights the effectiveness of using pre-trained fine-tuning models, as they come with prior training on extensive datasets.

If you had much more time, how would you continue the project?:

If I had more time to continue this project, the first thing would be multi-lingual training, explore training the model on multiple languages simultaneously. This could lead to better language representations and improved translation quality. The next thing would be trying out different transformer architectures along with hyperparameter tuning. The other thing could be trying out different data augmentation techniques such as paraphrasing, adding noise to training data and back-translation in order to improve generalization.

Also, the next thing to try out could be using ensemble methods by combining predictions from multiple transformer models which could provide more robust models. Considering latest advancements, Interpretability of the model's predictions could be an interesting idea to explore. Finally, the other things could be exploring methods for learning cross-lingual representations to enhance model's ability to generalize well across different languages, fine-tuning with domain specific data if task involves specialized vocabulary and try out other different quality metrics.