Homework 3: Parsing and Transformers

Since positional information in Transformers completely relies on the positional embedding after the embedding layer. I am considering how much benefits this mechanism brings to the model, especially in context-aware required tasks like POS-Tagging and Parsing. Here, I perform experiments on the model without positional embedding, the model embeds positional information with RNN, and the original Transformers. Here I notice some significant performance variance.

1 Model Without Positional Information

There is no doubt that this method fails devastatingly in tasks that attach great importance to contextual information, like Parsing.

In this setup, the model doesn't have any explicit way of understanding the order of tokens in a sequence. It treats all tokens as if they are unordered (Bag of Words). This is a problem for tasks that attach great importance to contextual information, like parsing. For parsing, the order of words in a sentence is crucial to determine the grammatical structure. Without positional information, the model may struggle to accurately parse sentences. Figure 1 shows an inference example of this model.

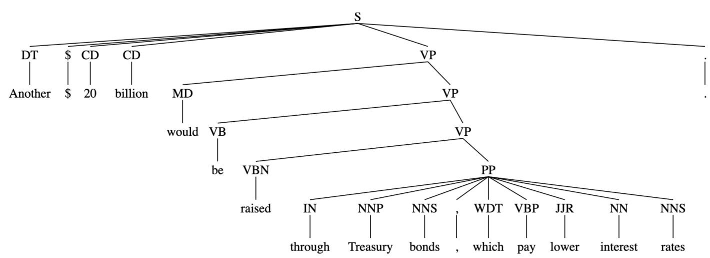

Figure 1: An Inference Example of Model Without Positional Information, in this setup, parsing fails catastrophically.

2 Positional Embeddings with RNN

With the importance of capturing positional information, I tried to apply RNN to embedding layer, aiming to capture these information, and see whether this method can work better or even has similar performance of Transformer.

Experiment results show that this mechanism does help to improve the performance of the model, which determines the importance of positional embedding in this task again. However, RNN-based method can not even have a close performance of Transformer. RNNs can capture some positional information, but they are not as effective as learned positional embeddings in transformer models, which provide consistent, non-vanishing, and highly parallelizable positional information. Figure 2 shows an inference example of this model.

3 Transformers

The original transformer model introduced positional embeddings as a crucial component to provide positional information to the model. Positional embeddings are added to the token embeddings to inform the model about the position of each token within a sequence. This is essential for understanding the

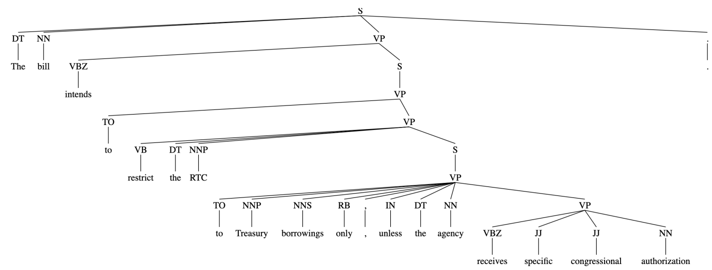

Figure 2: An Inference Example of Model Without Positional Information, in this setup, parsing fails catastrophically.

order and relationships between tokens in a sequence. Also, compared to RNN, this positional embedding method provides a highly parallelizable computational structure, significantly reduce inference and training time.

Transformers	**P**	**R**	**F1**
No positional embedding	73.86	30.31	42.98
RNN-based	78.43	45.39	57.5
Positional embedding	90.04	86.56	88.27

Table 1: Models Performance