Conference Resolution Homework 3 - NLP 2022

Mirza Mujtaba Hussain
Email: mirza.1989740@studenti.uniroma1.it

1. Introduction

Conference resolution is an essential task in Natural Language Processing (NLP) which involves identifying the entities in a text and linking them to their corresponding mentions.The goal of conference resolution is to enhance the understanding of the text by providing the context of the entities mentioned.In End-to-end conference resolution, a pipeline starts from identifying ambiguous pronoun, and than identifying entity in text and finally, conference resolution.

This project focuses on the third step of the pipeline, which is the conference resolution. The aim is to develop a model that can effectively predict the correct entity that the ambiguous pronoun refers to. This will help to improve the overall understanding of the text and enhance the quality of information extraction from the text.

2. Data Encoding

In this project, I used the BERT base model to encode the input text and generate the word embeddings. BERT base is a pre-trained deep learning model that uses the Transformer architecture to process and encode the input text. The model consists of 12 layers of transformer encoders, each producing an output for each token in the input text.The main reason behind using BERT is to get contextualized word representation.

As mentioned, BERT base model uses 12 layers of transformer encoders, each output per token from each layer of these can be used as a word embedding. The authors of who presented BERT found that summing the last 4 layers produced on of the best results for various NLP tasks. Hence, we used the sum of the last 4 layers as the embeddings in our project. This approach allowed us to capture the most relevant and contextual information from the input text, which is crucial for our task.

While training, I also experimented replacing BERT embeddings with SpanBERT embeddings which is one of the many variation of BERT.The SpanBERT is designed to better represent and predict spans of text.It extends BERT by (1) masking contiguous random spans, rather than random tokens, and (2) training the span boundary representations to predict the entire content of the masked span, without relying on the individual token representations within it.

3. Training

In the training process, the BERT embeddings were fed into a Bi-LSTM architecture consisting of two layers.While training the layers of BERT model are frozen. This is done to speedup the training and since the data used is not huge, training the layerrs of BERT would not really impact the results significantly. The outputs of the Bi-LSTM for each word in the sentence were averaged to obtain information about the overall sentence. Also ,the output corresponding to the ambiguous pronoun and the candidate entities 'A' and 'B' were then selected. These outputs were fed into a Multi-layer Perceptron (MLP) to make predictions.

During the training process, the output from the Bi-LSTM for the ambiguous pronoun and the Bi-LSTM for the two candidate entities (A and B) was explored to determine the best method for feeding the information into the MLP. Two approaches were tested. The first approach involved directly feeding the output from the Bi-LSTM for the pronoun and the Bi-LSTM for the two candidate entities into the MLP. The second approach involved subtracting the output of one candidate entity (A) from the output of the pronoun and dividing the result by two, and doing the same for the other candidate entity (B), before feeding the results into the MLP. The results of these experiments showed that both approaches produced similar results.

While training the Adam optimizer was utilized to update the model parameters. Dropout was also employed as a regularization technique to prevent overfitting. Also the training data was split into training and validation part.

4. Results

During this project, I used scikit-learn to calculate the f1-score and precision respectively on a separate test set. When evaluating the results, I found that using BERT embeddings performed slightly better as compared to using embeddings from SpanBERT and were faster to converge. Here is a table describing f1-score for each class, i.e when the pronoun was mentoining to A, B or none.

Model	A	B	None
Bert	0.79	0.79	0.59
Bert[*]	0.81	0.80	0.56
SpanBert	0.78	0.80	0.53
SpanBert[*]	0.77	0.79	0.55

The [*] refers to results using output from bilstm for A or B and subtract them with output of pronoun and divide by 2

5. Conclusion

In this project, we explored the use of BERT embeddings in conjunction with a Bi-LSTM architecture for conference resolution. The results showed that this approach performed well and was able to somewhat effectively identify the correct entity a pronoun referred to. The use of the last 4 layers of BERT was found to be a good choice for encoding the input text.

Additionally, experiments were conducted to compare the performance of BERT and SpanBERT embeddings. Although it was expected that SpanBERT would perform better than BERT, the results from all experiments were mostly comparable.