Subject: Large Language Model
Name: Dipen Prajapati
Net ID: dp1435

Project Topic:

Create a Contextualized Word Embeddings using existing pretrained Word Embeddings

Introduction

This project report presents the context word embeddings implementation and comparison against the ELECTRA model. Also, the performance comparison between BERT and ELECTRA models is carried out to evaluate their speed and efficiency in generating embeddings.

Part 1: Contextual Word Embedding with Same Word, Different Meaning

Objective

Visualizing context word embeddings in defining the variation of the same word employed in alternate contexts based on a comparison of embeddings developed by the ELECTRA model.

Methodology

1. **Loading the Model and Tokenizer:**

    - The ELECTRA model and tokenizer are loaded from the transformer library.

2. **Input Sentences:**

    - Give two input sentences for which wants to check context of same word relation with other word embedding.

3. **Generating Embeddings:**

    - Tokenize and translate the sentences to tensors.
    - Model generates embeddings without computing gradients (torch.no_grad()).

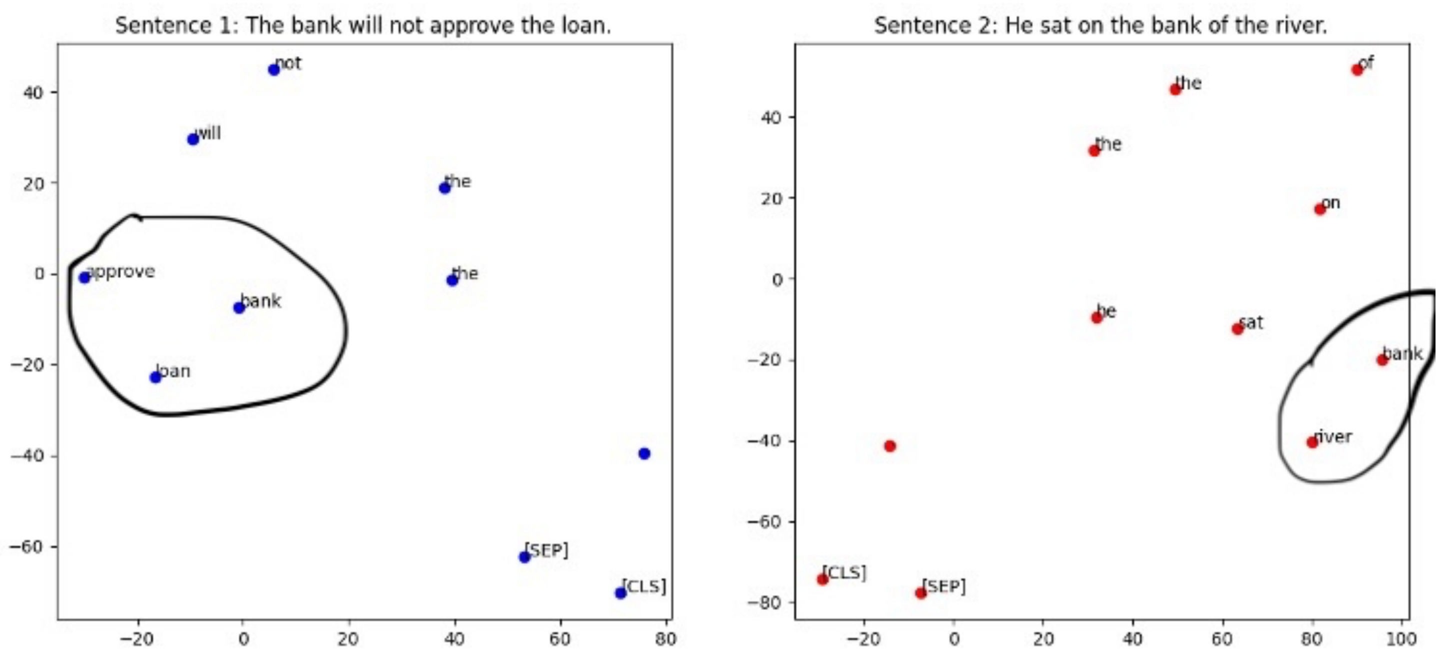
4. Token embeddings extraction and Conversion to Numpy:
    - Extracts the embeddings and converts them to NumPy arrays for visualization.
5. **Dimensionality Reduction (t-SNE):**
    - High-dimensional embeddings are mapped into 2D space for visualization.
6. **Visualization:**
    - Token embeddings of both sentences are plotted to examine how well the model can distinguish between different senses of a particular word.

Results & Analysis

- **Input Sentences:**
    1. "The bank will not approve the loan."
    2. "He sat on the bank of the river."

Input sentences can be anything. To check model performance, use same word with different context or meaning in different sentences.

- The embeddings generated by ELECTRA clearly separate the word **'bank'** in two different contexts: financial institution (sentence 1) and riverbank (sentence 2).
- The visualization plot shows distinct clusters for the word 'bank' based on its meaning context.



Part 2: Paragraph Testing - BERT vs. ELECTRA

Objective

To compare the speed and effectiveness of BERT and ELECTRA models in generating contextual embeddings for a paragraph.

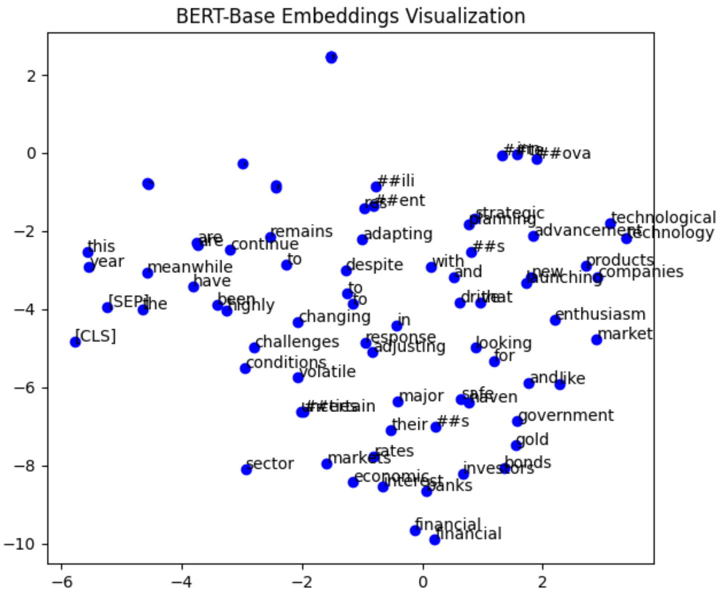
Methodology

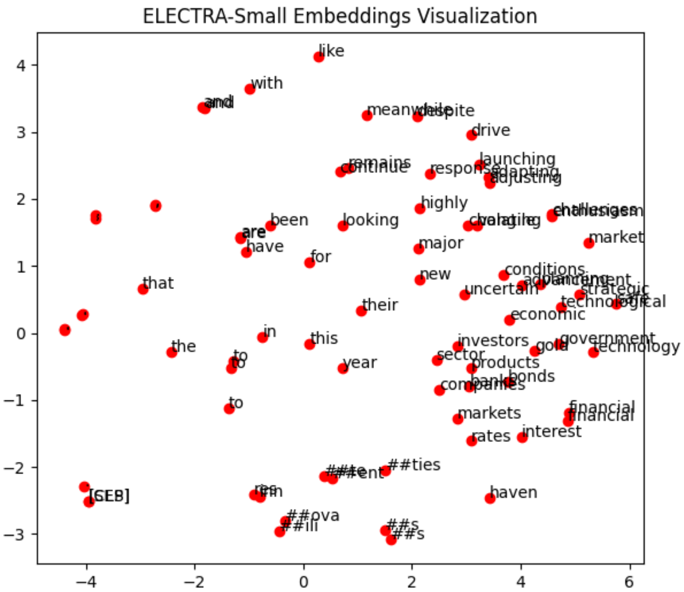
1. **Model Loading:**
    - BERT-base and ELECTRA-small models are loaded using the transformers library.
2. **Device Configuration:**
    - The models are run on GPU.
3. **Embedding Generation:**
    - Embeddings are generated for the same paragraph using both models and measuring time taken by each model is recorded.
4. **Dimensionality Reduction (t-SNE):**
    - Embeddings are reduced to 2D space for visualization and its comparison.
5. **Visualization:**
    - Comparison plots are generated to visualize the embeddings generated by BERT and ELECTRA.

Results & Analysis

- **Input Paragraph:** take any paragraph for embeddings.
- **Output Timing:**
  BERT-Base Time: 1.4768 seconds
  ELECTRA-Small Time: 0.2584 seconds
- **Observations:**
    - ELECTRA is faster than BERT in generating embeddings due to its more efficient architecture.

- Both models produce well embeddings, but ELECTRA produces equally good embeddings at a much faster rate.





Conclusion

When comparing the BERT and ELECTRA models for contextual word embeddings, it was shown that ELECTRA can effectively generate meaningful word embeddings in a significantly shorter amount of processing time. It was shown that the models could easily distinguish between multiple contextual meanings of a single word using t-SNE in embedding visualization.

For real-time applications where processing time is crucial, ELECTRA is faster in speed makes it an appealing option. However, further research is required to evaluate ELECTRA's performance in comparison to BERT on multilingual tasks. Future studies will compare these models' performance in this area by applying them to sentiment analysis of financial news.

The code and results are attached here below in this pdf.