Chaklam Silpasuwanchai, Todsavad Tangtortan
AT82.05 Artificial Intelligence: Natural Language Understanding (NLU)

A5: Sentence Embedding with BERT

In this assignment, we will emphasis on leveraging text embeddings and capturing semantic similarity using a powerful encoder like BERT.

Note: You are ENCOURAGED to work with your friends, but DISCOURAGED to blindly copy other's work. Both parties will be given 0.

Note: Comments should be provided sufficiently so we know you understand. Failure to do so can raise suspicion of possible copying/plagiarism.

Note: You will be graded upon (1) documentation, (2) experiment, (3) implementation.

Note: This is a one-weeks assignment, but start early.

Deliverables: The GitHub link containing the jupyter notebook, a README.md of the github, and the folder of your web application called 'app'.

Task 1. Training BERT from Scratch with Sentence Transformer - Based on Masked Language Model/BERT.ipynb, modify as follows: (1 points)

1) Implement Bidirectional Encoder Representations from Transformers (BERT) from scratch, following the concepts learned in class.
2) Train the model on a suitable dataset. Ensure to source this dataset from reputable public databases or repositories. It is imperative to give proper credit to the dataset source in your documentation.
3) Save the trained model weights for later use in Task 2.

Task 2. Sentence Embedding with Sentence BERT - Implement pretrained BERT network that use siamese network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. (3 points)

1) Use the SNLI [1] or MNLI [2] datasets from Hugging Face, or any dataset related to classification tasks.
2) Reproduce training the Sentence-BERT as described in the paper [3].
3) Focus on the Classification Objective Function:

$$o = \text{softmax}\left(W^T \cdot (u, v, |u - v|)\right)$$

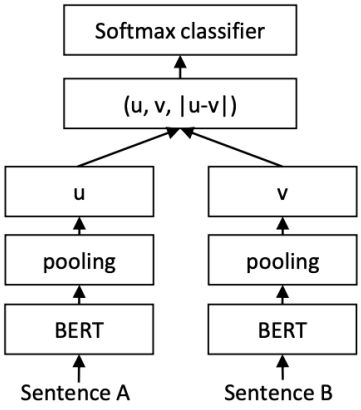

Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

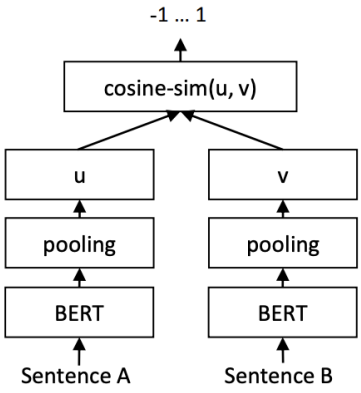

Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

[1] https://huggingface.co/datasets/snli
[2] https://huggingface.co/datasets/glue/viewer/mnli
[3] https://aclanthology.org/D19-1410/

Task 3. Evaluation and Analysis (2 points)

1) Perform a detailed evaluation of your sentence transformer model, considering different types of sentences and relevant metrics.
2) Compare the performance of your model to other pre-trained models and discuss any observed differences.
3) Analyze the impact of hyperparameter choices on the model's performance.
4) Discuss any limitations or challenges encountered during the implementation and propose potential improvements or modifications.

Note: Make sure to provide proper documentation, including details of the datasets used, hyperparameters, and any modifications made to the original models.

Task 4. Text similarity - Web Application Development - Develop a simple web application that demonstrates the capabilities of your text-embedding model. (3 points)

1) Develop a simple website with an input box for search queries.
2) Utilize a custom-trained sentence transformer model
3) Implement a function to calculate the cosine similarity using the sentence embeddings generated by the sentence transformer.

As always, the example Dash Project in the GitHub repository contains an example that you can follow (if you use the Dash framework).

Good luck :-)