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

A1: Engine Search

This assignment focuses on creating a search engine, a practical application of information retrieval in natural language processing. The search engine, deployed on a website, should return the top paragraphs with the highest similarity to a given query, such as "Harry Potter." This task will involve building upon existing code, understanding and implementing word embedding techniques, and creating a web interface for the search engine.

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. Preparation and Training - Build upon the code discussed in class. Do not use pre-built solutions from the internet.

1) Read and understand the **Word2Vec**[1] and **GloVe**[2] papers.
2) Modify the Word2Vec code from the course GitHub repository: (3 points)
   - Train using a real-world corpus (not too large). 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.
   - Use a window size of 2.
   - Implement the Continuous Bag of Words (CBOW) model.

Task 2. Model Comparison and Analysis

1) Compare Skip-gram (with and without negative sampling), CBOW, and GloVe models on syntactic and semantic accuracy, similar to the methods in the GloVe paper[3]. (2 points)
2) Use the similarity dataset[4] to find the correlation between your models' dot product and the provided similarity metrics. Assess if your embeddings correlate with human judgment. (2 points)

Note: Do not be surprised if you achieve 0% accuracy in these experiments, as this may be due to the limitations of our corpus. If you are curious, you can try the same experiments with a pre-trained GloVe model from the Gensim library for a comparison.

Task 3. Search Engine - Web Application Development - Develop a simple website with an input box for search queries. (3 points)

1) Implement a function to compute the dot product between the input query and a set of texts (e.g., Wikipedia texts), and retrieve the top 10 most similar passages.
2) You may need to learn web frameworks like Flask or Django for this task. Use readily available texts, such as from Wikipedia, as your search corpus.

Best of luck in developing your search engine!

[1] https://arxiv.org/pdf/1301.3781.pdf
[2] https://aclanthology.org/D14-1162.pdf
[3] https://nlp.stanford.edu/pubs/glove.pdf
[4] http://alfonseca.org/eng/research/wordsim353.html