Assignment 2: Word2Vec and GloVe Embeddings (15 points)

CS 410/510 Natural Language Processing, Fall 2023

In this assignment, you will explore `word2vec` and `glove` embeddings for text classification using the Gensim library. The goal is to build yet another text classifier that will predict whether a piece of text is "positive" or"negative", but this time, instead of engineering the features (unigrams, bigrams, sentiment words, etc.), you will use pretrained embeddings from the `word2vec` and `glove` models.

1 Instructions

1. **(same as last assignment) Data Preparation**: You are provided with a dataset containing customer reviews of products. The dataset is the "Multilingual Amazon Reviews Corpus" (1) in json format. It includes several columns. For this assignment, we will be using a small subset of the original dataset and are concerned with the following two columns:

   (a) "review_title" (the title of the review), and

   (b) "stars" (an integer 1 or 5 indicating the number of stars, which is considered as the sentiment label, with 1 indicating "negative" and 5 indicating "positive").

   Warning: *As is the case with most 'natural language' text, this data was collected from public websites and is mostly unfiltered. Therefore, it is possible that some text may be disturbing or you may not agree with it.*

   - The dataset is pre-split into a training set and a test set.
   - Load the dataset using Python and the appropriate library (e.g., pandas).
   - (optional) Perform any necessary data preprocessing steps, such as lowercasing, tokenization, and removing punctuation. The preprocessing will be more effective if it matches the preprocessing that was applied to the training corpus of the pretrained models.

2. **Pretrained Models** As part of this assignment, you will experiment with two specific pretrained models: `word2vec-google-news-300` and `glove-wiki-gigaword-300`. These can be downloaded using the gensim library[1].

[1] `https://radimrehurek.com/gensim/models/word2vec.html#pretrained-models`

- Comparison: Briefly describe the differences between the two pretrained models.
- Visualization: Using tools like t-SNE[2] reduce the dimensionality of the embeddings, visualize the following and discuss your results:
  (a) emotion words such as 'happy', 'sad', 'angry', 'joy', 'love', 'fear', etc. to observe how these emotions are positioned in the vector space.
  (b) gender and occupation (e.g., 'man', 'woman', 'king', 'queen', 'doctor', 'nurse', 'engineer', 'teacher', etc.) to explore biases and stereotypes in embeddings.

3. **Text Classification Model**: Now, it's time to build a text classification model. Implement a simple text classification model using any machine learning library of your choice (e.g., scikit-learn or NLTK). Follow these steps:
   - Choose *any one* suitable algorithm for text classification (e.g., Naive Bayes, Logistic Regression, or a basic neural network).
   - Create word representations of your text data (convert text into numerical features) using pretrained embeddings.
   - Train the classification model using the training set.
   - Evaluate the model's performance on the test set using appropriate evaluation metrics (e.g., precision, recall, F1-score).
4. **Results and Analysis**: Provide a detailed analysis of your model's performance by comparing the output of the two pretrained embeddings, as well as how the results compare with your previous assignment. Include the following:
   - F1-score and other relevant metrics.
   - Any observed challenges or limitations.

2 Submission Guidelines and Grading

Submit a PDF of your Colab notebook on Canvas, under "Assignment 2". Include any additional files or resources used. Ensure your code is well-documented and organized.

To convert your Colab notebook to PDF format, you'll need to install a Python package called nbconvert, a command-line tool that allows you to convert Jupyter notebooks to various formats. To install nbconvert, open a new code cell in your Colab notebook and run the following command:

```
!pip install nbconvert
```

Once nbconvert is installed, you can convert your Colab notebook to PDF format using the following command (replace <notebook-name> with the name of your Colab notebook):

```
!jupyter nbconvert —to pdf <notebook—name>.ipynb
```

[2] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

Your assignment will be assessed based on the following criteria:

- comparison (2 points),
- visualization and discussion (4 points),
- implementation of the text classification model (4 points),
- model evaluation and analysis (3 points),
- and overall clarity and organization of the assignment (2 points).

Good luck!

References

[1] Keung, Phillip, Yichao Lu, György Szarvas, and Noah A. Smith. "The multilingual amazon reviews corpus." arXiv preprint arXiv:2010.02573 (2020).