Report: Sentiment Analysis Using Various Embeddings

Introduction:

This assignment is on sentiment analysis of Twitter data during the stock market crash of 2022. Data collection was done using the hashtag #stockmarketcrash. It is divided into three categories. I have used different embedding techniques of sentiment analysis and the classification model Support Vector Machine (SVM), which will tell us which embedding technique will give the best accuracy. The different embedding techniques include BoW, TF-IDF, Word2Vec, GloVe, and USE.

Methodology

1. Preprocessing Steps:

Text data preprocessing steps shall be performed to clean and normalize the text to be analyzed, and they are as follows:

Lowercasing: This converts all text into its lower case to avoid case-sensitive variations. Removing **Mentions, URLs, and Hashtags:** These are removed since none of them contribute anything toward the sentiment of the tweet.

Removing Special Characters and Numbers: Non-word characters and numbers are removed.

Lemmatization: This reduces each word to its base form, called the lemma, using WordNetLemmatizer and part-of-speech tagging to make sure words are lemmatized according to their part of speech.

Example of Preprocessing:

Original: "Loving the stock market rally today! #bullish @user"

Preprocessed: "love the stock market rally today"

2. Embedding Techniques:

Text is embedded into numerical vectors through five different approaches:

Bag of Words (BoW):

Frequency-based method that counts the occurrences of each word in the text. Dimensionality reduction followed with the chi-square test and picked only the top 100 features.

TF-IDF:

Similar to BoW, with an improvement to give less frequent but important words higher weights. A chi-square test was used to choose the top 100 features.

Word2Vec:

Pre-trained word embeddings (word2vec-google-news-300) that are trained from context in text capture word semantics.

GloVe:

Pre-trained word embeddings from the GloVe model, glove-wiki-gigaword-300, which captured co-occurrence patterns between words.

Universal Sentence Encoder (USE):
Sentence-level embedding that provided a dense, 512-dimensional representation of the whole sentence to identify semantic structure and context.

3. Model Implementation

Support Vector Machine: SVM is chosen to be used for classification, for the following reasons:
It is ready-to-use with high-dimensional text data and works well with sparse vectors returned by techniques such as BoW and TF-IDF.
A linear kernel was used for the SVM in all experiments to keep results consistent.

Data Split:
Data was split such that 80% formed the training set and the remaining 20% was kept as the test set for the evaluation of the model performance to ensure that results evaluated were on unseen data.

Model Evaluation:
How each of these embedding techniques contributes in this regard, when considered in terms of model performance on test set accuracy.

Findings

1. Summary of Model Performance

The chart below shows the accuracy of the SVM classifier for various embedding techniques:

Embedding Type	Accuracy
BoW	0.7150
TF-IDF	0.7206
Word2Vec	0.6500
GloVe	0.6650
Universal Sentence Encoder	0.6800

2. Analysis and Discussion

BoW: It performed considerably well since BoW captures word frequencies, which can be helpful toward sentiment classification. It does not consider the semantic meaning or word order, however.

TF-IDF: This is the most successful performing method, giving more importance to those rare but informative words like "crash" and "soar," which usually carry significant sentiment. Chi-square selection fine-tuned the feature set further; this likely contributed to the excellent performance of this model.

Word2Vec and GloVe: Both methods capture the semantic relationships between words; both are, however, word-level embeddings. They do not take into consideration the structure and context of the sentence, which may make all the difference in sentiment analysis. This could be a reason for their relatively poorer performance compared with BoW and TF-IDF.

Universal Sentence Encoder: USE outperformed Word2Vec and GloVe since it captures the context of the whole sentence rather than relying on the embedding of individual words. However, the model probably would need more fine-tuning or trying different kernels in the SVM to get better results.

3. Example to Support Findings Consider the example below:

Tweet: "The stock market crash is devastating!"

BoW: It may probably center around the words "crash" and "devastating," and classify this tweet as negative.

TF-IDF: Similar to BoW, but gives a higher weight to rare words, such as "devastating," thus correctly classifying it as negative.

Word2Vec and GloVe embeddings might capture the meaning of words individually, but may fail at grasping the context of the complete sentence; hence, could lead to misclassifications.

Conjecturally, **USE** would capture the full meaning of the sentence and accurately categorize it as negative. Unfortunately, in this experiment, its accuracy turned out to be somewhat lower than TF-IDF's. That implies that to unlock the full power of USE, more advanced modeling techniques are required, which might come in the form of a deep learning-based classifier.

Conclusion

TF-IDF fared better than the embedding techniques on this dataset, with a maximum accuracy of 72.06%. It performed well over other typical word embeddings like Word2Vec and GloVe, along with the Universal Sentence Encoder.

TF-IDF: It worked best for this dataset due to the emphasis it places on the presence of important words that often carry sentiment.

Bag of Words: Performed well but does not provide context.

Word2Vec and GloVe: They did pick up word semantics but failed to provide the overall sense of the sentence.

Universal Sentence Encoder: It did well in this experiment but most probably requires more sophisticated models to realize its full potential.