NLP ASSIGNMENT 02 REPORT

Name: Jude Damian Sequeira

Student ID: 220431413

INTRODUCTION

Semantic similarity is an important aspect of Natural Language Processing and one of the fundamental problems for many NLP applications and related disciplines. It measures the similarity between texts/documents and this feature is used in many applications using Semantic Analysis in which similar product/company names are identified to compare the products and services with the competitive products or services in the market. Semantic Similarity is also used in Plagiarism Software's to detect similar texts even if the words are paraphrased.

This report proposes a methodology to use Semantic Similarity of dialogue texts of the UK show EastEnders and retrieve the most similar document vectors between the characters of the show. Various Pre-processing and feature selection methods are applied on training data and compared until the best similarity score (in terms of mean rank and accuracy) for validation set is found. The best model is then tested on the unseen test data to find the most similar dialogues.

EXPERIMENTS

The training dataset has approx.15000 Character dialogues from the EastEnders TV show which consists of attributes such as Episode, Scene, Scene_Info, Character_Name, Line and Gender. For purposes of avoiding the influence of the unbalanced datasets since the number of dialogues of each character are unequal, the training data uses only the first 400 lines of each character whereas for the test set uses only the first 40 lines. The Train-Validation split of 90:10 is considered.

The initial code which only uses the dictionary vectorizer of counts of each word without any pre-processing achieves mean rank and accuracy of 4.5 and 25% resp. for the validation set, whereas for the test set it achieves mean rank of 5.12 and accuracy of 31.25%. The aim is to achieve best mean rank value close to 1 by performing the experiments below.

1) For the first experiment(question1-3), the pre-processing stage is included, which takes the dialogue text and performs text lowercasing, punctuation removal, tokenisation, digit removal, Stop-word removal and Lemmatization. For this experiment the lemmatization *'pos'* parameter was given input as 'v' which means the lemmatizer function will only convert verbs to its base form. Example: "eating" will lemmatize to "eat".
   The pre-processed tokens were then converted to a feature vector dictionary, which had unigram keys with its Parts-Of-Speech (POS) tag as features along with its feature counts in the document vector. The POS tags accounts for sentence structure in dictionary as opposed to BOW model which does not take word position in sentence into account. The unigram feature extraction performed better than other n-gram models, hence Unigram features were considered for further evaluation.
   The output of the feature vector dictionary was in the form: [(('lesley', 'NN'), 2), (('im', 'NNS'), 3)]
   The output of this model with pre-processing and feature counts gave an improved mean rank score of 2.375 with an accuracy of 56.25%. From the Similarity matrix between document vectors, the most similar pair were Jack and Phil with similarity score of 0.87 and the least similar were Ian and Max with a score of 0.77. Other least similar pairs were that of (Max, Christian) and (Minty, Shirley). In case of the most similar pair of Jack and Phil, the similarity is high as there are many common word features between their dialogues, eg: (want,what,drink,mean) and many other words

occur in both character dialogues, hence giving rise to high similarity score, whereas in the case of the least similar pairs, there are lesser number of common words between the characters, thus giving the least similarity scores.

2) For the second experiment (question 4-6), I have used the TFIDF method instead of just using counts of the feature which will give high TFIDF score for features that are rare and also not too common. For this experiment the TfidfTransformer() library is used which takes as input the count vector matrix of the previous experiment. The final validation mean rank is further improved giving a rank of 1.43 and accuracy of 68.75%. The most similar character pairs are found to be again of Jack and Phil scoring 0.65, with the least similar pair of Heather and Jane with a score of 0.47.

RESULTS:

Comparison of Mean Rank and Accuracy metrics scored for the Validation Set

Technique	Mean Rank	Accuracy
a. Word Count Vector *without* pre-processing and POS Tagging	4.5	25%
b. Word Count Unigram Vector *with* pre-processing and POS Tagging	2.375	56.25%
c. TF-IDF Unigram Vector *with* pre-processing and POS Tagging.	1.437	68.75%

The table above shows the mean rank and accuracy for different techniques applied to improve performance. The best performance is observed in the case of TFIDF feature extraction method, which shows a good improvement in the metrics considered, which is due to the fact that TFIDF gives more weightage to terms that are rare and less weight to very common words appearing across the documents.

Mean Rank computes the mean of the cosine similarities across all the character pairs in the corpus. The best model is the one with mean rank close to the value 1 with the worst being close to the value 16.

The best method i:e the TF-IDF method with the same pre-processing and POS tagging technique, is applied to the unseen test data, which now considers the entire 400 training samples and 40 unseen test samples. Below table records the performances on the unseen test data.

Comparison of Mean Rank and Accuracy on Unseen Test Set

Technique	Mean Rank	Accuracy
a) Word Count Vector *without* pre-processing and POS Tagging	5.12	31.25%
b) TF-IDF Unigram Vector *with* pre-processing and POS Tagging.	1.437	87.5%

CONCLUSION

After comparing all the experiments, which included a combination of pre-processing and feature extraction techniques, the mean rank has highest score of 1.437 when the TFIDF approach was used on the Test data as compared to just considering vector counts giving an accuracy of 87.5%. The increase in accuracy compared to validation set could be due to the extra dialogue texts considered in training phase which improved accuracy score.

The experiment can be further analysed using other feature extraction techniques like word2vec or SelectKBest to improve mean rank of the model.