ECS763P NLP Assignment 2: Vector Space Semantics for Similarity between Eastenders Characters

Animesh Devendra Chourey 210765551

Queen Mary University of London —- January 17, 2022

Q1. Improve pre-processing

Preprocessing is a method of preparing text data for the model by cleaning it of noise such as emotions, punctuations, stop words, and words with similar lemmas. To do this, we use several methods we took a number of precautions, such as Lemmatization, stemming, removing stop words, removing punctuations, tokenizing, lowercasing the characters, checking if the length of the word is always greater than 1, making a list of ”serOfWords”, etc. Due to doing all these things, the mean rank went from 4.5 to 2.625 with an accuracy of 0.5 and a mean cosine similarity of 0.93.

Q2. Improve linguistic feature extraction

To extract features for the NLP task, feature extraction is used. The extracted features are PosTag, n-grams, and previous words. The PosTags are added using the nltk library; the bigrams and previous words are added by enumerating each token It is also initialized with a counter which keeps counts of each feature, such as word@nextword:3, word@previousword:6, etc. Lastly, a dictionary is returned with keys as features and values as counts. The previous mean rank was 2.5 and after improving the feature extraction function the mean rank is now 2.5 with 0.91 mean cosine similarity and 0.5625 accuracy.

Q3. Add dialogue context data and features

This dataset consists of only current lines spoken by the character in a scene of an episode. By adding the lines spoken by other characters in the same scene of the same episode, more information is obtained, and this thus aids in recognizing character names better. We accomplished this goal by adding the lines spoken by other characters in the same scene of the same episode.

We started by zipping the Line, Character name, Scene, and then enumerating through each row of data. Then, using the index, we extract the previous line spoken by some character, and a string of PRE is attached to the end of it. Similarly, the next line spoken by some character is also extracted, and a string of NEXT is attached to it. To keep the dataset unbiased, only the first 360 lines spoken by each character are selected for the training set, and 40 lines for the validation set. By doing so, the data imbalance is resolved. The next step is to concatenate all the lines and add a string ending in EOL to identify the end of the episode. The following results are obtained: Mean rank - 2.1875, mean cosine similarity - 0.91, accuracy = 0.37

Q4. Improve the vectorization method

In this task, we should use frequency-inverse document frequency instead of dc vectorizer. Alternatively, one-hot encoding can also be used. The dc vectorizer encodes features one-at-a-time using a straightforward approach. Certain tasks cannot be encoded one-at-a-time. The feature dimension, for instance, increases dramatically. Token similarity is not captured. In contrast, the TfIDF solves the issue of increasing dimensions and also detects some similarities between the tokens. The TfIDF is calculated by taking the total number of documents, divided by the number of documents that contain the token in question. The results are : mean rank 1.625 , mean cosine similarity 0.83, accuracy: 0.56.

Q5. Select and test the best vector representation method

We used all the methods from the previous tasks in this task as well. Each character's first 400 lines of dialogue were used for training. There are 199455 words in the entire train dataset. The test set uses the first 40 lines spoken by each character. It has 19989 words in total. This task uses tokens, tokens@postag, tokens@previous word, tokens@previoustopreviousword, and tokens@nextword. Tf-IDF is the vectorizer. The results are as follows: mean rank 1.25, mean cosine similarity 0.84, accuracy: 0.81.