15th December 2018
Group Project Report
Natural Language Processing
Kalyssa Owusu, Nana Ama Atombo Sackey, Nutifafa Amedior and Molife Chaplain

Topic Modeling

The three ideas chosen included:

- Preprocessing

  The corpus that was going to be used had to go through some form of normalization in order to make training and modelling of our topics very efficient and effective. Our preprocessing stage included removing numbers, unknown characters and question marks. In addition, stopwords were removed, the text was case folded, and unicode normalization was performed on the text corpus.

- Vectorizing

  The corpus was initialised into a Pandas dataframe containing the topics and answers using the pandas library after the preprocessing stage. The Tfidf Vectorizer from the the Scikit Learn library was used in vectorizing, in an effort to establish the importance of certain words in the training corpus. Considering the wide variety of topics in our training corpus, we believe this was necessary in order to make a distinction between various questions and their relationships with their respective topics in building our classifier.

- Classification

  After training with various classifiers including Naive Bayes, and Logistic Regression, and evaluating the F1 measure, we decided to use Logistic Regression, as it produced the best metrics out of all the models tested. Comparing the output of .

15th December 2018
Group Project Report
Natural Language Processing

Kalyssa Owusu, Nana Ama Atombo Sackey, Nutifafa Amedior and Molife Chaplain

Question Answering

The three ideas chosen were:

- Preprocessing

  We needed to process the available raw data in order to make it useable for our ultimate task of classification and answer determining. The provided corpus had tokens (;',.) and in order to make our data 'cleaner' we had to remove these tokens by normalization to ensure that training was smooth and the model was trained on clean data. We removed the tokens and saved the data as a pickle file in order to save time on continuous training during subsequent executions of the model.

- Vectorization

  Since Cosine Similarity was to be employed, the vector representation of our questions was needed in order to calculate the norms and the angles between the two vectorized questions to find their similarities - using words would not suffice.

- Cosine similarity

  The cosine similarity uses the euclidean dot product, and to be able to compute this, we needed the term frequencies of the words within our corpus. Cosine similarity can then use these vectors to calculate the similarity or dissimilarity between two words or documents, and given the measure is ranged from -1 to 1, but because of Tf-Id, the measure ranges between 0 and 1, meaning unsimilar and exactly the same respectively. This enables the model to assess the similarity between the asked question and the questions within the corpus, and return the response to the most similar or exact match of the question.