MCA ASSIGNMENT 3

Saksham Vohra
2016085

1. For data preparation, I first created tokens from documents present in the abc corpus, and then removed all punctuations from the tokens.
2. Created 2 dictionaries to map from an index to word and a word to index
3. Created a list containing (window size = 2)
    a. positive pairs (index_of(focus word),index_of(context word), 1)
    b. Negative pairs (index_of(focus word), index_of(non-context) word,0)
4. Created batches of size 2832
5. For the model:
    a. Create embeddings for the focus word - e1
    b. Create embedding of context word - e2
    c. Prediction = log sigmoid(e1*e2)
    d. MSE loss used for calculating loss between predicted and Ground Truth Label
    e. Adam Optimizer used
    f. 5 epochs
6. TSNE created for each epoch using weights of the embedding layer.
7. In summary, now given 2 words in the corpus, we can predict if the second word lies in the context of the first word or not in context for the corpus.
8. While analysing the scatter plots, it was observed that the words tend to form 2 clusters, which was already expected since the abc corpus contained 2 files- science and rural. Although the labels are not very clear, these clusters seem to belong to these respective documents.
9. As the number of epochs increase, the number of words in the center cluster tend to increase.

For Question 2

1. Retrieval score for Relevance Feedback over 3 iterations: 0.645
2. Retrieval score for RF + Query Expansion over 3 iterations: 0.610

For RF, for every query, we extract top n relevant and top n irrelevant documents, and modify the vec_query vector according to the formula given in the instructions, and then calculate cosine_similarity for new vectors.
For RF+QE, we perform the same operations in RF, additionally, I created a matrix A, which contained the count of word x in document y.
We then calculate word-similarity matrix C = A*A.T
For top n1 words in a query, we retrieve n2 similar words to those n1 words, and add tfidf transform of those words to the query.
For both the algorithms, the score improves significantly from the score achieved from simple cosine_similarity method, but the score for RF + QE was lower than that of RF, which was not expected since in RF+QE method, we explain the query in more detail, which should improve the retrieval performance.

MCA ASSIGNMENT 3

Saksham Vohra
2016085

A possible explanation for this behaviour could be the fact that QE leads to addition of some terms that are not relevant to the query, hence decreasing the overall relevance of the query.

Reference for QE: https://nlp.stanford.edu/IR-book/html/htmledition/query-expansion-1.html