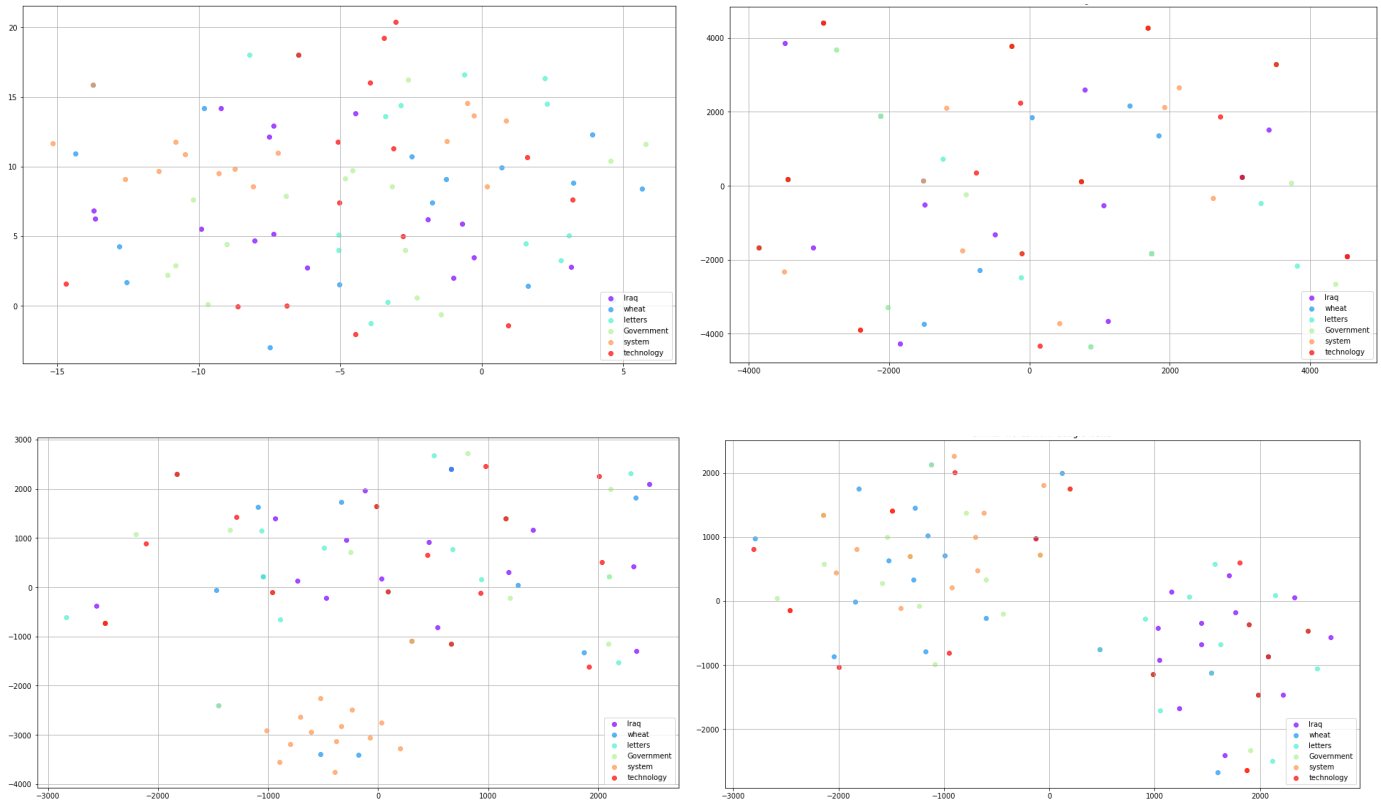
MCA Assignment-3

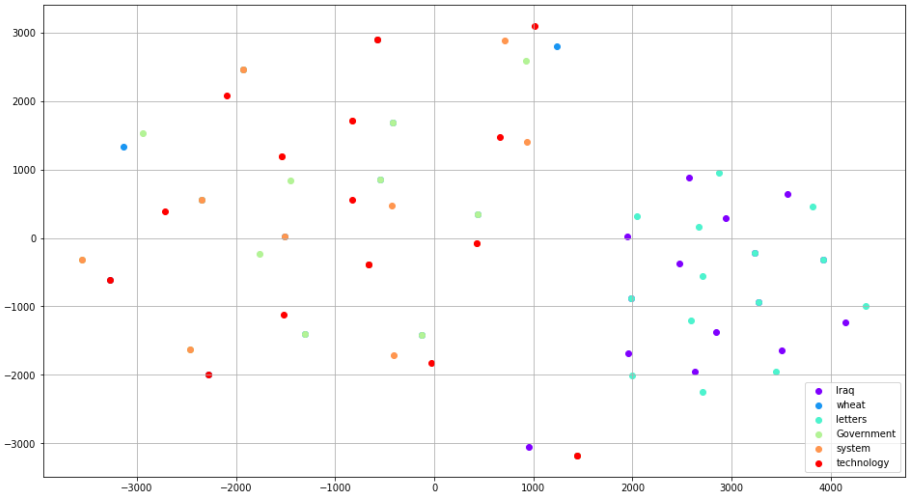
Priyanshi Jain
2017358

Problem 1:

Firstly from the 'abc' corpus of nltk, a vocabulary of 1000 words is created based on the most frequent words. Stop_words are removed from the corpus due to their high occurrence frequency. For each word, we take the neighbouring words to be context words. A window size of 5 is taken. We use negative sampling to train our model, i.e, for each target word, either the given word is a context word or a negative sample. We give two words as input to the model, one a target word, second a context word or a negative sample. It looks up the embedding (400-length vector) of these words, calculates the similarity between the words, passes it through a sigmoid layer and outputs 1 or 0 (1 for context word, 0 for negative sample). To calculate the similarity, cosine similarity is used. The error is back-propagated to update the embedding layers so that the similarity score between related words is high.

The model was trained for 1000000 epochs.
Following are the graphs for every 200000 epoch:





It can be seen from the graphs that the words that are similar to each other start forming clusters together. In the last plot, there's a clear distinction in two groups and some clusters between those groups. The reason for not being clear distinction between all the words could be that these target words are somewhat similar to each other, which could also be seen from the similar words printed for these.

Problem 2:

MAP values for Relevance feedback over 3 iterations:

1. 0.7595650320984177
2. 0.8133791197004292
3. 0.8142558354559608

MAP values for Relevance feedback with query extension over 3 iterations:

1. 0.796775862622374
2. 0.7974801499220459
3. 0.7964785740319891

For the normal relevance feedback, the MAP values seem to increase with iterations, the reason being that we give the retrieval our feedback step by step and it improves.

For relevance feedback with query extension, there doesn't seem to be an increase/decrease in the MAP values, the reason could be that we add the top n terms from the ground truth to the query, so once those terms are added, we don't expect it to change much with epochs.

Resource used for problem 1:

1. https://adventuresinmachinelearning.com/word2vec-keras-tutorial/
2. https://towardsdatascience.com/google-news-and-leo-tolstoy-visualizing-word2vec-word-embeddings-with-t-sne-11558d8bd4d