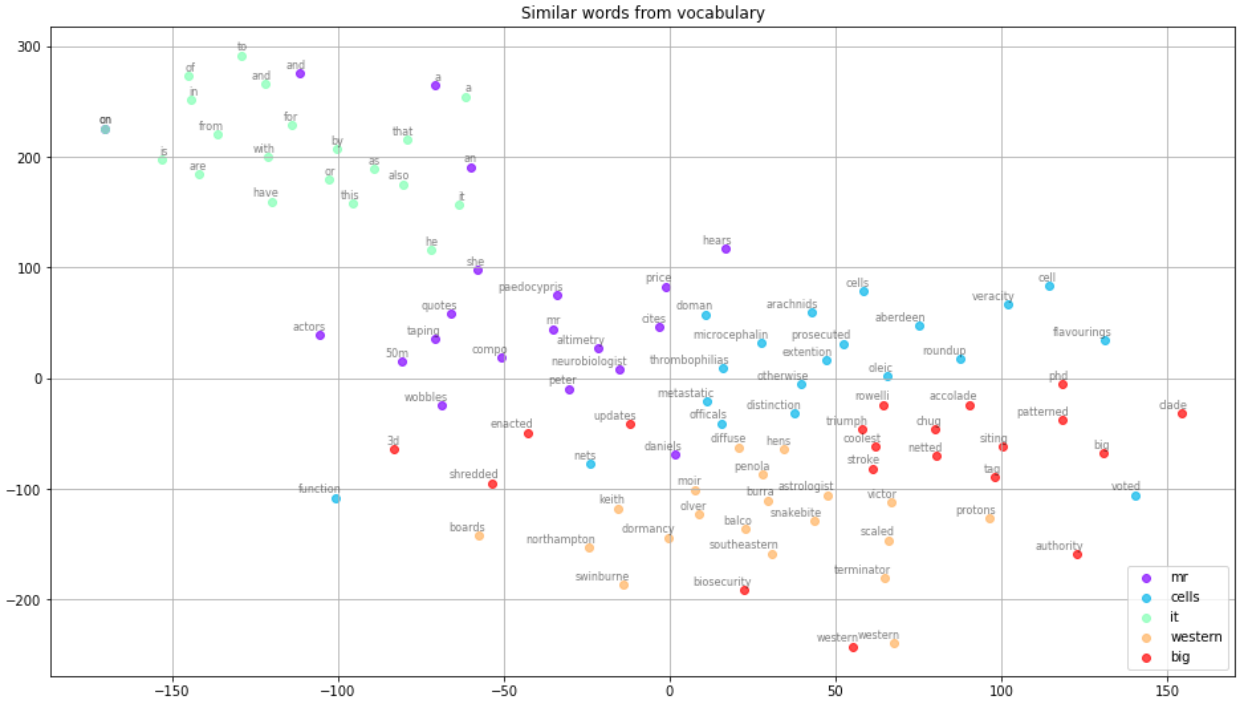
MCA Assignment 3

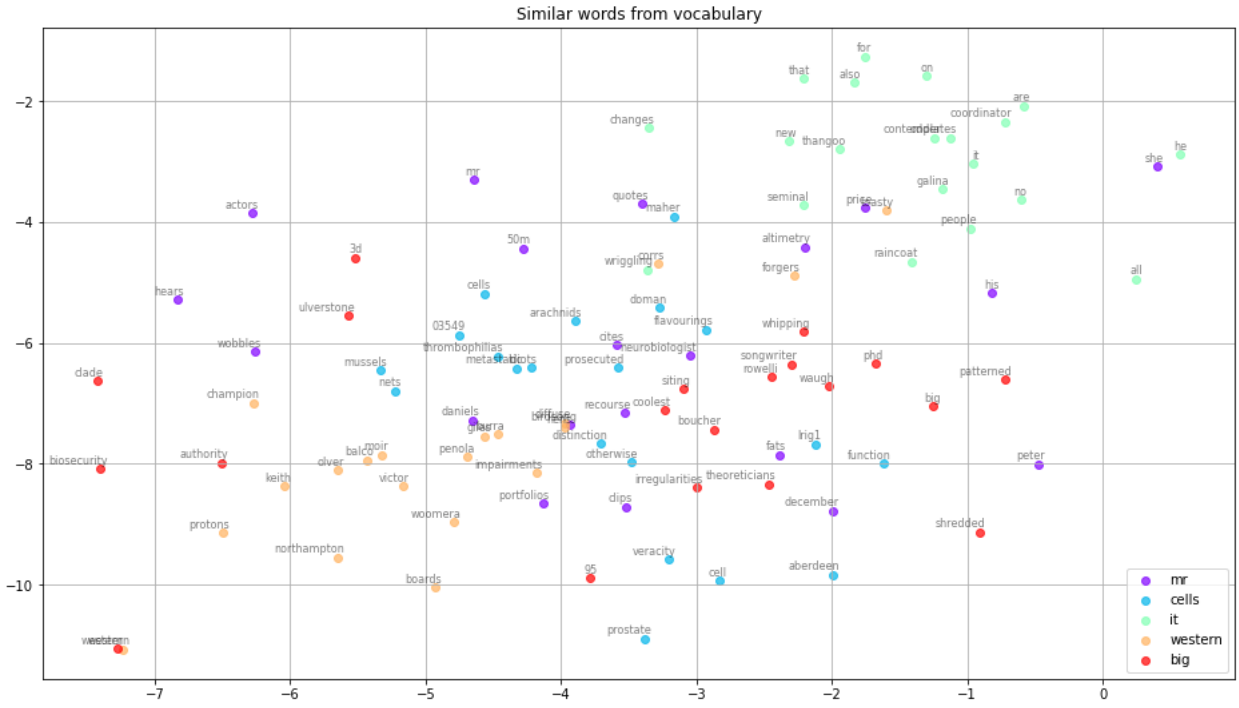
Shwetank Shrey (2016095)

Question 1 - Implement Word2Vec

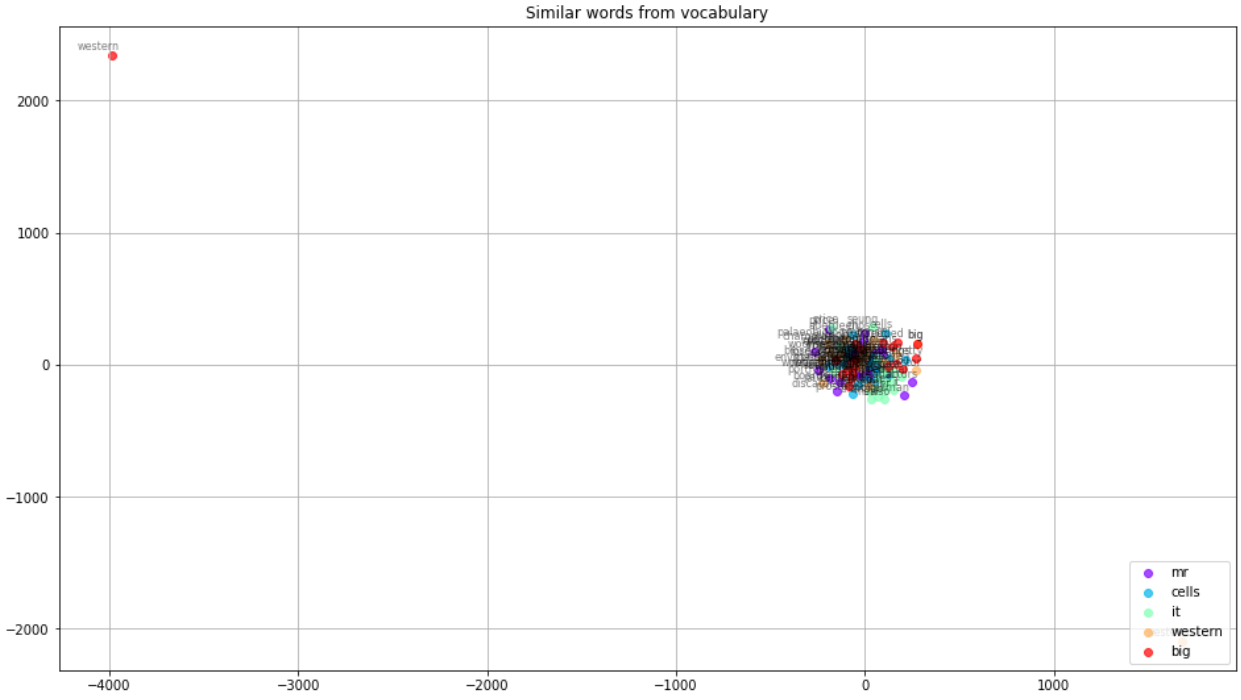
For the first part, the word2vec algorithm has been implemented from scratch. For this purpose, Keras was used to build the neural network architecture and TSNE from scikit learn to visualise the embeddings through the epochs.
Word2vec can be trained using either of two ways - continuous bag of words (CBOW) and skipgrams. In this implementation, we use skipgrams for the purpose of training the model. The technique includes further optimisations using negative sampling and subsampling. We assume context is defined by proximity and take words in a context window as context words to a target while others are not. Therefore, we create pairs of words called skipgrams, which are basically pairs of words with some distance between them. The skipgrams with distance less than a threshold are positive samples for us. Due to our optimising using negative sampling, we also take some negative samples along with positive samples. We further optimise using subsampling by having the probability of a sample skipgram taken from the dataset inversely proportional to the frequency of the words.
The architecture of the word2vec neural network model is described as follows - we have two layers for the input target and context words which are then passed to embedding layers to create two layers for target and context word vectors. We compute the dot product of these vectors which is projected as the output. While training, the input target and context words are passed as inputs while the information whether this is a positive or negative sample is the output.
We train our dataset and after every epoch, we calculate the twenty most similar words to random validation examples creating clusters and plotting using TSNE.
We find that clusters are being formed for each validation example which gets denser and sparser. *There are some visible cluster-like structures but these seem more intermingled as we further train our model. This may be due to Principal Component Analysis reducing the dimensions to two and the similarity being more distinguishable at a higher dimension than in two dimensions.*



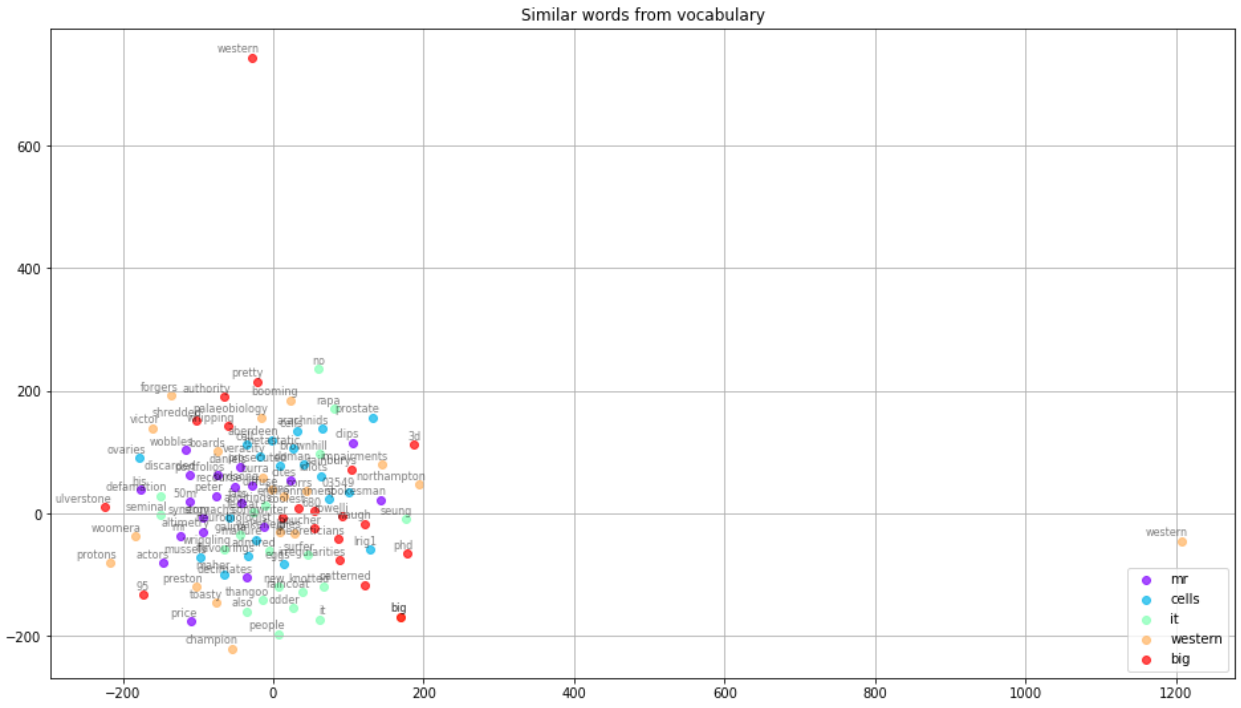
TSNE after the first epoch.



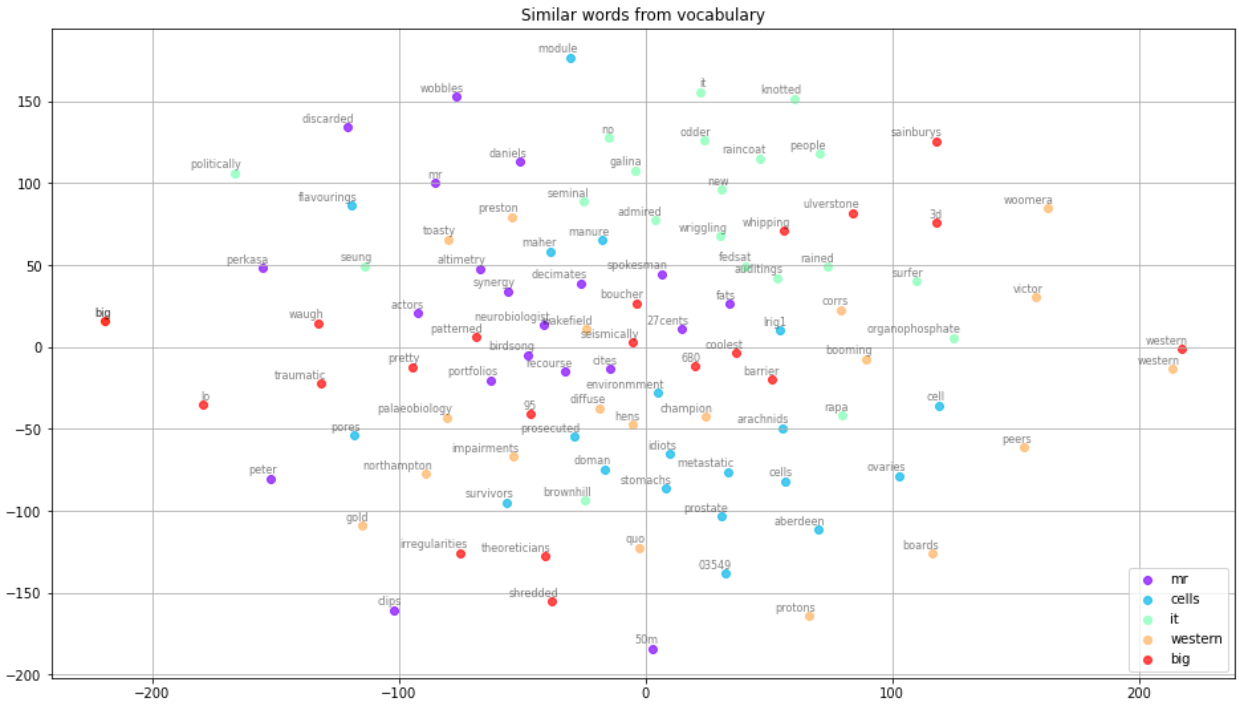
TSNE after the second epoch.



TSNE after the third epoch.



TSNE after the fourth epoch.



TSNE after the fifth epoch.

Question 2 - Document Retrieval using Query Expansion

#	Baseline	Relevance Feedback	Relevance Feedback and Query Expansion
1	0.5184	0.5918	0.5794
2		0.6107	0.6030
3		0.6207	0.6095
4		0.6233	0.6121

In this question, we take α = 0.75 and β = 0.15. For relevance feedback, we take the top 10 most relevant documents and for query expansion, we take the top 5 most similar terms to the most important term in the query using automatic thesaurus generation from the vocabulary. *Increasing the number of iterations increases the accuracy which is expected.*

References

- https://adventuresinmachinelearning.com/word2vec-keras-tutorial/
- Relevance Feedback and Query Expansion (Helen Yannakoudakis)