Natural Language Processing

# Assignment 2

## Q1) Word similarity

Improve the similarity function for matching query entered by user with a predefined list of bank names. An example is given in the attached code.

## Q2) Sentiment Analysis

Build sentiment classifiers using bag of words and ngram model for the given dataset. The data set contains reviews about Apple products with labels 1 (negative), 3(neutral), 5 (positive), and not_relevant. You can use scikit-learn (machine learning tool for python) for using implementations of classification algorithms. Perform multiclass classification. For multi class you have to classify the tweets into one of the four categories (negative, neutral, positive, not_relevant).

Split the data into train and test set by using "train_test_split(DataSet)" of scikit.
Implement following feature extraction methods.

- Bag of words based on raw counts
- Bag of words based on TfIDF
- ngrams (unigrams, bigrams, trigrams)

Read following links about using Vectorizer (Bag of words based on raw counts) and transformer (Bag of words based on TfIDF) for converting list of sentences to vectors
https://scikit-learn.org/stable/modules/feature_extraction.html
https://scikit-learn.org/stable/auto_examples/text/plot_document_classification_20newsgroups.html#sphx-glr-auto-examples-text-plot-document-classification-20newsgroups-py

You can use scikit learn for implementation of different classifiers as explained in above links. Use following classifiers: Naïve Bayes, Logistic Regression, Random Forest, SVM, Perceptron.
Calculate accuracy, Precision, Recall and F-score for all classifiers and report the results in tables. Make a table for multiclass classification results for different classification algorithms. Report both micro average and macro average of all measures.

**Submission**
Submit the code files and result tables as zip file on Google classroom.