Multimedia Information Systems *CSE408, Fall 2022*

# P01: Text Classification and Sentimental Analysis

## 1 Basic Requirements

1. Programming language: Matlab (recommended and template codes provided), or other languages (without template codes)
2. Implement your code under **Code** folder, and read data from **Data** folder (one **neg** folder contains all the negative reviews and one **pos** folders contains all the positive ones).
3. Final project report under name **report.pdf** at the project root folder.

## 2 Part I: Text Classification with Bag of Words and kNN (50pt)

### 2.1 Vocabulary (lexicon) creation (10pt)

1. Template file **buildVoc.m**
2. Function template **function voc = buildVoc(folder, voc)**;
3. Input: a folder path, which contains training data (a bunch of .txt files from Multimedia App reviews in HW00);
4. Output: matlab cell array **voc**, which represents the vocabulary of the words shown in the training data, except the stop words (stop words list is embedded in the code template)
5. Implement your code under **%PUT YOUR IMPLEMENTATION HERE** tag;
6. Useful matlab functions **strtok(), lower(), regexprep(), ismember(), any()**;

### 2.2 Bag of Words feature extraction (20pt)

1. Template file **cse408_bow.m**
2. Function template **function feat_vec = cse408_bow(filepath, voc)**
3. Input: a file path **filepath**, which contains one review (one .txt file) and a vocabulary cell array **voc** from previous sub-section.
4. Output: one dimentional matlab array **feat_vec**, which represent the bag of words feature vector given the vocabulary **voc**;
5. Implement your code under **%PUT YOUR IMPLEMENTATION HERE** tag;
6. Useful matlab functions **strtok(), lower(), regexprep(), ismember(), any()**;

### 2.3 k-Nearest Neighbor Classification (20pt)

1. Template file **cse408_knn.m**
2. Function template **function pred_label = cse408_knn(test_feat, train_label, train_feat, k, DstType)**
3. Input: 1) test feature vector **test_feat**; 2) training set groundtruth label set **train_label**; 3) training set feature vector set **train_feat**; 4) Hyperparameter **k** of knn; 5) Distance computation method **DstType**, 1 for sum of squared distances (SSD) and 2 for angle between vectors and 3 for Number of words in common;
4. Output: predicted label **pred_label** of the testing file. 1 for positive review, 0 for negative review;
5. Implement your code under **%PUT YOUR IMPLEMENTATION HERE** tag;
6. Useful matlab functions **sort()**;

### 2.4 Test your implementation

1. After your implementation, you could run **P01Part1_test.m** to debug and validate your code. It basically iteratively select one of the training review file as a validation file.
2. Question? Where is the testing data. Come up with one more product review and see if your system be able to classify it correctly?

## 3 Part II: Text Sentimental Analysis (30pt)

1. Implement a basic sentimental analysis module. Read in a lexicon, in which each word has a sentimental score. Iterate through each review file and sum up the sentimental scores for each word that exists in the sentimental strenghth lexicon;
2. Template file **sentimentalAnalysis.m**
3. Input: a file path **filepath**, which contains one review (one .txt file) and a word with sentimental strenghth file **wordwithStrength.txt** under **Data** folder.
4. Output: one sentimental score.
5. Implement your code under **%PUT YOUR IMPLEMENTATION HERE** tag;
6. Useful matlab functions **strtok(), lower(), regexprep(), containers.Map()**;

### 3.1 Test your implementation

1. After your implementation, you could run **P01Part2_test.m** to debug and validate your code.
2. Question? What is the accuracy of the performance of your code?

## 4 Report Requirements (20pt)

Please include the following analysis in your report.

1. Make sure to expalin where the algorithms worked and where they didn't and why. You are encouraged to use both text and plots to explain your observations.
2. Analyze Hyperparameter $K$ in the KNN part, which $K$ you emprically observed that could achieve the best performance?
3. Among the three distance metrics (sum of squared distances (SSD), and the angle between vectors and Number of words in common), which one intuitively makes more sense for classifying positive and negative review? Which one you empirically observed it to achieve the best performance?
4. For Text Sentimental Analysis task (Part II), which review in our dataset has the highest positive score but it is actually a negative review? And, which one has the lowest negative score, but it is indeed a positive review? Which set of words from these reviews confused your sentimental analysis system?

## 5 Submission Instruction

Please place your answers under one .zip file with a formatted file name: P01.zip and submit it on Canvas (as a group project). The .zip file shall include one folder with Matlab source code under name "code" and one report in .pdf format. Each group only needs one submission.