NLP Project 1 Text Classification Writeup
Jing Jiang

Instruction to use my code

My implementation used Python 3.7.3. The necessary packages to run the code are numpy, pandas, nltk, tqdm, scikit-learn. You can pip install them if you don’t have them on your computer. Once ready, you can do python knnsubmission.py to run the code.

Machine learning method

I used weighted-KNN algorithm to implement this text classification task. I used word_tokenize function in nltk with the Porter Stemmer to tokenize the text. When tokenizing the text, I lowercased all tokens and removed all tokens with numbers and punctuations. I also used the English stop-word list in scikit-learn. I found that removing the numbers had the most significant improvement to the performance of the classifier. I use TfidfVectorizer in scikit-learn to convert and normalize my vectors. I used cosine_similarity to calculate the tf-idf distance of two texts.

Weighting Scheme

Adding weights based on the distance to the nearest neighbors will improve the overall performance of a KNN system. The reason for this is: the farther a neighbor is, the more it "deviates" from the "real" result. Or in other words, we can trust the closest neighbors more than the farther ones. Trying different weighting schemes, I found the harmonic series weighting scheme worked best for the 1st corpus data. That is to say, the weights for the neighbors (from the nearest to the farthest) will be 1, 1/2, 1/3, etc.

$$\sum_{i}^{k} 1/(i+1) = 1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{k}$$

Fig. 1. Harmonic Series

For Second and Third datasets

Before prediction for the test set, my system runs a 5-fold cross validation in order to tune the hyper parameter. There is only one hyper parameter for my system: k. The best k value is chosen via grid-search from a list of potential k values. k= [1, 3, 5, 7, 9, 11]. When tuning finished, my program will print the best k value in terminal.