CSE 6240 Spring 2020 - Homework #2

Analyzing a Movie Review Dataset - Part 2

Due: Feb 24th, 2020

The aim of this homework is to give hands-on experience with word2vec, Google's pre-trained vocabulary sets.

Guidelines:

- You are given a solution template (an .ipynb file) and you are required to add your code at the appropriate places and submit it.
- The solution template contains questions with multiple sub-parts for each one.
- For the coding exercises, please add your code below the comment "Add your code here."
- For the theory questions, please add your answers in a text cell below the questions. The cell type can be changed under Cell -> Cell Type.
- Points distribution for each question is added for your reference.
- Please **do not** modify any of the function definition or documentation.

The Problem Statement:

Read through this tutorial on kaggle, https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-1-for-beginners-bag-of-words , to familiarize yourself with its python tools and workflow. Write your own annotated ipython notebook(s) to reproduce the steps in the blog and complete the exercises below. You can start with the solution template and the sample code provided in the tutorial, but you should clean it up, document and refactor as necessary.

0: Text Pre-Processing[10 points]. Using the blogs as a reference and as per the template code provided to you, perform text pre-processing for the Word2Vec model.

1: Word2Vec [80 points]. Using the blogs as a reference:

a. [20 points] Create vector representations for each movie post in your training set by training word2vec with context=5, embedding dimension = 100, min_words=40. We'll call the collection of these representations Z1.
b. [20 points] Create vector representations for each movie post in your training set by loading the pretrained Google word2vec model. We'll call the collection of these representations Z2.

c. [20 points] With k=10, do k-means clustering on each set Z1, Z2. Print a table of the words in each cluster for Z1 and for Z2.
d. [20 points] Featurize the training and test reviews in Z1, Z2 to produce design matrices X1, X2 as described in part 3 of the blog series. Basically, each review is converted into a bag of centroids feature vectors, for a review we return an array of length equal to the number of clusters with each element of the array indicating how many words(tokens) of the review belong to that cluster.
e. Save X1, X2 (both train and test values) as .npy files. You will be penalized a total of 10 marks if the 4 .npy files are missing from your submission.

2: Classification Experiment[10 points]. Using the Kaggle blog series as a guide:

a. [5 points] Properly train and tune a collection of random forest classifiers using cross-validation for each of the design matrices X1, X2. You should end up with two classifiers; M1, M2.
b. [3 points] Calculate the f1 scores for the best value of n-estimator that you obtained using cross-validation in the previous part for each classifier computed on the test set.
c. [2 points] Which featurization technique works best for sentiment classification? Is this better or worse than the simple bag-of-words approach? What are at least three things you could do to improve the efficacy of the classifier?