CSE 261 & DSC 253

Adv. Data-Driven Text Mining

Homework 1: Basic Text Classification Introduction

Deadline: Jan. 27th (Tuesday of 4th Week)

In this assignment, you will train a text classifier using different feature representation techniques. You should submit the final writeup and code for this assignment. The writeup should be a pdf that includes experimental findings of the programming part. Submissions should be made on **Gradescope (course code: N2G2NV)**. Please complete homework **individually.** Please include the code of your solutions in the submission with a write-up describing how to run the code.

Resource: Please two datasets for this homework that are included in the HW1 download files on Piazza: **New York Times (NYT) news** and **AG News**. **NYT dataset** contains a text column consisting of news articles and a label column indicating the category to which this article belongs. **AG News** has just the text column. For questions 1 and 2, use the logistic regression classifier.

Classifier: It should be trained and tested on the NYT dataset. Shuffle the NYT data with random seed 42, and split it into training, validation, and test splits, with an 80%/10%/10% ratio.

(Grading will be only based on correct implementation, trivial performance difference caused by running random won't affect your grade.)

Task 1: Bag Of Words (30')

Train a text classifier using 3 types of bag of words techniques and report accuracy & macro-f1 score on the NYT test set.

・Each document is represented as a **binary-valued** vector of dimension equal to the size of the vocabulary. The value at an index is 1 if the word corresponding to that index is present in the document, else 0. (Consider using "`from nltk import word_tokenize`")

・Each document is represented by a vector of dimension equal to the size of the vocabulary where the value corresponding to each word is its **frequency** in the document.

・Each document is represented by a vector of dimension equal to the size of the vocabulary where the value corresponding to each word is its **tf-idf value**.

Task 2: Word2Vec (30')

Train a text classifier using 3 types of word2vec representation techniques using **100-dimensional** word vectors and report accuracy & macro-f1 score on the NYT test set.

・Using publicly available pre-trained **Glove embeddings** (http://nlp.stanford.edu/data/glove.6B.zip) as word vectors, a document vector is represented as an average of word vectors of its constituent words.

・Train Word2Vec (e.g., by `gensim` package) on AGNews text data and use them as word vectors to compute document vectors by averaging word vectors of its constituent words.

・Train Word2Vec on NYT text data and use them as word vectors to compute document vectors by averaging word vectors of its constituent words.

Task 3: Pre-trained Neural Models (40')

Train a text classifier using 2 types of pre-trained neural models and report accuracy & macro-f1 score on the NYT test set.

・Fine-tune the **BERT** (`https://huggingface.co/google-bert/bert-base-uncased`) for text classification. While tokenizing, set the maximum length to 64 and fine-tune for 3 epochs.

・Fine-tune the **ModernBERT** (`https://huggingface.co/answerdotai/ModernBERT-base`) for text classification. While tokenizing, set the maximum length to 64 *(ignore if you can train with full-length input)* and fine-tune for 3 epochs.