CS235 Homework 3 (Text Classification)

Fall 2025

1 Description

In this assignment, you will train a text classier using different feature representation techniques. You should submit the writeup and code for this assignment to gradescope. The writeup should be embedded into the notebook and includes experimental findings. Please complete homework individually.

You will need two datasets for this homework that are included in the HW3 download: New York Times (NYT) news. NYT dataset contains a text column consisting of news articles and a label column indicating the category to which this article belongs. Use the logistic regression classifier for the questions below. The classifier 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 a 80/10/10% ratio(e.g., use random_state in *sklearn.model_selection.train_test_split*).

You are encouraged to use built-in libraries. No need to code these methods from scratch.

2 Bag Of Words (20 points):

Train a text classifier using the following document representation techniques and report accuracy, macro-f1 score, and micro-f1 score on the test set. **macro-f1** score refers to evaluating multi-class classification models by calculating the F1 score for each class independently and then taking the unweighted average of those scores.

(a) 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.

(b) A 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.

(c) 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.

3 Word2Vec (20 points):

Train a text classifier using the following document representation techniques using 100-dimensional word vectors and report accuracy, macro-f1 score, and micro-f1 score on the test set. Compare and analyze their performance.

(i) Using publicly available pre-trained Glove embeddings as word vectors, a document vector is represented as an average of word vectors of its constituent words.

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

4 BERT (20 points):

Fine-tune the BERT (bert-base-uncased) for text classification and report accuracy, macro f1-score, and micro f1-score. If you are using PyTorch, hugging face transformers is highly recommended for this task. While tokenizing, set the maximum length to 64 and fine-tune for 3 epochs.

Note, refer to this webpage for example (`https://huggingface.co/google-bert/bert-base-uncased`). You will have to install required libraries by typing the following in your notebook.

```
# Install & import required libraries

!pip install -q torch transformers scikit-learn tqdm

import torch

from transformers import BertTokenizer, BertModel
```

5 Summary of Results (10 points)

Discuss your findings and results.