

COSC 426 F24 HW 2

In this homework you will be using the code you wrote in Lab 6 and the `NLPScholar` toolkit to train and evaluate different Bayesian classifiers on the [IMDB sentiment analysis dataset](#) from Stanford. This assignment is Due **Monday Oct. 28**.

Provided files

- `HW2.py`
- [Google doc template](#)

What to submit

- `HW2.py`
- pdf of your google doc

Part 0: Creating the dataset

As a first step, you will need to format the data from this dataset in a way that is usable with your classification code. Download the zip folder from the link above. Check out the data and get a sense for the structure of the folder. Create train, validation, and test sets in the correct format. Note: this is intentionally left a bit vague. Handling data preprocessing like this is an important skill in NLP and will help you as you work on your final project. Take this as an opportunity to productively struggle through this.

Part 1: Building a Bigram LM based Bayesian Classifier

Use the code you wrote in Lab 6 to build a Bayesian Classifier that uses a bigram Language Model with `add-0.01` smoothing. Report the accuracy of your model on the `test` split of your data in your google doc.

Part 2: Building a Neural Network LM based Bayesian Classifier

Use `NLPScholar` and the code you wrote in Lab 6 to build a Bayesian classifier that uses the `distilbert/distilbert-base-cased` Language Model (**not the text classification model**). Report the accuracy of your model on the `test` split of your data. You should finetune the model for **one epoch** on the relevant data. Note, while training is relatively quick, evaluating your a masked language model on data can take time. Make use of turing jobs, start small, and leave yourself time.

In your google doc, also describe how building a Bayesian Classifier using `distilbert/distilbert-base-cased` is different from using this model to run a `TextClassification` experiment using `NLPScholar`.

Part 3: Interim results discussion

Answer the following questions in the google doc template:

1. The result of evaluate for both the Bigram and Neural LM was a tsv file with one row per token, along with probability estimates associated with the word. Conceptually, what do the probabilities represent for each of the models?
2. Pick a few reviews and describe what differences you observe between the token-by-token probabilities between the models.
3. Which model was ultimately better at sentiment classification? Why does this make sense?

Part 4: Improving your Bigram models

The Bigram LM based Bayesian Classifier you trained was only one of many types of models you could have trained.

1. There are at least three properties (or *hyperparameters*) of your bigram model you can modify. List what these are. Which of these properties do you think is most likely going to have an impact on classification accuracy? Why?

2. For the hyperparameter that you think is most likely going to have an impact on classification accuracy, concretely explore how different settings of this hyperparameter influences classification accuracy.
3. Which of the hyperparameter settings that you tried resulted in the best classification accuracy? Why do you think this was the case?