COSC 426 F25 Lab 5

Introduction

In this lab you will build and evaluate a naive bayesian classifier for sentiment analysis. You will also get practice working with File I/O, objects and classes in Python.

This lab las three required parts, and one optional part.

- Part 1: Using python objects and classes to implement a unigram language model
- Part 2: Implementing the building blocks of a naive Bayesian classifier
- Part 3: Training and evaluating a Bayesian classifier on sentiment analysis
- Part 4 (optional): Writing a BigramModel class, and using this in a Bayesian classifier

Grading

In order to get the `Meets Expectation` designation for this lab, you should correctly complete the first three parts. To get the `Exceeded Expectation` designation you should also correctly complete part 4.

Provided files

- `UnigramModel.py`: the file where you will implement your unigram model
- `Lab5.ipynb`: the part where you will implement the building blocks, and run the Bayesian Classifier
- `util.py`: Python file with additional useful functions

What to submit

- `UnigramModel.py` with the functions implemented
- `Lab5.ipynb` with the code implemented and answers to questions in each part
- `BigramModel.py` (if you choose to attempt part 4)

Part 1: Building a unigram model

In this part you should:

1. In `UnigramModel.py` implement the incomplete functions: `get_prob` and `evaluate`
2. In `Lab5.ipynb` familiarize yourself with how to use the `UnigramModel` class.
3. In `Lab5.ipynb`, answer the following questions:

- What does training the unigram model entail?
- What does evaluating the unigram model entail?
- How is the `unk_token` parameter used?

Part 2: Implementing the building blocks of a Bayesian Classifier

In this part you should implement and test the following functions:

- `get_likelihood`
- `get_prior`
- `get_posterior`
- `classify`
- `calc_accuracy`

Part 3: Building a Naive Bayesian Sentiment classifier

Use your code from the previous two parts to train and evaluate a Bayesian classifier on the IMDB sentiment analysis dataset.

Concretely, do the following:

1. Download and unzip the data
2. Train your positive vs. negative sentiment analysis models on the files in the `train` subfolder
3. Generate predictions from your models on the files in the `test` subfolder, and evaluate these predictions.
4. Display and discuss your results.

Part 4 (Optional): Building a Bigram Bayesian Sentiment classifer

For this part, repeat the experiment in Part 3, but instead of using a unigram model, use a bigram model. Concretely you should:

1. Create a `BigramModel` class in `BigramModel.py`
2. Repeat steps 2-4 from Part 3, but using the bigram model