COSC 426 F24 Lab 1

Introduction

The purpose of this lab is to introduce you to the NLPScholar toolkit we will be using in the class and to serve as a Python refresher. By completing this lab, you will demonstrate that you can:

- Describe the structure of the toolkit
- Write config files to run different types of experiments
- Work with File/IO, lists, dictionaries and strings in Python.
- Use the toolkit in the interact mode to develop intuitions about word probabilities.

Pre-requisites This lab assumes that you have already cloned the NLPScholar repository and have installed the `nlp` environment by following the instructions in `Install.md`.

Structure

This lab has three parts:

1. Read through the documentation of the toolkit and answer questions.
2. Write three helper functions in Python.
3. Develop intutions about the predictability estimates that the toolkit returns. To do this, you will select some sentences to explore with the helper functions from part 2, and answer some questions.

Provided files

- Lab1.py
- through-the-looking-glass.txt
- A google doc template to write responses

What to submit

- Lab1.py
- A **pdf** of the google doc template with the answers.

Part 0

Before starting each lab, get the latest version of the NLPScholar rep by first navigating to the folder on terminal and then executing

```
git pull
```

Part 1 (Suggested time: 20 minutes)

Read through the README of the Toolkit. Use the google doc template to answer the following questions:

1. Which experiment and mode would you use if you want to:

   - Train a model classify whether a given sentence is talking about the election, which experiment and mode would you use?

   - Identify the sentiment of each of the words in a dataset of movie reviews given a model that is already trained on this task.

   - Find the word by word probability of an interesting sentence you found on the internet.

   - Find the average accuracy of an existing part-of-speech tagger.

2. Write a config file that can train a `roberta-large` model on the `wikipedia` dataset, which is within a larger `wikimedia` dataset. Set the modelfpath to `wiki_model`.

3. Where will the model that you trained in step 2 be saved?

4. You've trained a model to detect sarcasm and called it `sarcasm_model`. Evaluate this model on a new set of sentences called `test.tsv` which is stored in the following folder: `data/sarcasm/`. Set the predfpath as `test_results.tsv`
5. Where will the predictions you generated be saved?
6. Write a config file that will use the pretrained `huggingartists/taylor-swift` causal langauge model, and give you word by word predictability estimates for any sentence that you enter.

Part 2 (suggested time: 60 minutes)

Complete the three functions in Lab1.py. Make sure to read the function headers and docstrings carefully.

Part 3 (suggested time: 15 minutes)

Use the code you wrote in Part 2 and the google doc template to answer the following questions:

1. If you ignore all words that occur less than three times, what is the most frequent word in the `through-the-looking-glass.txt` file? What is the least frequent word?
2. For the most and least frequent words in your previous answer, what are the first three sentences in which these words occur? What is the probability of these words in these three sentences according to the masked language model distilbert-base-uncased? Do these probablities make sense? Why or why not?
3. If you repeated step 2 with the 500th most and least frequent words, do any of your observations change?