Project Description

Description

This project requires you to train two bigram models on text from Jane Austin's "Sense and Sensibilities", to test the models on three small datasets, and to write a short report about your results.

There are 3 tasks to complete, which are described in sections Task1, Task2, and Task3 respectively below. You need to obtain at least 60% to receive a passing grade.

Requirements for all tasks:

- It must be possible to run your scripts from the command line like this:

  ```
  python3 <model-script> <train-data> <test-data>
  ```

- Do not import any additional packages.
- Do not share your results or code with anyone, except your project partner if you have one. If you are using git, create a private repository for your code.

Submit your project by **Jan. 28, 2022** via Moodle. See the Submission Details section below for details.

Starter Code and Data

The parts of the starter code that you need to complete are described in *ToDo*'s in the starter code.

The starter code reads the training data, replaces OOV words, and generates the bigram count dataframe, vocabulary list, and unknown word list.

The training and test data contain one sentence per line. Tokenization is done by simply splitting on whitespace, and tokens are converted to lowercase when reading the data.

Three test corpora are included in the project. Two of the test corpora consist of sentences held out of the training data, *ja-sas-test1.txt* and *ja-sas-test2.txt*. The third test corpus, *ja-pap-test.txt*, contains the first sentences of Jane Austin's "Pride and Prejudice".

Task Descriptions:

Task 1: Add-1 Bigram Model 30%

Use the `BigramModel.py` starter code to create an smoothed bigram model using add-1 (Laplace) smoothing, as described in Jurafsky & Martin 3.5. See also the starter code for instructions.

Name your script `Add1BigramModel.py`

Task 2: Kneser-Ney Bigram Model 50%

Again using the `BigramModel.py` starter code, implement a smoothed bigram model using Kneser-Ney smoothing, as described in Jurafsky & Martin 3.6.

Use formula 3.35 in your implementation, and notice that formula is **not recursive**.

Detailed instructions are in the starter code, including some hints about how to improve the efficiency of your code. The Kneser-Ney algorithm requires quite a few calculations, and one needs to take care that the implementation is done efficiently.

Name your script `KNBigramModel.py`

Project Description

Task 3: Tests and Report 20%

Run the following tests:

1. Test your add-1 model on the test datasets *ja-sas-test1.txt*, *ja-sas-test2.txt* and *ja-pap-test.txt*
2. Test your KN model on the test datasets *ja-sas-test1.txt*, *ja-sas-test2.txt* and *ja-pap-test.txt*

Record the perplexity of the test data:

	ja-sas-test1.txt	ja-sas-test2.txt	ja-pap-test.txt
Add-1			
KN			

Also record the OOV rate of the training corpus, and make observations about the generated sentences for each of your models.

Write a short report of your findings, max 1 page in length. Include your results table, and discuss your results and observations. Compare smoothing methods and any differences between the test results of the two datasets. Explain other observations you made, for example: compare differences in the generated sentences; if the perplexity values match your expectations; OOV rate and how you would expect your results to change if you trained the model on text from a different author/writing style...

Do not just repeat what is in your table - describe what you learned.

Submission Details

The project is due on **Jan. 28, 2022**, and must be submitted via upload to Moodle.

Create a directory named *<lastname>*, or *<lastname1_lastname2>* if you are working in a pair, with the following contents:

- Add1BigramModel.py
- KNBigramModel.py
- *<lastname>*Report.pdf, or *<lastname1_lastname2>*Report.pdf

Submit the zipped directory to Moodle.

Honor Code

Use following Honor Code template, with your name(s) and other details, and include it at the top of each source file:

```
"""
Course:         Statistical Methods for NLP I: Parsing
Project:        Bigram Models
Author(s):      <Your first and last name(s) and matriculation number(s)>
Description:    <very short description of what the code does (e.g. Bigram model
                with add-1 smoothing)>
Honor Code:     I/We pledge that this code is my/our own work, and that
                no part of this work was copied from, or shared with, others.
"""
```