EECS 487 Introduction to NLP
HW1

Assigned date: September 7th, 2023
Due date: September 20th, 2023

Instructions:
Homework 1 is a collection of three assignments: a written (W) section, a coding (C) section, and an analysis (A) section. Each one must be submitted separately before the due date.

- **(W) Written:** You must complete the written section individually (collaboration is not allowed). Upon completion, turn in your submission (a single `.pdf` file) for the written section to the Gradescope Homework 1 Written assignment.
- **(C) Coding:** You may (but are not required to) work collaboratively with one other partner the coding (C) section. Upon completion, turn in one copy per group of the coding section to the Gradescope Homework 1 Coding assignment. You only need to turn in your filled-in versions of the files `language_model.py` and `naive_bayes.py` (do not turn in supplementary materials). All materials required for the coding section can be found on Canvas under Files > Homeworks > Homework 1.
- **(A) Analysis:** You may work with the same partner for the analysis section as you did for the coding section. Upon completion, turn in one copy per group of the analysis section (a single `.pdf` file) to the Gradescope Homework 1 Analysis assignment.
- Please refer to the syllabus for the late policy.

(W) Written Section

(W.1) Byte-Pair Encoding [10 points]

In the lectures on text normalization, we introduced Byte-Pair Encoding (BPE) for subword tokenization. In this exercise, you must implement the BPE algorithm for learning tokens by hand to process the following text:

```
thinkers thought this
```

Use $k = 3$ (the number of merge operations). Show your work by listing the merges in the order that you learned them. Please note: in the event that multiple token pairs are tied for most-frequent, you may arbitrarily choose which one to merge first.

Once you have done this, use your vocabulary to implement a token parser by hand. Tokenize the following text:

```
is he kith or kin
```

(W.2) Naive Bayes Classifier [10 points]

You are given the following table that contains 5 movie reviews. Your task is to train a naive bayes classifier on these reviews and use it to predict the label for an unseen review: "boring and very overrated movie". You need to use unigram features with add-α smoothing, $\alpha = 0.5$. Simply discard unseen unigrams (do not use add-α smoothing for them). Show the calculation of each probability **step by step**.

Label	Review
positive	great film very imaginative
negative	takes too long
negative	quite boring movie
positive	long but very very interesting
negative	waste of time

Table 1: Movie review training data.

(C) Coding Section

In this part, you need to solve two programming problems, one focused on n-gram language modeling and another focused on naive bayes classifiers. `hw1.ipynb` contains more detailed instructions for coding and serves as a "driver" for the code you will write. `language_model.py` and `naive_bayes.py` are where you will fill in your answers to the coding questions as directed by `hw1.ipynb`.

(C.1) N-Gram Language Model [40 points]

See hw1.ipynb for more details.

(C.2) Naive Bayes Classifier [26 points]

See hw1.ipynb for detailed instructions.

(A) Analysis Section

In this section, you will answer free-response analysis questions related to the coding problems.

(A.1) Analysis Questions for N-Gram Language Model [7 points]

After you have completed C.1, come back and answer these reflection questions.

- (A.1.1) Which model produced smaller perplexity values: the word-level or character-level model? Why? [3 points]
- (A.1.2) Which model generates better text? The word-level or character-level model? Why? [2 points]
- (A.1.3) How did you choose to split your data? Were you able to correctly predict the class of the sample review? [2 points]

(A.2) Analysis Questions for Naive Bayes Classifier [7 points]

After you have completed C.2, come back and answer these reflection questions.

- (A.2.1) Compare micro and macro averaging of F1 scores. What might be the advantage(s) of one over the other and why? Hint: what if the test data is unbalanced? [3 points]
- (A.2.2) Suggest some possible ways for improving this Naive Bayes approach. [2 points]
- (A.2.3) What is the purpose of removing tokens that occur in over 80% and under 3 headlines? How would the model be affected if we did not remove tokens in such a manner? [2 points]