EECS 487 Introduction to NLP
HW1

Assigned date: January 5th, 2023
Due date: January 23rd, 2023

Instructions:

- Please submit all the materials in one .zip file to Canvas before the due date. Name your file as hw1_<uniqname>.zip, where <uniqname> should be replaced by your unique name.
- Your .zip file should contain four files: **hw1_<uniqname>.pdf** that includes your answers to written questions and analysis questions of the programming part, **hw1.ipynb** that has the output for the programming part (please make sure that you run all cells before submission), **language_model.py**, and **naive_bayes.py**.
- Please refer to syllabus for late policy.

1 Written Questions

1.1 Knerser-Ney Smoothing [10 points]

In the lecture on language models, we introduced Kneser-Ney Smoothing for bigrams:

$$P_{\text{KN}}\left(w_i \mid w_{i-1}\right) = \frac{\max\left(C\left(w_{i-1}, w_i\right) - d, 0\right)}{C\left(w_{i-1}\right)} + \lambda\left(w_{i-1}\right) P_{\text{CONTINUATION}}\left(w_i\right) \quad (0 < d < 1)$$

where $\lambda\left(w_{i-1}\right) = \frac{d}{C(w_{i-1})}\left|\{w : C\left(w_{i-1}, w\right) > 0\}\right|$, and $P_{\text{CONTINUATION}}\left(w\right) = \frac{|\{v:C(v,w)>0\}|}{\sum_{w'}|\{v:C(v,w')>0\}|}$
(1) Prove $P_{\text{KN}}\left(w_i \mid w_{i-1}\right)$ is a proper probability distribution.
(2) Write out the equation of Kneser-Ney Smoothing for n-grams: $p_{KN}\left(w_i \mid w_{i-n+1}^{i-1}\right)$ and the corresponding $\lambda\left(w_{i-n+1}^{j-1}\right)$ and $P_{\text{CONTINUATION}}\left(w_i\right)$

1.2 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 overrated movie". You need to use unigram features with add-α smoothing, $\alpha = 0.5$. Show the calculation of each probability **step by step**.

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

Table 1: Movie review training data.

2 Programming Problems

In this part, you need to solve two programming problems about ngram language model and naive bayes classifier. 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.

2.1 Ngram Language Model

2.1.1 Coding [40 points]

See hw1.ipynb for more details.

2.1.2 Analysis Questions [7 points]

After you have completed 2.1.1, come back and answer these reflection questions.

1. (3 points) Which model produced smaller perplexity values: the word-level or character-level model? Why?
2. (2 points) Which model generates better text? The word-level or character-level model? Why?
3. (2 points) How did you choose to split your data? Were you able to correctly predict the class of the sample review?

2.2 Naive Bayes Classifier

2.2.1 Coding [26 points]

See hw1.ipynb for detailed instructions.

2.2.2 Analysis Questions [7 points]

After you have completed 2.2.1, come back and answer these reflection questions.

1. (3 points) 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?

2. (2 points) Suggest some possible ways for improving this Naive Bayes approach.

3. (2 points) 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?