CSE 4392 Special Topic: Natural Language Processing

Homework 2 - Spring 2025

Due Date: Jan 31, 2025, 11:59 p.m. Central Time

Problem 1 (Written)

The corpus provided for training the bi-gram model consists of the following four sentences:

- Cats chase after playful mice
- Birds sing at morning dawn
- Fish swim in the pond
- Dogs bark at passing cars

The corpus provided for testing the bi-gram model consists of the following two sentences:

- Cats sing and dogs swim
- Playful dogs chase birds at dawn

Question 1.1 - 15%

Define the vocabulary V corresponding to the training set shown.

Question 1.2 - 15%

Compute the matrix of bigram counts B and vector of unigram counts u. Write the probability of an arbitrary bigram $P(V_j|V_i)$ in terms of B and u.

Question 1.3 - 15%

Incorporate add-1 Laplace smoothing to B and justify why it is helpful in this particular case. Adapt u accordingly as well so that the probability is computed using the same formula you defined the previous question.

Question 1.4 - 10%

Please calculate the PPL of the **whole test set** after Laplace Smoothing.

Problem 2 (Python)

The corpus provided for training the trigram model consists of the following sentences:

- the cat watched children play in the park.
- their laughter echoed near the fragrant garden.
- the breeze spread the garden's scent into the city.
- it wafted past the cafe, famous for apple pie.
- the cafe's aroma reminded people of the nearby library.
- the library held tales of the ancient clock tower.
- the tower tolled, echoing in the quiet morning streets.
- these streets, bustling by day, were peaceful at dawn.
- at night, they lay under a star-filled sky.
- the moonlight shone on the lake where a fisherman waited.

The corpus provided for testing the trigram model consists of the following three sentences:

- the cat watched the moonlight.
- the library held the ancient clock.
- the park was peaceful in the morning.

Note that punctuation and 's are treated as separate tokens.

Question 2.1 - 30%

Define a class **TrigramModel** that has a train method that computes the trigram tensor, bigram matrix and unigram vector needed to compute any $P(w_k|w_{k-1}, .., w_{k-n})$ $n < 3$ involving words in the training vocabulary given a list of training sequences. Outside the class, write a test function that ensures the trigram counts are computed correctly using at least three examples that you compute manually.

Question 2.2 - 20%

Create an inference function that computes the conditional probability using **Linear Interpolation** smoothing technique given a word and a bigram. You can set the λ_1 as 0.5, λ_2 as 0.4, λ_3 as 0.1 for initialization. Outside the class, write a test function that ensures the probabilities are computed correctly using at least three examples that you compute manually.

You can perform smoothing to avoid undefined probabilities as a bonus but we will also accept submissions that simply ignore such probabilities by setting them to zero for simplicity.

Question 2.3 - 10%

Calculate the perplexity of the **whole test set** by adding an evaluation function to the class. Use a helper function that first computes the probability of an entire sequence using the inference function previously defined.

Your submission should include:

- Written answer to the first problem (ideally in LaTeX), but could be handwritten.
- A `.py` file for your answer to the second problem. Please note that the implemented class should not contain any information about the data (i.e., implement generically). When this file is run, it should print the test cases used in the two functions and the result, as well as the final perplexity.
- Include a screenshot of the output from running your `.py` file. This should include all the information mentioned in the last bullet.
- Convert the `.py` file to a PDF using a tool such as `https://www.i2pdf.com/source-code-to-pdf` and include it in your submission.
- Using libraries is okay as long as they do not do the probability computation end-to-end (ie, you can use libraries to construct the count matrices or equivalent datastructures to that will be used to later compute conditional and joint probabilities in inference and evaluation)