Assignment 3

CS-541: Artificial Intelligence

Spring 2022

In this assignment you will implement a markov model on three real world datasets.

In order to get full credit, you should complete both the problems without using any python scientific packages including `sklearn`, `scipy`, `nltk` etc. You can only use `pandas` or `numpy` for reading the data or for matrix operations.

1 Language Modelling

Details The task is to create an n-gram language model on different domains. The datasets include TED Talks, Reddit, and News domain. There are two training files:

- *ted.txt*
- *reddit.txt*

And three testing files:

- *test.ted.txt*
- *test.reddit.txt*
- *test.news.txt*

Part 1. [5 pts] Use the *ted.txt* to create a 1-gram language model using uniform probability. Calculate the perplexity score for the files *test.reddit.txt* *test.ted.txt*, and *test.news.txt*.

Part 2. [5 pts] Use the *ted.txt* to create a 1-gram language model using relative frequencies as probability. Calculate the perplexity score for the files *test.reddit.txt* *test.ted.txt*, and *test.news.txt*.

Part 3. [10 pts] Create two an n-gram language models using relative frequencies for files *ted.txt* and *reddit.txt*. Use values of n from 1 to 7. Apply each language model on the three test files (*test.reddit.txt* *test.ted.txt*, and *test.news.txt*) and plot the perplexity scores.

Part 4. [10 pts] Create two an n-gram language models using relative frequencies for files *ted.txt* and *reddit.txt* with Laplace smoothing ($\lambda = 1$). Use values of n from 2 to 7. Apply each language model on the three test files (*test.reddit.txt test.ted.txt*, and *test.news.txt*) and plot the perplexity scores.

Part 3. [10 pts] Generate two 500 word documents (*ted.out* and *reddit.out*) using the language models which gives the **lowest** perplexity on the in-domain test.
For example, if 4-gram language model gave the lowest perplexity for *ted.txt* and 5-gram for *reddit.txt*, then use 4-gram ted talk language model to generate the *ted.out* file and 5-gram reddit language model to generate *reddit.out* file. Compute the perplexity of the text for each.

Creating a Language Model

Given a sequence *'this is a sentence .'* language models (LM) can be calculated as follows.

Unigram LM

Probability of the sentence is the product of relative frequencies of each word.

$$P(this, is, a, sentence, .) \approx P_r(this)P_r(is)P_r(a)P_r(sentence)P_r(.)$$

$$\text{Here } P_r(w_i) = \frac{\text{Frequency of word } w_i}{\text{Total frequency of all the words}}$$

Bigram LM

$$P(this, is, a, sentence, .) \approx P_r(this)P_r(is|this)P_r(a|is)P_r(sentence|a)P_r(.|sentence)$$

$$\text{Here } P_r(w_i|w_{i-1}) = \frac{\text{Frequency of words } w_i \text{ and } w_{i-1} \text{ occurring together}}{\text{Total frequency of word } w_{i-1}}$$

Trigram LM

$$P(this, is, a, sentence, .) \approx P_r(this)P_r(is|this)P_r(a|this, is)P_r(sentence|a, is)P_r(.|sentence, a)$$

$$\text{Here } P_r(w_i|w_{i-1}, w_{i-2}) = \frac{\text{Frequency of words } w_i, w_{i-1} \text{ and } w_{i-2} \text{ occurring together}}{\text{Total frequency of word } w_{i-1} \text{ and } w_{i-2} \text{ occurring together}}$$

n-gram LM

$$P(this, is, a, sentence, .) \approx \prod_{i=1}^{n=N} P_r(w_i|w_{i-1}, \cdots, w_{i-(n-1)}) = \prod_{i=1}^{n=N} P_r(w_i|history)$$

$$\text{Here } P_r(w_i|history) = \frac{\text{Frequency of word } w_i \text{ occurring together with } history}{\text{Total frequency of } history}$$

Calculating Perplexity

Perplexity of a sequence $w_1, w_2, \cdots, w_N$ is:

$$\begin{aligned} PPL &= P(w_1, w_2, \cdots, w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1, w_2, \cdots, w_N)}} \end{aligned}$$

Here $P(w_1, w_2, \cdots, w_N)$ is the probability for the sequence calculated using the specific language model.
The smaller the perplexity value, the better the language model.

Generating Text

To generate text, use the specific language model and pick the highest probability word given the previous predictions.

To generate the **first word**, pick the word with the highest probability

$$\arg\max_{w_i} \Big\{ P(w_i) \Big\}$$

To generate **any word**, pick the word with the highest probability given the previous $n-1$ predictions

$$\arg\max_{w_i} \Big\{ P(w_i|w_{i-1}, \cdots, w_{i-(n-1)}) \Big\}$$

Here $P(\cdot)$ is the probability calculated using the specific language model.

Submission

This is an individual assignment. Each person should submit as a single zip file named with assignment number and the username (e.g. *HW2_akhan4.zip*). The zip file should contain the required code file and a readme file. The readme should include the following:

- one line descriptions of the code file
- Perplexity scores for uniform language model (part 1)
- Perplexity scores for unigram language model (part 2)
- Plot for ted talk and reddit n-gram language models without smoothing (part 3)
- Plots for ted talk and reddit n-gram language models with laplace smoothing (part 4)
- Perplexity scores for the two word documents (part 5)

Remember that after general discussions with others, you are required to work out the problems by yourself. All submitted work must be your own, though you can get help with others, so long as you cite the help. Please refer to the Stevens Honor System for clarifications.