Assignment 1

Due on March, 23 (23:59:59)

Click here to accept your Assignment 1

Introduction

Language Modeling is one of the fundamental concepts of Natural Language Processing (NLP). In this assignment, you will build some of the basic language models and generate sentences with these models.

0.1 Part 1: Basic Model

As a first step, you will implement a simple n-gram language model that allows n to vary from one to three using the **Facebook children stories dataset** as your training data. You may use various preprocesssing steps on the given dataset (removing punctuation, tokenizing punctuation, lowercasing the tokens). You are free to try out various preprocessing steps to observe the performance of your model under those operations.

0.2 Task 2: Evaluation

In this task you will use your language models to generate a new sentences and estimate the probability of the generated sentences.

To this end, you will implement a simple Ngram class with the following interfaces:

- **dataset(string folderPath):** It takes only one argument: folder path and returns the list of sentences.
- **Ngram(int n):** It takes only one argument: n, which is the order of the n-gram model and builds the n-gram model accordingly.
- **prob(String sentence):** Returns the MLE probability of the given sentence.
- **sprob(String sentence):** Returns the smoothed probability of a given sentence. Use add-one (Laplace) smoothing for the out-of-vocabulary (OOV) problem.
- **ppl(String sentence):** Returns the perplexity of the given sentence.
- **next(String word):** Samples a word from the conditional distribution of given context. For example, *next("I")* should return a random word w according to the distribution $p(w—"I")$. Use the MLE distributions for this exercise. **The history can be larger than a single word based on the n value.**

- **generate(int length, int count):** It takes two arguments *length* which is the maximum length of a sentence (e.g. 20) and *count* which is the count of the sentences that will be generated and returns the generated sentences.

What is Perplexity?
Perplexity is the inverse probability of the set, normalized by the number of words.

$$PP(W) = P(w_1 w_2 \cdots w_N)^{-\frac{1}{N}} \quad (1)$$

$$= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_1 w_2 \cdots w_N)}} \quad (2)$$

When we use the log probabilities for the calculation, perplexity is calculated as follows:

$$PP(W) = 2^{-\frac{1}{N} \sum_{i=1}^{N} log_2 P(w_1 w_2 \cdots w_N)} \quad (3)$$

Regarding the perplexity, you may look at this blog.

0.3 Dataset

The dataset is released by Facebook that is used to train artificial intelligence software to understand childrens stories and predict the word that was missing from a given sentence in a story. In this assignment , we will use a part of this dataset to build our language model. Dataset link

You can get details of the dataset from the link.

Submit

You are required to submit all your code. You will implement the assignment in **Python** (Python 3.5). You will submit a report in latex format template). The codes you will submit should be well commented. Your report should be self-contained and should contain a brief overview of the problem and the details of your implemented solution. Give the answers of all questions raised in the definition of the assignment above. You can include pseudocode or figures to highlight or clarify certain aspects of your solution.

- report.pdf (Report Template)
- code/ (directory containing all your codes as Python file .py)

The ZIP file will be submitted via Github Classroom. Click here to accept your Assignment 1

Grading

- Code: 85 points
- Report: 15 points

> **Note**: Preparing a good report is important as well as the correctness of your solutions! You should write your results for each part of the task and answer the questions raised in the related sections (We expect data structures that you used to build your language model, Error analysis of your generated sentences, calculation of perplexity and results of sentence generation).

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.