Assignment-1 Problem Statement- II

14/07/2024

10 Points Possible

Attempt 1

In Progress

NEXT UP: Submit assignment

Add comment

Unlimited Attempts Allowed

28/06/2024 to 14/07/2024

Details

General Instructions

1. Each group is expected to submit jupyter notebook (.ipynb) with output for each cell. The **name of the assignment** file must be the **Group ID**. For Example. NLP Group 123.ipynb.
2. Please mention the contribution of each group member (NAME, ID, CONTRIBUTION).
3. No extension on the deadline.
4. As it is a group assignment, only one group member needs to submit the assignment on behalf of the group.
5. Submissions using other Python IDEs will not be considered for grading.

Training Data

We'll be using the English news Corpus (2019 year) as our training data. There are around 10,000 sentences. eng_news_2019_10K-sentences.txt (https://bits-pilani.instructure.com/courses/2826/files/548834?wrap=1) (https://bits-pilani.instructure.com/courses/2826/files/548834/download?download_frd=1)

Problem: N-gram language model

You need to build 2 language models, the Bigram model, and the Trigram model using Laplace smoothing. With each model, you will do the following tasks:

1. Display 10 generated sentences from this model.
2. Score the probabilities of the provided test sentences and display the average and standard deviance of these sentences.
   1. once for the provided test set
   2. once for the test set that you curate

Part 1: Build an n-gram language model (6 marks)

1. Pre-processing of the data.
2. Split the data for training and testing.
3. You need to develop an n-gram model that could model any order n-gram, which we'll be using specifically to look at bigrams and trigrams. Specifically, you'll write code that builds this language model from the training data and provides functions that can take a sentence in

(formatted the same as in the training data) and return the probability assigned to that sentence by your model.

4. Handling of unknown words and smoothing.
5. Evaluating the language model.

Part 2: Implement Sentence Generation (4 points)

In this part, you'll implement sentence generation for your Language Model. Start by generating the <s> token, then sampling from the n-grams beginning with <s>. Stop generating words when you hit an </s> token.

Notes:

1. When generating sentences for unigrams, do not count the pseudo-word <s> as part of the unigram probability mass after you've chosen it as the beginning token in a sentence.

1. All unigram sentences that you generate should start with one <s> and end with one </s>

1. For n-grams larger than 1, the sentences you generate should start with $n{-}1$ <s> tokens. They should end with $n{-}1$ </s> tokens.

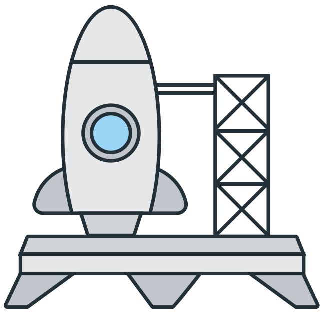
Justification of the output obtained for all the above tasks is mandatory

! Keep in mind, this submission will count for everyone in your Assignment Groups group.

Choose a submission type

Upload

Arc



Choose a file to upload