CMP 670 - Statistical Natural Language Processing

Assignment 1 - N-gram Language Models

(Due date: 21.03.2019 - 11:59 pm)

Part 1: Basic Model

As a first step, implement a simple n-gram language model that allows n to vary from one to three. Your code should be able to estimate the probability of a given sentence and generate a new sentence.

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

- `Ngram():` It takes only one argument: n, which is the order of the n-gram model.
- `prob():` Returns the MLE probability of the given sentence.
- `sprob():` Returns the smoothed probability of a given sentence. Use Laplace smoothing for unigrams and Good-Turing smoothing for higher order n-grams.
- `ppl():` Returns the perplexity of the given list of sentences.
- `next():` 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.

You need to perform two tasks using the n-gram class interface.

1. Print out perplexity score of the given test file using unigram, bigram and trigram model.
2. Using each of the n-gram model, generate 5 random sentences with the next() method you wrote. Adjust the maximum length of the sentences up to 20 words.

Now take a look at the type of errors your language model is making. The next two parts of the assignment are designed to target these errors. Analyse these errors and write them in your report.

Part 2: Linear Interpolation

After building the simple model, your next task is to build a model that uses linear interpolation. You will apply linear interpolation for a bigram and trigram language models. In this task, you will need to decide how to set the parameters (i.e. λ_1, λ_2, λ_3). Use the development set for this purpose. See the Collins notes and lecture slides for some suggestions.

Again print out the perplexity scores of the trigram model and compare it with the previous results. Comment on the compared results in your report.

Part 3: Discounting

You will use discounting in the last task for smoothing. Again you need to decide how to set the parameters using the development set. You will need to apply both bigram and trigram language models using discounting method.

Once you implement each model described above, you will (1) compare the basic model with each of the smoothed models and (2) compare the smoothed models with each other. Perform some tests on some examples to be able to compare each model.

About the report

Your report should be no more than 3 pages long, and should include:

1. Describe how you set the parameters in each model.

2. Compare different models using quantitative and qualitative analysis.

3. Error analysis – in different models, what are the typical errors you could observe? How can you fix those errors? Give some potential solutions to those problems.

What to submit?

You should submit your implementation files, a README file with instructions on how to run your code, and report (as a pdf file).

Implementation details

Please use only Python 3.0 for your implementation. The datasets and other details regarding the submission of your assignment will be shared through Piazza.