M2 Project Sketch

Prachatorn Joemjumroon and Sam Briggs

May 20, 2021

1 Perplexity

Perplexity is a metric that we use to describe the probability of an unseen test set. We are given that perplexity is:

$$= \sqrt[N]{\prod_{1}^{N} \frac{1}{\mathbb{P}(w_i | w_{i-1} \dots w_1)}}$$

For this assignment, we will be taking the log of all the probabilities, and thus we get something that looks like entropy:

$$-\frac{1}{N} \sum_{i=0}^{M} \log P(s_i)$$

Where N = the total count of words within the unseen test set, M = the total number of sentences, and s_i is the n-gram probability. We see that because this the inverse of the probabilites normalized by the number of words, minimizing perplexity maximizes the probability of a test set.

2 Progress thus far

1. What is a new skill/concept that we have learned?
   First, we learned what an n-gram is, and why we use them. We also learned how to calculate various things on them, like perplexity by assignment 3 and lecture 11. We also learned how to calculate the probability of an n-gram, Baye's theorem, why we $< unk >$ values so that we have a value for unseen data, why we invoke the Laplace smoothing algorithm to avoid a divide by 0 error.

2. Is there anything that you are currently having trouble with (e.g., bugs, confusing algorithm components)?

We have not started coding, but we plan to start on Sunday, but we did looked over at the skeleton of the program. We have some confusion about the ArgumentParser package in the main.py and what it does in the code. Also Prachatorn was confused how to implement a nested dictionary for both the bigram and the trigram in the code.

3. Are there any skills you still need to build?

   We need to learn not to procrastinate lol. But we need to read the documentation for starter code that we don't understand, as well as ensure that we have a good grasp of how to compute the unigram/bigram/trigram probabilities. I am fairly confident in our coding skills

3 Revised Timeline

- Sunday, May 23, 2021: Start coding aspect of the assignment.
- Monday, May 24, 2021: Finish Unigram.py
- Wednesday, May 6, 2021: Finish bigram.py
- Friday, May 28, 2021: Finish the trigram.py, thereby finish the coding base aspect of this assignment.
- Wednesday, June 2: Finish coding extension of project.
- Friday, June 4: Finish project deliverable, thereby finishing whole project