NLP 2023 Assignment #1: Language Modelling and N-grams

Rowan University

January 17, 2023

Instructions for submission: all code and results should be packaged and zipped. For large outputs, please save them as pickled files. Code should include enough information for me to reproduce your results. When questions ask for a specific answer or explanation, written answers should be included in a PDF formatted '{first_name}_{last_name}_HW1.pdf'.

1. **Problem 1** Write Python functions to compute unsmoothed unigrams and bigrams (HINT: the collections.Counter and collections.defaultdict methods will be useful here) without using the NLTK package.

2. **Problem 2** Download the scripts for preparing two text corpora from our course Github site and run them in the HW1 directory.

   (a) What is the corpus size (overall number of tokens) and the vocabulary size for each corpus?

   (b) Run your code from the prior problem over these two corpora. Sort the bigram statistics. What are the top 10 most frequent bigrams in each corpus?

   (c) For each corpus, provide the unsmoothed bigram probability for the pair 'hath' and 'discretion'.

   (d) What are the differences in the common unigrams between the two corpora?

   (e) What are the similarities and differences in the bigrams between the two corpora?

3. **Problem 3** Modify the functions in Problem 1 to take and apply the following optional arguments:

   (a) Define a new function to generate a random sentence using the unsmoothed bigram statistics (i.e. uses the bigram statistics to generate a new sentence, not a sentence randomly selected from the training data). This function should additionally have an option to set the max sentence length. (HINT: use Python's random module). Make sure sentences start and end with the correct start and end tokens.

   (b) Set your random seed to 17 and output a random sentence using the function defined above with sentence length equal to 10 and the bigram model from the Hamlet corpus. Print the tokens of this sentence in your PDF write-up.

   (c) Define a function that takes on input a sentence (like the random one above) and the bigram model to compute the perplexity. What is the perplexity of your random sentence?

4. **Problem 4**

   (a) Add a flag to the unigram and bigram functions turn on/off Laplace smoothing.

   (b) Compare the resulting unigram and bigrams with and without Laplace smoothing. What similarities and differences do you notice? How did these conclusions change from your answers to Problem 2?

   (c) For each corpus, provide the smoothed bigram probability for the pair 'hath' and 'discretion'.

   (d) Run your random sentence through both models, which has the lower perplexity? Why?

   (e) What are good alternates to Laplace smoothing?