Assignment 1

Nathan Cho
nc5123@bard.edu

1. Part 1: N-Gram Models

Implementation

For Part 1, the `blake-poems.txt` corpus was used. All 4 models used `nltk.lm.preprocessing.padded_everygram_pipeline` to create n-grams and vocabulary for the language model.

Unigram, bigram, and trigram language models used `nltk.lm.MLE`, a maximum likelihood estimator model provided in the NLTK package. Unigram with tf-idf model was implemented with custom class `TfIdf`, which overrides the default scoring function `unmasked_score` to score each word with tf-idf formula. All 18 books in the Gutenberg Corpus were used as documents for tf-idf computation.

Results

Sentence Generation

Language model	Generated sentences (12 words, detokenized)
Unigram	was voice,, sweetest of maid THE in in heat; I, and;, are thy sigh put Frame for Nor kiss' Near GIRL of me thee, because, FOUND dwelling
Unigram (tf-idf)	watery walked Bowed Dare skylark morn joy Secresy frowning fruitful fade INFANT Motto Earth Weeping IMAGE Chase Willm tired river pearly Maker delight Printer heaven APPENDIX Prays Maker morn knits thee Deceit ancient Among Lyca clay
Bigram	wanderer, And I turned to see In the morning dew, A SONG, Or think they seemed, We ' human Brain . nibbled, sweeter smile, ON
Trigram	virtues of delight! IV. hand?

Perplexity

Tested sentence	Language Model			
	Unigram	Unigram (tf-idf)	Bigram	Trigram
He replied that he had not.	456.4	inf	inf	inf
You do not know what I suffer.	inf	inf	inf	inf
Do you bite your thumb at us, sir?	inf	inf	inf	inf
Forget to think of her.	inf	inf	inf	inf
The white kitten had had nothing to do with it.	inf	inf	inf	inf

To further verify that the perplexity is being computed correctly, the perplexity of 2 sentences from the corpus and 4 sentences generated by each language model were computed.

Tested sentence	**Source**	**Language Model**			
		Unigram	**Unigram (tf-idf)**	**Bigram**	**Trigram**
Where a thousand fighting men ...	**Corpus**	986.3	inf	7.3	inf
Night and morning with my tears,	**Corpus**	219.1	inf	23.8	inf
kiss' Near GIRL of me thee, ...	**Unigram**	**459.8**	inf	inf	inf
heaven APPENDIX Prays ...	**Unigram (tf-idf)**	2583.9	**496.5**	inf	inf
human Brain . nibbled, ...	**Bigram**	639.1	inf	**inf**	inf
hand?	**Trigram**	345.4	inf	inf	**inf**

2. Part 2: Expanding the Corpus

Implementation

For Part 2, all 18 books in the Gutenberg Corpus were used. All 3 models are identical to Part 1.

Results

Sentence Generation

Language model	Generated sentences (12 words, detokenized)
Unigram	were was,, the round of all live load it: Isaiah . but:, called tract teach she Honourable him Till my, They I ruminant of through, counsel, Have happened
Bigram	well versed in THAT voices of the drunkard found it is all " Let her, and upon thee in consequence in haste made, and come to rock or a letter, and money
Trigram	way up as a strange speech and my mother; but I 17: 12 All the unaccomplished works of engineers, Our it, and come to public disgrace if Franklin had not had

Perplexity

Tested sentence	Language Model		
	Unigram	Bigram	Trigram
He replied that he had not.	251.5	36.2	inf
You do not know what I suffer.	507.0	30.9	9.8
Do you bite your thumb at us, sir?	1486.5	inf	inf
Forget to think of her.	494.7	inf	inf
The white kitten had had nothing to do with it.	669.2	inf	inf

To further verify that the perplexity is being computed correctly, the perplexity of 2 sentences from the `blake-poems.txt` corpus and 3 sentences generated by each language model were computed.

Tested sentence	Source	Language Model		
		Unigram	Bigram	Trigram
Where a thousand fighting men ...	**Corpus**	2734.2	109.0	8.4
Night and morning with my tears,	**Corpus**	583.0	354.7	inf
my, They I ruminant of through, ...	**Unigram**	**775.9**	inf	inf
made, and come to rock or a ...	**Bigram**	283.2	**inf**	inf
it, and come to public disgrace ...	**Trigram**	573.7	inf	**inf**

3. Part 3: Reflection

Part 1: Differences Between Models

The most notable difference between the four different models was the naturalness of the generated sentences. Unigram generated a sequence of words without any coherent pattern, but bigram generated more coherent words next to each other. On the other hand, the trigram model failed to generate longer sentences. This seems to be caused by the number of sentence-ending punctuations, such as trigram (`“.”, </s>, </s>`) will exist for all sentences ending with a period, which increases the likelihood of ending the sentence right away.

Increasing the length of context resulted in generating more natural sentences but also increasing the perplexity much more. This higher perplexity is due to the nature of n-gram perplexity computation, as it requires the language model to be trained on a corpus including the exact sequence of words (i.e., n-gram) to have low perplexity. Shown in `part1_results.txt`, all sentences excluding “He replied that he had not.” included unknown words due to the small size of the corpus (i.e., a word not in the vocabulary of the language model), resulting in infinite perplexity. This can be resolved with smoothing techniques.

Additionally, the trigram failed to output low perplexity on the sentences directly from the corpus. This seems to be due to the structure of the original corpus, as the `blake-poems.txt` corpus had whitespaces and different indentations before the sentences, being a book of poems.

Using unigram with tf-idf score generated a sequence of words more frequently found in the original corpus, and less likely to be found in other documents (i.e., other books in the Gutenberg Corpus). We can see this in the word “APPENDIX” in the third sentence, which only appears in the `blake-poems.txt` corpus.

Part 2: Using Larger Training Set

Using the entire Gutenberg Corpus as the training data has decreased the perplexity of the given sentences, but didn’t change which model performed the best. Although the sample size was small, both in Part 1 and Part 2, bigram generated the most natural sentences between all models.

Improving N-Gram Models

One change to improve the n-gram language models would be removing non-semantic elements during training. Non-semantic elements could include punctuation, capitalization, specific stop words, etc. This could improve the language model by generating more natural sentences for applications that don't require strict grammatical correctness.