William Powers A59008589

Language Modeling Report

Introduction

The goal of this experiment is to build probabilistic language models and use them to analyze texts from the following three corpora: The Brown Corpus[1], The Gutenberg Corpus[2], The Reuters Corpus[3]. First, we start with a simple unigram model as an initial language model that does not incorporate context and does not perform well. It creates a marginal probability distribution for in-vocabulary words by obtaining a word's empirical frequency within the corpus of words. It also includes an EOS token to end each sentence in the corpus. Whereas with out-of-vocabulary words it assigns a parameter value, here being set to 0.000001. For our measurement of the accuracy of a model, we will be calculating the perplexity of the constructed probability distribution on the trained corpus as well as other corpora. The object then for optimizing our model is to reduce its perplexity defined as the exponential of the cross-entropy. To find the probability of the corpus we will multiply the probabilities of the sentence probabilities, to find the joint probability of the words in the sentence, we will expand the sequence probability with the chain rule and instead of conditioning on all the previous words, we will condition on the n-1 words, where n is the length of the n-gram model we are using.

Unigram Language Model Analysis

Perplexity Results on Training Data

Corpus	# Training Words	Test Set Perplexity
Brown	693683	1604.20
Reuters	965053	1500.69
Gutenberg	1471033	1005.79

Empirical Analysis of In-Domain Text

On the right, we can see the results on our unigram model and the corpus sizes they were trained and tested on. For our final analysis of the model, we will of course be examining the perplexities of the model on the Test Set for the respective corpus. The order of perplexity scores from lowest (best) to highest are: 1. Gutenberg, 2. Reuters, 3. Brown. Let us note the relative sizes of the training corpora for each on the models. We can see the number of words in the training corpora for the models follow the similar ordering from least to most as the perplexity scores: 1. Gutenberg, 2. Reuters, 3. Brown. This is likely due to the availability of randomized training data available to get the empirical frequencies for creating an accurate unigram probability distribution.

To examine the results, we can see how the test set perplexity score of a model on a randomly shuffled corpus changes depending on the size of the corpus while training. The plot on the right details that relationship.

As we see, the perplexity of the model on the training set decreases as the size of the training data increases. However, we can see that still, the Gutenberg training corpus outperforms when smaller than the other training corpora. This is likely due to the corpus composition, such that words in test sections of the corpus have high empirical frequency in train sections. Indeed, the Gutenberg corpus is comprised of famous literature from repeat and similarly cultured authors, detailing mostly

[1] http://www.hit.uib.no/icame/brown/bcm.html
[2] http://gutenberg.net/
[3] https://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection

interpersonal affairs. Whereas the Reuters and Brown corpora were constructed purposefully to have varied resources within them.

Empirical Analysis of Out-of-Domain Text

Below we see the perplexity scores of 3 unigram models trained on different corpora and tested on different corpora. Of course, the models perform best on their own corpora.

Perplexity Results on Test Data across Corpora

Model	Brown Corpus	Reuters Corpus	Gutenberg Corpus
Brown	1604.2	6736.6	1762.01
Reuters	3865.16	1500.69	4887.47
Gutenberg	2626.05	12392.5	1005.79

The Brown model and the Gutenberg model had their second-best performance on each other's Corpora, while the Reuters model had its second-best performance on the Brown Corpus. The results are the same when analyzing which model was the best performing on a particular corpus. This suggests that the vocabulary and probability distribution of the Brown and Gutenberg corpora are more similar to each other than to the Reuters. It also suggests that the Reuters corpus is more similar to the Brown corpus than the Gutenberg corpus.

Context-aware LM: Implementation

To improve upon the simple unigram model, we will be implementing a new model that will incorporate the sentence context when predicting the probability of the next word. Here, we will be prepending sentences with an SOS token and appending it with an EOS token to signify/predict the boundaries of a sentence. We will lowercase all words as tokens are case-sensitive. We will use a trigram model where the probability of the next word in a sentence is conditioned on the previous bigram (or just the SOS token if it is the first word in a sentence.

For predicting the conditional probability, we will not simply be creating a maximum likelihood estimate by dividing the frequency of the full word sequence by the frequency of the previous bigram. This will result in some conditional probabilities being zero, which will make the perplexity score infinity. To fix this, we will need to do two things.

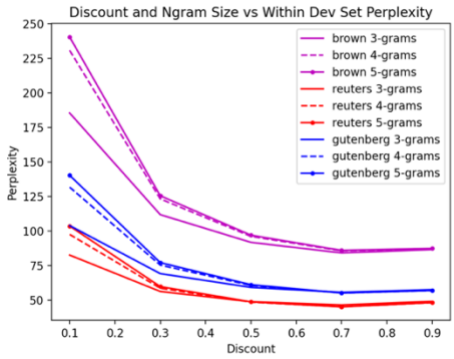
First, we will not easily be able to preemptively create probabilities for words we have not yet seen before. To handle out-of-vocabulary words, when creating our model, we will replace words with a frequency of one with the UNK token. Then when evaluating the perplexity of our model on a new corpus, we will replace the out-of-vocabulary words with the UNK token, to get around issues with evaluating perplexity. We believe it is fair to replace words only seen once in a corpus, as they must not be very substantive to the language of the corpus, and it will be nearly impossible to create a probability distribution to understand the context they should be used in.

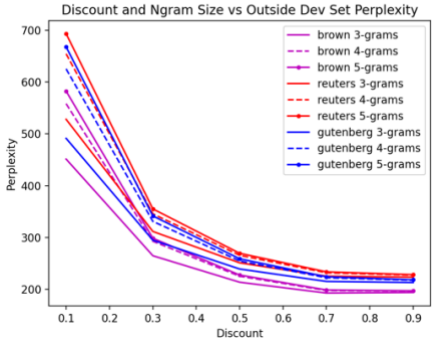
Second, we will implement the Katz Back-Off smoothing method to redistribute the probability of frequently seen n-grams to unseen n-grams comprised of in-vocabulary words. For this, we'll define some values and terms. For discounting seen probabilities, we have a discount value (*DV*). Further: $\text{n-gram} + \text{word} = n\text{-}gram\ appended\ by\ word.$ $\text{n-gram} - 1 = \text{n-gram}$ excluding the first word. $Count(\text{n-gram}) = \text{n-gram's training corpus frequency.}$ $Count^*(\text{n-gram}) = Count(\text{n-gram}) - DV.$ $ML(\text{word}, \text{n-gram}) = Count(\text{n-gram} + \text{word}) \div \text{Count(n-gram)}.$ $DML(\text{word}, \text{n-gram}) = Count^*(\text{n-gram} + \text{word}) \div \text{Count(n-gram)}.$ $Suffix(n\text{-}gram) = \{word: Count(n\text{-}gram + word) > 0\}.$ $\text{MPM(n-gram)} = 1 - \sum_{w \in Suffix(\text{n-gram})} DML(\text{w}, \text{n-gram}).$

After defining Maximum Likelihood Estimation (ML), Discounted Maximum Likelihood Estimation (DML) and Missing Probability Mass (MPM), among others, we can define the conditional probability for Katz Back-Off, $p(\text{word} \mid \text{n-gram}) = KBO(\text{word, n-gram})$ where:

$$KBO(\text{word}, \text{n-gram}) = \begin{cases} DML(word, n\text{-}gram) & if\ count(n\text{-}gram + 1) > 0 \\ \frac{MPM(\text{n-gram}) \times KBO(\text{word, n-gram}-1)}{\sum_{w \notin Suffix(\text{n-gram})} KBO(\text{word, n-gram}-1)} & if\ n > 1 \\ \frac{MPM(\text{n-gram}) \times ML(\text{word})}{\sum_{w \notin Suffix(\text{n-gram})} ML(\text{w})} & if\ n = 1 \end{cases}$$

The hyperparameters available in our model are the discount value and the length of the maximum n-grams available. To tune those parameters, we repeatedly trained the model on each corpus and tested it on dev sets, while changing the discount values between .1, .3, .5, .7, .9, and changing between using 3, 4, and 5-grams. The graphs below show the results for in-corpus perplexity and the sum of out-of-corpus perplexities.





Perplexity Results on Training Data

Model	Brown Corpus	Reuters Corpus	Gutenberg Corpus
Brown	20.44	103.798	90.798
Reuters	114.759	21.6611	106.566
Gutenberg	102.874	110.547	21.3729

Perplexity Results on Dev Data

Model	Brown Corpus	Reuters Corpus	Gutenberg Corpus
Brown	86.1442	104.363	90.4674
Reuters	113.861	49.48	107.554
Gutenberg	102.413	111.24	57.9311

Perplexity Results on Test Data

Model	Brown Corpus	Reuters Corpus	Gutenberg Corpus
Brown	86.5854	103.524	90.2808
Reuters	116.38	49.0677	106.992
Gutenberg	103.278	109.734	57.5106

Since we want a model that generalizes well, we will be choosing the one with the best out-of-corpus perplexities. Thus, we will be choosing a 3-gram model with a .9 discount model as it performed best for Reuters and Gutenberg. Brown performed better with a .7 discount but for consistent analysis across models we will be using its second-best discount, .9. To the left, you can see performances for our tuned model across train, dev and test sets.

Sampled Sentences

To sample words for creating sentences from this probability distribution, we will start with the SOS token as the previous n-gram, and to select a new word we will collect the conditional probabilities for each vocab word given the previous n-gram. We will augment each probability p with a temperature value t by p^t. This will have the effect of amplifying the relative weight of larger probabilities over smaller ones. We will then normalize the probabilities by dividing them by the sum of them all. We will then select the words with the largest probabilities summing to a certain threshold percent. This will remove the unlikeliest of words. To sample words, we will randomly choose between each remaining word weighted by its new probability. We will continue this process until we reach an EOS token. Further, we will skip SOS tokens, UNK tokens, and numbers. Number tokens appear frequently in the Gutenberg text, which produces poor samplings. This is likely due to bible verse numerations and thus poor data cleansing.

Corpus	Sentence	Probability
Brown	the speech fall of the people of the state of	1.3e-18
Reuters	the speech to the company said it has agreed to	1.4e-16
Gutenberg	the speech of his own of the children of the	8.6e-18

Using the sentence sampler, we will be sampling sentences of length 10 starting with the prefix "The speech". To choose the best temperature and percentage threshold, we iterated over many Samplers with different values and optimized for the sum between corpora of the average sentence joint probability for 3 fixed-length samples. The best values were a temperature of 1.1 and a threshold of .1 and those are the ones we will be using. Above are the sampled sentences and their joint probabilities. Their relative probability differences are quite random over many samplings so there is little pattern to analyze between them.

Empirical Analysis of In-Domain Text

The perplexity scores of the models are shown above, as well as the hyperparameter tuning performance and sampled sentences.

Looking at the in-corpus test set perplexities above, the Reuters corpus now has the lowest perplexity, with Gutenberg second and the Brown corpus still having the highest perplexities. If we were to hypothesize why this is the case, we can first rule out corpus size, as shown above. Second, adding to our previous theory that it has to do with the similarity of training and test sets, perhaps Reuters beat Gutenberg this time, because when the context of surrounding words was given, the training and test sets ended up being far similar, whereas with the Gutenberg texts the surrounding contexts (n-grams) of the training set were less similar to the contexts (n-grams) in the test set and thus it was less helpful for word prediction. This could be that the Reuters texts were news articles from various unrelated topics, but once the topical context was assessed, the article texts in those topics were very similar. Also, it is hard to compare the unigram and trigram models as the trigram model replaced low-frequency and unknown words with UNK whereas the unigram model had a standard low percentage to assign to all unknown words. Perhaps, the Reuters model did better because it had the most low-frequency words replaced, thus decreasing the perplexity. The chart below proves that it did have the most words replaced.

Empirical & Qualitative Analysis of Out-of-Domain Text

The perplexity scores are shown above. The trigram model beat the unigram model for every perplexity score. Again, the Gutenberg and Brown models had the best out-of-domain performance on each other's corpora and their corpora had the best out-of-domain performance on each other's models. The Reuters Model's corpus has the best out-of-domain performance with the Brown model. However, this time, the Reuters model had the best out-of-domain performance data with the Gutenberg data. It is hard to compare the unigram and trigram models when different methods for low-frequency/unknown words were used**.** What is likely to have caused this change is that the Reuters corpus has the most low-frequency words, creating a high probability for UNK tokens and the Gutenberg corpus has the most words unknown to the Reuters model, creating a high occurrence of the UNK tokens, thus artificially creating a low perplexity. And we can see from the chart to the right that is correct.

Number of Low-Frequency Words

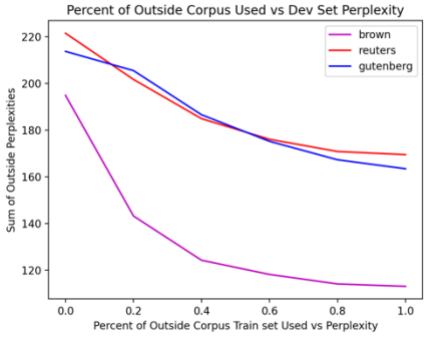
Brown	Reuters	Gutenberg
92,205	202,424	170,608

Number of Words replaced with UNK

Model	Brown Corpus	Reuters Corpus	Gutenberg Corpus
Brown	10,570	38,277	27,738
Reuters	23,301	14,412	61,374
Gutenberg	17,533	52,435	15,305

Adaptation

To adapt a model trained on one corpus for another corpus, the only apparent solution is to additionally fit the model to some of the other corpus as well, combing their counts for the Katz Backoff algorithm. To test how much of the secondary corpus should be added, we have plotted the dev set perplexities to the right as different percentages of the train set are added to the out-of-domain model training set. The best solution then is to add 100% of the secondary training corpus.



In fact, in the chart to the right, we can see that the models perform better for all test corpora when their training set is added during model fitting. Perhaps this is due to an increased vocabulary count, reducing the low-frequency and unknown words that give the model a better structural understanding of the English language. It also increases the total amount of training data, which we have seen before always seems to decrease the perplexity.

Combined Perplexity Results on Dev Data

Model	Brown Corpus	Reuters Corpus	Gutenberg Corpus
Brown	----	52.8	60.3
Reuters	104.2	----	65.2
Gutenberg	102.7	60.8	-