CSE 156 Assignment 1 Report

David Dai

April 17, 2019

1 Code

The code files with the README are zipped as A1.zip and uploaded to GradeScope

2 Language Model Implementation

2.1 Model Description

For the project, I used trigram language model with laplace smoothing. Trigram language model is part of the statistical language model which is calculating a probability distribution over all the words. As for the trigram language model, the current word probability is calculated based on the previous two words. The technical term to calculate each word's probability in the trigram language model is called Maximum Likelihood Estimated (MLE). For example, if we want to calculate the probability of the third word in a sentence (A B C), the probability of word C is based on the probability of word A and word B. The exact MLE calculation is $q(C|A,B) = \frac{c(A,B,C)}{c(A,B)}$. $c(A,B,C)$ is the number of times that the sequence A, B, C appeared in the sentence or training corpus; $c(A,B)$ is the number of times that the bigram sequence A,B appeared in the sentence or training corpus. This situation above only works for the current word position is bigger than 2. Thus, it is important for this model to insert two start symbol at the front of each sentence in the training corpus. In my implementation of the trigram language model, specifically at the fit sentence function, I inserted two starting symbol ('*') at the start and one end of sentence symbol ('END OF SENTENCE')(EOS) at the end of the sentence. With my implementation the sentence (A B C) will be transformed into (* * A B C EOS). The first and second word will be calculated as $q(A|*,*) = \frac{c(*,*,A)}{c(*,*)}$ and $q(B|A,*) = \frac{c(*,A,B)}{c(*,A)}$ In the training corpus, it is equally important to know the when to end the sentence; with my implementation, it will be $q(EOS|B,C) = \frac{c(B,C,EOS)}{c(B,C)}$.

2.2 Smoothing Method Description

As I have state in section 2.1, I implemented a trigram language model with laplace smoothing. For all the trigram language model words, they are sorted as categorical data and each data is represented by a probability. The trigram language model, however, tends to fall into the pattern of many categorical data, which has a spiky distribution. Thus, with the laplace smoothing and trigram language model, the spikey distribution would be smoothed. The fundamental idea is to manipulate each word's probability with the an alpha value (α) to adjust the equation. Using the example of a three-word sentence of (A B C). Without laplace smoothing, the $\alpha = 0$ and the probability of the last word is $q(C|A, B) = \frac{c(A,B,C)}{c(A,B)}$. With the fundamental laplace smoothing, we set $\alpha = 1$ and the probability of the last word becomes $q(C|A, B) = \frac{c(A,B,C)+1}{c(A,B)+1*|V|}$. However, this fundamental laplace smoothing tend to reassign too much weight to all the unknown symbols (UNK). If we set α between 0 and 1, then we can avoid the scenario above. The α-adjusted laplace smoothing is implemented in the cond logprob function with the equation $q(C|A, B) = \frac{c(A,B,C)+\alpha}{c(A,B)+\alpha*|V|}$. In my implementation, the α is set to 0.2 and $|V|$ is the total number of keys in the dictionary. $|V|$ is assigned in the norm function and sets the length of the vocabulary dictionary after training as $|V|$.

2.3 Hyperparameter tuning and data

For the data, we have three distinct data sets or corpus. We have the Brown Corpus, Gutenberg Corpus and the Reuters Corpus. Each corpus is also divided into three components, train, dev, and test. The trigram language model is trained on the training part of the corpus; the model then is validated with the dev component of the corpus; and lastly, the model is tested with the testing part of the corpus.

As for the hyperparameters, one of the obvious hyperparameter to be tuned in this language model is the α in the laplace smoothing. The α value is between 0 and 1 and the corresponding $|V|$ is normalized by the α value as well.

3 Analysis of In-Domain Text

3.1 Perplexity Value

The perplexity value from the unigram and trigram regarding the three corpus, Brown corpus, Reuters corpus, and Gutenberg corpus are showing in table 1, table 2, and table 3.

Brown	train	dev	test
Unigram	1513.8018008490042	1589.3868225664532	1604.198220472703
Trigram	792.6259528352632	827.5737251951878	823.1565500048324

Table 1: Brown Corpus Perplexity

Reuters	train	dev	test
Unigram	1471.209708062019	1479.0930760437175	1500.6949333808543
Trigram	892.2259863095398	919.7897877036229	923.0082903481521

Table 2: Reuters Corpus Perplexity

Gutenberg	train	dev	test
Unigram	982.5718078017832	991.5002070957079	1005.7898465085633
Trigram	714.1346332790389	722.8554947231961	724.98100331883679

Table 3: Gutenberg Corpus Perplexity

3.2 Unigram Model vs. Trigram Model

As section 2 Language model implementation, the trigram language model has already been introduced. The above table 1, table 2 and table 3 are showing the perplexity value of the trigram and unigram language model. Unigram is an n-gram model when n equals to 1; on the other hand, trigram is an n-gram model when n equals to 3. In the unigram language model setting, all the words are independent to each other; there is no correlation between one word and the other. The only thing used to count the word is the number of appearances of each word in the corpus. As for the idea of complexity, it is calculated with entropy. Entropy is the average of the log probability of all the words in all the sentences of a corpus. For example, in a corpus' testing data set, there are m sentences consists of $x_1, x_2...x_m$. Then, entropy $= l = \frac{1}{M} * \sum_{i=1}^{m} \log_2 p(x_i)$. Perplexity is calculated by $2^{-entropy} = 2^{-\frac{1}{M} * \sum_{i=1}^{m} \log_2 p(x_i)}$.

3.3 Hyperparameters Tuning

To tune the unigram and trigram model to receive reasonable perplexity value, I changed the α value from the laplace smoothing and the normalized $|V|$ respectively. More importantly, I tuned the backoff value in the init function. The reason that i changed the backoff value for trigram is the value is unreasonable. The goal of having backoff value is for all the UNK times the backoff value to reach 1 or bigger than 1. However, in this case, the backoff value is originally set to 0.000001 and thus it needs 6000000 UNK to satisfy the statement above. Because of the backoff value is off, the perplexity of the trigram is always off from the ideal output. Thus, as table 4 is showing, by tuning the backoff value to 0.008, the perplexity value became more reasonable.

backoff value	train	dev	test
0.000001	338029.99112484214	346073.33677695679	348047.5843252652
0.008	714.1346332790389	722.8554947231961	724.98100331883679

Table 4: Trigram Perplexity for Glutenberg

x train	trimodel	trimodel	trimodel	x train	unimodel	unimodel	unimodel
	brown	reuters	gutenberg		brown	reuters	gutenberg
brown	828.777	1053.19	933.368	brown	6780.82	6780.82	1758.06
reuters	962.312	932.86	1036.59	reuters	3806.39	1471.21	4882.8
gutenberg	906.789	1078.42	803.796	gutenberg	2616.57	12420.1	982.572

Table 5: Trigram vs. Unigram x train

x dev	trimodel	trimodel	trimodel	x dev	unimodel	unimodel	unimodel
	brown	reuters	gutenberg		brown	reuters	gutenberg
brown	861.674	1053.98	932.551	brown	1589.39	6675.63	1739.41
reuters	964.102	957.948	1036.82	reuters	3808.87	1479.09	4833.88
gutenberg	908.069	1077.24	813.054	gutenberg	2604.28	12256.3	991.5

Table 6: Trigram vs. Unigram x dev

x test	trimodel	trimodel	trimodel	x test	unimodel	unimodel	unimodel
	brown	reuters	gutenberg		brown	reuters	gutenberg
brown	857.757,	1057.68	933.386	brown	1604.2	6736.6	1762.01
reuters	963.689	960.664	1034.82	reuters	3865.16	1500.69	4887.47
gutenberg	909.025	1081.51	813.87	gutenberg	2626.05	12352.6	1005.79

Table 7: Trigram vs. Unigram x test

3.4 Examples of sampled sentences

Unigram sample sentences are the following:
sample 1: now for the mln the pct Inc said cts in Group DLRS BankEast acquring reform shrs effect from it slaughtering witness Executive being Hong pct dlr It pct produce buy well Weinberger three Japan should freeze size mln of and is still OFFER 152 assist grain for 24 third will CANADA and
sample 2: of governments of
Trigram sample sentences should be the following:
Sample 1: Southern California is concerned.
Sample 2: It is the one exercise involving merely at response.

4 Analysis of Out-of-Domain Text

4.1 Unigram Model Perplexity Value

As showned in table 5, table 6 and table 7

4.2 Trigram Model Perplexity Value

As showned in table 5, table 6 and table 7

4.3 Performance and discussion

Table 5 to table 7 gave shown significant perplexity difference between unigram language model and trigram language model across three language corpus. Trigram language model out performs unimodel in all scenarios. Trigram language model, compares to unimodel, takes consideration in grammar structure. Unimodel has all the words as independent events between each other. The trigram language model outperforms the unigram language model might be caused by the modified backoff value. For the trigram language model, the backoff value is set to 0.0008 to adjust to the numbers of UNK ; however the unigram language model backoff vlaue is set to the default provided value.