Summer Semester 2020 SNLP Assignment 3

Name: Awantee Deshpande
Id: 2581348
Email: s8awdesh@stud.uni-saarland.de

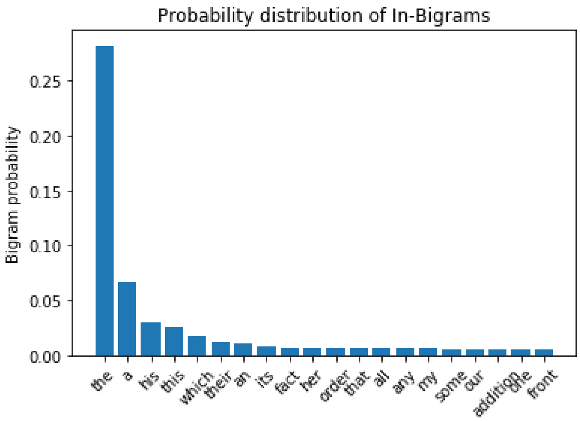
Name: Lakshmi Rajendra Bashyam
Id: 2581455
Email: s8laraje@stud.uni-saarland.de

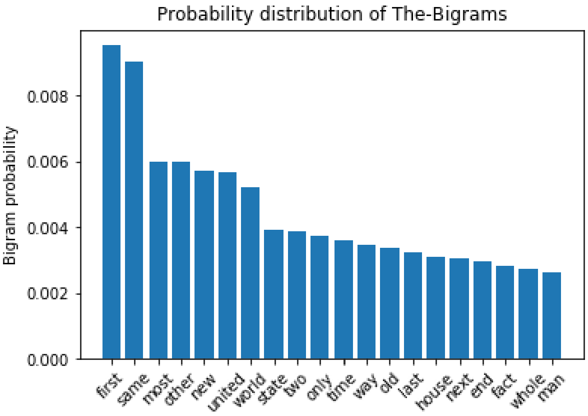
1) N-gram probabilities

The relative frequency of a bigram (w1, w2) is calculated as

$$\boxed{P(w1, w2) = \frac{\text{count}(w1,w2)}{\text{count}(w1)}}$$

Using this, we find the relative frequencies of all bigrams (‘in’, x) and (‘the’, x). The probabilities of the top 20 most frequent bigram tokens we find using this are shown in the plots below:





The expected value for the negative log probabilities is given by E[-logP(X)]. For the ‘in’ and ‘the’ bigram distributions, these are found by summing over the negative logarithms of all corresponding bigram probabilities.

E[-logP(‘in’,x)] = 13.628551495376783
E[-logP(‘the’,x)] = 15.01264590971976

The value is greater for the-bigrams than for in-bigrams. This is because the in-bigrams are very few as compared to the the-bigrams, and their probability distribution is less spread out with respect to the the-bigrams, as can be seen in the plots above as well. The total number of in-bigrams is 21337 of which 3524 are unique, and that of the-bigrams is 69566 of which 13944 are unique.

2) Perplexity

Perplexity is a useful evaluation metric when the test data is close to the train data.
The given datasets are initially preprocessed by removing punctuation, changing to lowercase and splitting by space. Corresponding unigrams and bigrams are found in each text. The formula for perplexity is given by

$$\boxed{\begin{aligned} PP &= P(w_1 \ldots w_N)^{-1/N} \\ &= \exp\left(-\sum_{w,h} f(w,h) \log\left(P(w \mid h)\right)\right) \end{aligned}}$$

The left term f(w.h) is the relative frequency of the test corpus ngrams, and the logarithm term P(w|h) is the conditional probability of the ngrams in the train corpus. Each of them are found using the following formula

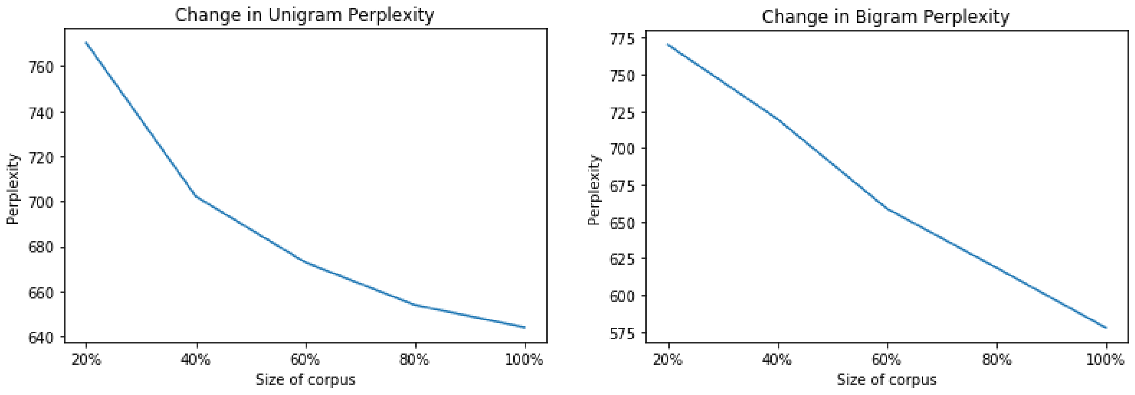
$$\boxed{\text{f(w,h) = f(ngram)} = \frac{\text{count of ngram in corpus}}{\text{count of all ngrams in corpus}}} \qquad \boxed{P(w|h) = \frac{N(w,h) + \alpha}{N(h) + \alpha V}}$$

The conditional ngram probabilities are obtained by getting the ngram counts from the corpus and normalising them so they lie between 0 and 1. Here, N(h) for unigrams will be the total number of words in the corpus, and for bigrams it will be the count of the first token in the bigram. V is the vocabulary size i.e. total number of unique words in the corpus. We get the following values:

Unigram perplexity = 643.957565690078
Bigram perplexity = 577.7969854928702

In practically all language models, no training corpus can capture the entire vocabulary in it. There are always words in the test set that are not present in the training set. This is especially common in higher order n-grams. This is termed as a sparse-dataset problem. In such cases, calculating the conditional probabilities and relative frequencies would result in a value of 0 on unknown tokens and thus would not predict correctly on the test set. For this purpose, we use smoothing. The idea behind smoothing is that we offset the probabilities of words existing in the training set by distributing them amongst unknown terms. This enables the language model to give probability estimates on words from the test set whose count is 0 in the training corpus.

After changing the corpus sizes to 20%, 40%, 60%, 80% and 100%, we obtain the following plot of Perplexity vs. Corpus Size:

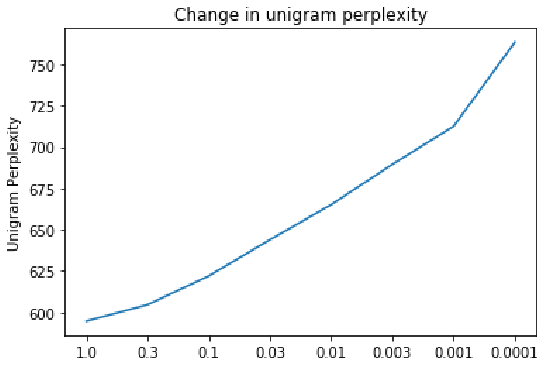


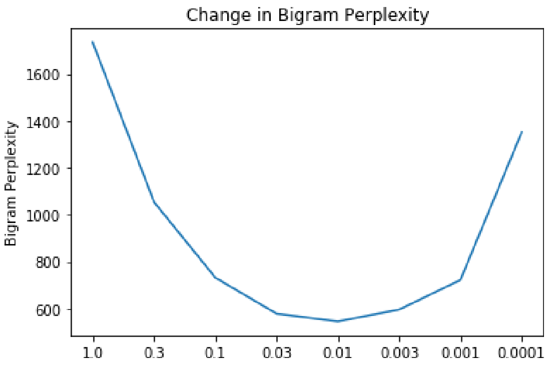
It can be seen that the perplexity decreases with increasing corpus size as expected. It makes intuitive sense that more information would be available from a larger corpus w.r.t. the ngram probabilities and vocabulary size. Thus, the perplexity goes down. Also, the values of bigram perplexities are still higher than the values of unigram perplexities. This is because bigrams are much more informative about the word order than unigrams, and they provide more context of the ngram token.

3) Bonus

3.2 Perplexity vs Alpha

Alpha is the smoothing factor we use to offset the probability values in the conditional distribution. The following plots are obtained for unigrams and bigrams over a changing range of alpha from [1.0, 0.3, 0.1, 0.03, 0.01, 0.003, 0.001].





The bigram-plot shows the more general case. For high values of alpha, we risk reducing the existing probability distribution values drastically, while for really low values of alpha, the assigned probabilities for the new words become negligible and tend to zero, thus resulting in higher perplexity values. Thus, such plots help to determine the optimum value of alpha that can be set. For this case, we can see that alpha can range between 0.01 to 0.03.

3.2 Expected Values

(a) We know that

$$P(X,Y) = P(Y \mid X).\ P(X)$$

Taking negation and log on both sides,

$$-\log(P(X,Y)) = -\log(P(Y \mid X).\ P(X))$$
$$\therefore -\log(P(X,Y)) = (-\log P(Y \mid X)) + (-\log P(X))$$

Taking Expectation on both sides

$$E[-\log(P(X,Y))] = E[\ (-\log P(Y \mid X)) + (-\log P(X))]$$

$\because$ E[A + B] = E[A] + E[B], we can rewrite the above equation as

$$E[-\log(P(X,Y))] = E[\ -\log P(Y \mid X)] + E[-\log P(X)]$$

Hence proved.

(b) Using Bayes Theorem,

$$P(Y|X) = (P(X|Y).\ P(Y))\ /\ P(X)$$
$$\therefore P(X).\ P(Y|X) = P(X|Y).\ P(Y)$$

Taking negation and log on both sides and splitting the log terms,

$$-\log P(X) + -\log P(Y|X) = -\log P(Y) + -\log P(X|Y)$$

Taking expectation on both sides and using E[A + B] = E[A] + E[B], we get

$$E[-\log P(X)] + E[-\log P(Y|X)] = E[-\log P(Y)] + E[-\log P(X|Y)\ ]$$

Hence proved.