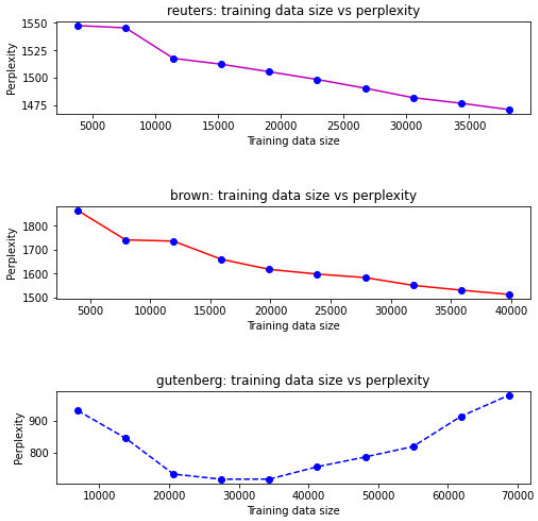
Assignment 3: Language Modeling

1 Unigram Language Model Analysis

The data sets of the three domains are divided as shown in the following table:

Corpus	Train	Dev	Test
brown	39802	8437	8533
reuters	38169	8082	8214
gutenberg	68740	14729	14826

1.1 Analysis on In-Domain Text



In this part, we divide the amount into 10 different size of training data chunks in each of the three domains. Then we train a unigram language model for each of the domains, calculate the perplexity on different training data size and plot the result.

From the above graph, we can see how perplexity changes as mount of training data varied in each of the three domains. On Reuters corpus and Brown corpus, we can see that when the training data size becomes larger and larger, the perplexity becomes smaller and smaller. I think this is because when the training data size increase, there can be more sentences fitted in the model, and the model's performance increases. However, we can see that on Gutenberg Corpus, as the training data size increase, the perplexity first goes down, but when the size is larger than 40000, the perplexity goes up. I think this is because of the unusual distribution of the data of Gutenberg Corpus. The words closer to the front fits the model of the whole dataset better. While words closer to the back is not very common.

1.2 Analysis on Out-of-Domain Text

X Test	brown	reuters	gutenberg
brown	1604.2	6736.6	1762.01
reuters	3865.16	1500.69	4887.47
gutenberg	2626.05	12392.5	1005.79

In this part, we learn the unigram models for each of the domains, and evaluate it on all three domains, which can get the perlexities for three 3 $\times$ 3 matrix. Then we insert the results in the tables above.

Because I think the perplexities on the test set is more important, here I mainly discuss the trends I find according to the result in the last matrix:

1. Firstly, we can find that the unigram models we trained can get best results on the test size when the model's training data and the testing data are from the same domain.
2. Secondly, we can find that the unigram model that is trained on the data from Brown Corpus can also get good results on the test data from Gutenberg Corpus; Also the unigram model that is trained on the data from Gutenberg Corpus can also get good results on the test data from Brown Corpus.
3. Thirdly, we can find that both the models that are trained on the data from Brown or Gutenberg can get much worse results on the test data from Reuters Corpus.

From the first trend, I think the reason for this is that the model can be more fit on the test data after learning on the data from the same corpus, thus can get better results. For the second and third trends, I think the reason is that the data from Brown and Reuters is more similar to each other(they are both modern English). However, the Gutenberg corpus is so different than the other two that it contains the text from English words or grammar from centuries ago.

2 Context-aware Language Model: Implementation

In this part, I implemented a Trigram Language Model using the starter code. For any sentence $x_1 \dots x_n$ where $x_i \in V$ for $i = 1 \dots (n-1)$, and $x_n = STOP$, the probability of the sentence under the trigram language model is:

$$p(x_1, \dots, x_n) = \prod_{i=1}^{n} q(x_i | x_{i-2}, x_{i-1})$$

In our model, let $|V|$ Be the vocabulary size, we can use maximum likelihood estimate method to calculate $q(x_i | x_{i-2}, x_{i-1})$:

$$q(x_i | x_{i-2}, x_{i-1}) = \frac{count(w_{i-2}, w_{i-1}, w_i) + \delta}{count(w_{i-2}, w_{i-1}) + |V|\delta}$$

Here we use add-δ smoothing method to flatten spiky distributions so that they can generalize better. When $\delta = 1$, it is Laplace smoothing method.

3 Context-aware LM: Analysis of In-Domain Text

3.1 Perplexity of the test set

Performace of Unigram	brown	reuters	gutenberg
brown	1604.2	6736.6	1762.01
reuters	3865.16	1500.69	4887.47
gutenberg	2626.05	12392.5	1005.79

Performace of Trigram	brown	reuters	gutenberg
brown	10108.6	25658.7	17913.9
reuters	21193.6	1499.88	30317.3
gutenberg	17072.1	36810.1	3232.75

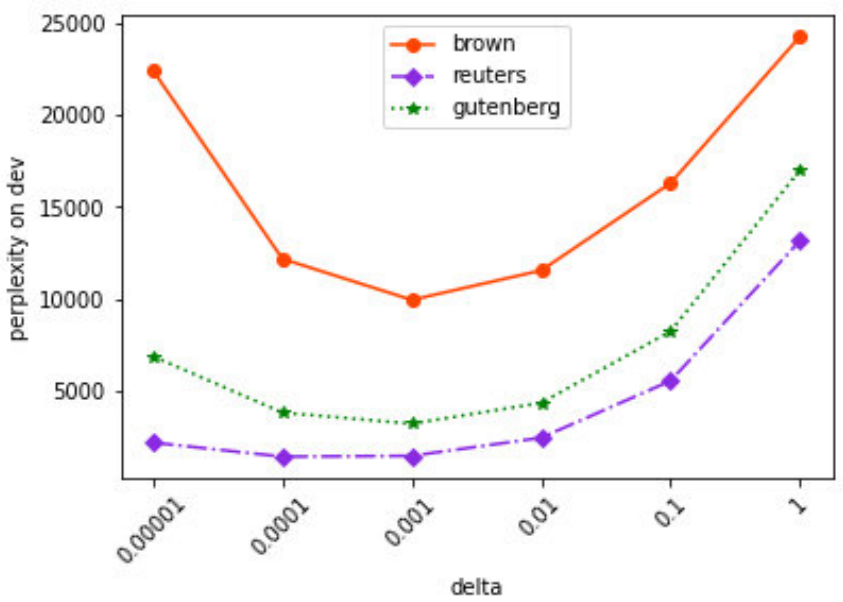
I compute the perplexity of the test set for each of the three domains and insert the result in the above table. Compared to the results of the unigram model which is shown in the **1.2** section. We find that the results is worse than the results got from the unigram model.

	brown	reuters	gutenberg
Perplexity on Train data	54.344	34.4187	49.8836

I check the models perplexities on each domains' training data, the perplexity is really low. Because from the **2.1** section, we find the best $delta$ is 0.001, I think the reason for the result of trigram model is worse than unigram model is that our model is overfitting. I think this can be solved if we can have larger data sets or use some normalized method.

3.2 Performance for Different Hyperparameters

The hyper-parameter in our model is δ, which is greater than 0 and less than 1. To search for the best δ, we perform experiments on the dev data set for each of the three domain using different δ values, and the results are show in the following graph:



We can find that we $\delta = 0.001$, we can get the best result. Therefore, in the following performance discussion, we set $\delta = 0.001$.

3.3 Eample of Sampled Sentences

We use the prefix 'they have' and generate the following sampled sentence on each of the three domains. Due to space constraints in the report, I only show the first 20 words of the example sentence here：

- Brown: *they have oblivious alternately Republican inside little hooliganism criss Share Reifenrath contemplating attrition Significant clandestine Approaching ornaments stove Hurrah decidedly Christiana*
- Reuters: *they have automatically SNOWS CUBA GGLF advantage terribly NAT Yohai incentive Affiliates SHR tungsten Cerier APPROVAL skies flavor left BIFFEX formulas*
- Gutenberg: *they have flashy Greater witcheries 153 leg notched doubtingly owe cannot tide Astronomy restrained Toyles Dondra burly meteyard LAST myself revilers*

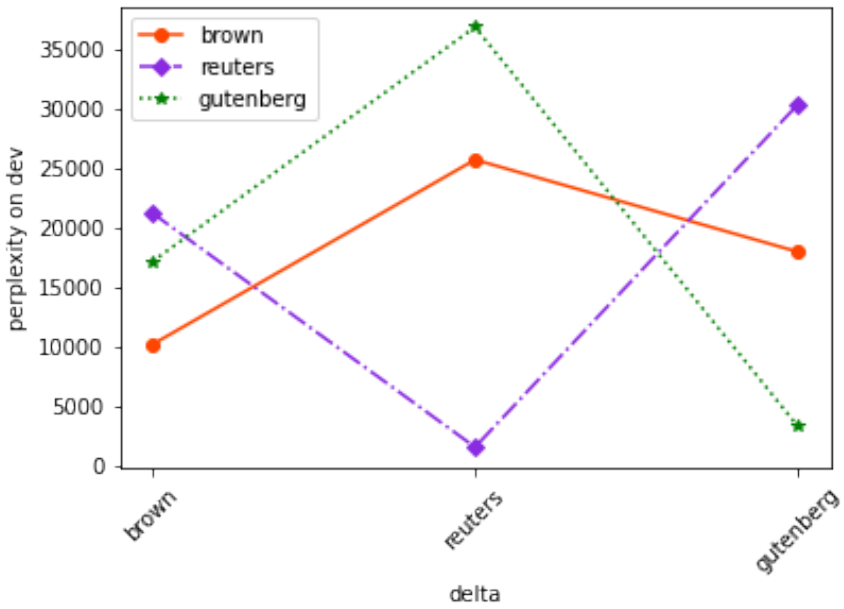
4 Context-aware LM: Analysis of Out-of-Domain Text

4.1 Empirical Analysis

Performace of Trigram	brown	reuters	gutenberg
brown	10108.6	25658.7	17913.9
reuters	21193.6	1499.88	30317.3
gutenberg	17072.1	36810.1	3232.75

We got the above matrix for the perplexity of all three of your models on all three domains. Compared to the unigram model, we can still find that models we trained can get best results on the test size when the model's training data and the testing data are from the same domain. The results on the reuters training model on the reuters set is better than the unigram model, but the results on other domains are worse than the unigram model. I think the reason for this is that our model is overfitting and this can be solved if we can have larger data sets or use some normalized method.

4.2 Qualitative Analysis



After plotting the experimental results into a graph I find that the trigram model trained on Brown and Gutenberg have the similar trends on the three test set from three different corpus. I think the reason for that is is that the data from Brown and Reuters is more similar to each other(they are both modern English). However, the Gutenberg corpus is so different than the other two that it contains the text from English words or grammar from centuries ago.