CS565 : Assignment - 2

Devaishi Tiwari (170101021)

Colab Notebook Link:
https://colab.research.google.com/drive/1IzfvO0xR4WhfzUPd33esarQfNs5vbg7i?usp=sharing

---

1. Dataset Segmentation

I have performed sentence segmentation on the provided dataset using **NLTK Sentence Tokenizer.** This set of sentences is our basic dataset.

This dataset is then randomly shuffled and segmented into **training data** (90% of the dataset) and **test data** (10% of the dataset). The training data is then randomly shuffled and divided **5 times** to create different partitions of **training data** (90% of the training dataset) and **validation data** (10% of the training dataset).

2. Trigram Language Model

In this section, I have implemented a trigram language model and trained it on the training dataset. This model was then run on both the validation data and the test data. Following smoothing techniques were used in the model individually.

2.1 Interpolation Smoothing

In this smoothing technique, three parameters corresponding to the onegram, bigram and trigram probability of occurrence of a word. We iterate these parameters from **0.1 to 1.0 in increments of 0.1**, **keeping their sum equal to 1.0**. We calculate likelihood in each of these cases and choose the parameter tuple that yields maximum log likelihood.

In our implementation, the ideal parameter values were,

Lamda_1 = 0.8, Lambda_2 = 0.1, Lambda_3 = 0.1

NOTE: The language model used in case of validation sets is trained on the corresponding training sets, while the language model in case of test dataset is trained on the first training set.

Test Data Set	Log Likelihood	Perplexity
Validation Set 1	-0.063	1.044
Validation Set 2	-0.035	1.025
Validation Set 3	-0.101	1.072
Validation Set 4	-0.025	1.017
Validation Set 5	-0.034	1.023
Test Dataset	**-0.041**	**1.029**

2.2 Discounting Smoothing

In this smoothing technique, a chunk of probability is taken from words present in the training data and distributed evenly amongst the words that aren't present in the training data. We use a parameter to **define what size of chuck is to be taken out**. This value **strictly lies inside 0 and 1** and can be set to provide maximum log likelihood. We iterate this parameter from **0.1 to 1.0 in increments of 0.1**. We calculate likelihood in each of these cases and choose the parameter that yields maximum log likelihood.

In our implementation, the ideal parameter value was,

Beta = 0.4

NOTE: The language model used in case of validation sets is trained on the corresponding training sets, while the language model in case of test dataset is trained on the first training set.

Test Data Set	Log Likelihood	Perplexity
Validation Set 1	12.853	-3.684
Validation Set 2	11.680	-3.546
Validation Set 3	10.622	-3.409
Validation Set 4	12.142	-3.602
Validation Set 5	14.733	-3.881
Test Dataset	**11.852**	**-3.567**

2.3 Laplace Smoothing

In this smoothing technique, our main aim is to somehow provide **non-zero probability to every word** present in the dataset. This is achieved by **adding some constant, say k, to the count** of every distinct word in the test dataset. To compensate for the extra count, we will also **add k times of the vocabulary size in the denominato**r.

In our implementation, we have taken our constant **K = 1.**

NOTE: The language model used in case of validation sets is trained on the corresponding training sets, while the language model in case of test dataset is trained on the first training set.

Test Data Set	Log Likelihood	Perplexity
Validation Set 1	177.01	-7.467
Validation Set 2	109.97	-6.781
Validation Set 3	150.75	-7.236
Validation Set 4	192.40	-7.588
Validation Set 5	121.51	-6.925
Test Dataset	**179.26**	**-7.485**

2.4 Conclusion

From the data mentioned above we can easily see that, in our implementation the perplexity is least for interpolation smoothing technique, followed by discounting technique and finally laplace technique.

Interpolation > Discounting > Laplace

While Discounting and Interpolation techniques both work well after being trained, the laplace approach does not give desired results. Another important conclusion is that the parameter values for Interpolation and discounting methods remain the same in all the cases.