BBM497: Introduction to Natural Language Processing Lab. **(Due: 30/03/2020)**

Submission Assignment #1

Instructor: Necva BOLUCU *Name:* Mehmet Emin TUNCER, *Netid:* 21591022

1 Data Structures and Language Models

First, I created the **dataset()** function to read the sentences in the dataset. I received the sentences by reading the file line by line, then I split the sentences according to the space character, and deleted the punctuation marks. I replaced the word XXXXX in the sentences with the correct one with the **findXXX()** function. After reading the sentences, I have created three language models such as UnigramModel, BigramModel and TrigramModel with using the class structure. For each token type, I created dictionaries that hold the token counts using the list of sentences. I used the tuple structure when creating dictionaries. In this way, I was able to access double or triple word groups more easily. And I assigned these dictionaries as the variable of the models.I used **Ngram()** function for creating those models. It takes one integer argument from and creates the models with respect to the argument. I created sentences with **generate()** function for every model.Within the generate () function, I created separate generate functions for each model. These functions were generateUnigramSentences (), generateBigramSentences () and generateTrigramSentences (). When creating sentences in Bigram and Trigram models, I used the next () function. I also divided the Next () function into two functions for bigram and trigram models.

2 Sentence Generation Results

With using the generate functions, I created some sentences with for all models. I tried to creating sentences with different length and sentence counts.

```
modelName="Unigram"
generate(10,3)
modelName="Bigram"
generate(12,3)
modelName="Trigram"
generate(15,2)
```

```
Generating sentences with UniGram Model

1. Sentence:   <s> of she that knees her people girl tales fairy </s>
2. Sentence:   <s> the fears that can sometimes of of when pond </s>
3. Sentence:   <s> broader of me be and though men had clergyman to </s>

Generating sentences with BiGram Model

1.Sentence:  <s> at first they call that flits across the ivory perspective glass </s>
2.Sentence:  <s> it is a party with bitterer than this story </s>
3.Sentence:  <s> and i did nt think we could hardly walk in the </s>

Generating sentences with Trigram Model

1.Sentence:  <s> then billy mink back towards the city and i can read nonsense but alice hardly </s>
2.Sentence:  <s> in the sky </s>
```

Some examples for generated sentences

3 Calculation of Perplexity

For the perplexity value, I first found the probabilities of each sentence produced for all three models with the **prob()** function. I divided the **prob()** function for each model. If the probability value is 0 in a model, then I used the **sprob()** function for the new probability. Again, I split the **sprob()** function for models. I calculated the probabilities with logarithm. Then I used the probability values I obtained in the **ppl()**function.

```
<s> empty start it with built boots would to brothers stand </s>

Probability for Unigram is ==>>  -116.81624262702488
Probability of for Bigram is ==>>    -107.66495702566776
Probability of for Trigram is ==>>    -136.8176549511993
Perplex for Unigram is ==>>  3407.9462956190177
Perplex for Bigram is ==>>  2008.7464405428982
Perplex for Trigram is ==>>  10820.437754388762

<s> when they danced on the baskets concealed her cousins </s>

Probability for Unigram is ==>>  -104.1213376401212
Probability of for Bigram is ==>>    -45.444202039273534
Probability of for Trigram is ==>>    -105.92676523416944
Perplex for Unigram is ==>>  2828.0191010238154
Perplex for Bigram is ==>>  70.09750036900738
Perplex for Trigram is ==>>  3168.7671399545725


<s> as soon as she always did me a palace on </s>

Probability for Unigram is ==>>  -92.46358007343248
Probability of for Bigram is ==>>    -47.880763598081096
Probability of for Trigram is ==>>    -88.38525586515465
Perplex for Unigram is ==>>  834.8066828359264
Perplex for Bigram is ==>>  63.56072312879538
Perplex for Trigram is ==>>  659.5956109930362

<s> one very excellent </s>

Probability for Unigram is ==>>  -40.226325757694106
Probability of for Bigram is ==>>    -23.11349900125912
Probability of for Trigram is ==>>    -12.69047408026725
Perplex for Unigram is ==>>  1056.637756200395
Perplex for Bigram is ==>>  98.54425082384779
Perplex for Trigram is ==>>  23.232859326190763
```

Some examples for probability and perplexity values of generated sentences

4 Error Analysis for Sentences

Considering the generated sentences, probability and perplexity values, the Trigram model is generally more reliable than other models.However, this does not apply in all circumstances. Some sentences may have lower perplexity value for Bigram than Trigram model. However, looking at the general results and other sentences I have generated, the most reliable model is the Trigram model.