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

Submission Assignment #1

Instructor: Asst. Prof. Dr. Burcu CAN, Necva BÖLÜCÜ *Name:* Deniz Zağlı*, Netid:* 21527668

- Data Structures that you are used to build your language model:
  - I used a **list** to keep the dataset read in the desired file reading section. Readlines, one of Python's own functions, is already throwing lines in the text file into a **list**. I did not need an extra data structure. I read the lines in the dataset one by one and deleted the new line mark at the end of the line. I kept these new lines in a list by deleting the line numbers at the beginning of the line.
  - I used python's own **list** functions when removing punctuation marks, removing other marks, and converting all uppercase to lowercase. This made my job much easier. This was one of the most important situations in using **lists** as a data structure to hold sentences.
  - I used dictionary while creating Unigram, Bigram and Trigram models. In this case, the biggest aim in using **dictionary** is data access time. In this case, let's assume that we use a **list**. In order to access frequencies every time, we would have to know the index and keep this index in a variable. I used this **dictionary** structure because of these convenience.
  - I also used a **list** for the preprocessing of the sentences I wanted to measure. I kept the sentence in a single-element **list**. This is due to the ease of the proccess I can apply on the list in the following articles.
  - One of the great advantages of using **lists** in the next function is random operations. It allows us to do a random python library and random operations for the elements in the **list**. I used the list to provide this convenience, since we actually implemented a weighted random operation during the next operations.
- Error Analysis of your generated sentences:
  - Below are the first sentences produced by our dataset. I produced sentences using unigram, bigram, trigram respectively. There are 3 sentences with maximum 10 words for each model. The reason for the termination of the sentences early is because they correspond to the end of the sentence.
  - **Unigram:**
  - "Replies and old could and one wildest which"
  - "Answered him she the"
  - "On the said had it"
  - **Bigram:**
  - "The barn boy they changed sides grandfather man should be"
  - "Whereupon boats with all them verbatim with me go and"
  - "And to the chimneypots down my leave to the sultan"
  - **Trigram:**
  - "It looks very foolish folk who asked what she did"
  - "He stood up human beings as if she had to"
  - "This had been there and it was that she was"
  - When we examine the sentences created with unigram, bigram and trigram, it is obvious that which is more meaningful. When creating a sentence with Unigram, we actually make predictions without thinking about completely random history. The only thing we pay attention to is the frequency of the words. Without thinking about the past, we see that random estimates made only by frequency cannot make logical sentences. The reason why sentences are short in Unigram is that our prediction is the sign of the end of the sentence. Since the sentence ending sign is also numerous, it appears early. When we look at the sentences we create with bigram, we see that it is more meaningful than

the unigram. The reason is that each word is related to the previous word coming from it. This link provides better integrity of the sentence than the unigram. When we look at the sentences formed by the trigram, we see that the sentences now have more meaning. because in this case, all the words except for the first two words are related to the two preceding words. It is noticeable how much it has to do with the two words preceding it has an effect on the integrity of the sentence. It seems that if the number of N increases according to our tests, the meaning integrity increases. But it's actually not like that. For this, we need to increase N and do more tests and observe this. One of the most visible conditions due to the N increase is the processing time. This increase can be felt seriously, especially if you are working on a large dataset, just like the dataset of this assignment.

- Calculation of perplexity:
  - Perplexity is the inverse probability of the set, normalized by the number of words. Because of the difficulty of multiplication in the normal formula of Perplexity and the number is very small, we do these operations on the log base. For this reason, the multiplication process turns into addition.
  - **First sentence (Generated with unigram)** ”Replies and old could and one wildest which”:
  - Unigram perplexity: 1.0000010448930332
  - Bigram perplexity: 1.0000023651761287
  - Trigram perplexity: 1.0000026546702219
  - **Second sentence (Generated with bigram)** ”The barn boy they changed sides grandfather man should be”:
  - Unigram perplexity: 1.0000013120626798
  - Bigram perplexity: 1.0000023870376795
  - Trigram perplexity: 1.000003112429651
  - **Third Sentence (Generated with trigram)** ”It looks very foolish folk who asked what she did”:
  - Unigram perplexity: 1.0000012852078115
  - Bigram perplexity: 1.0000021558589856
  - Trigram perplexity: 1.000002582247432
  - When we examined the results, his perplexity was low because we could not create a 10-word sentence with the unigram. But when we examine the sentences we have created with bigram and unigram, we see that perlexity is significantly reduced. I used smoothing in the Perplexity calculation. Since the new word groups created cannot be found in the model, the perplexity of trigram and bigram is higher than the unigram. But our observations and the tests we have done are that the meaning integrity of bigram and trigram is higher. So I attribute the problem here to the lack of these word groups in the model.
- Sentence of generation:
  - Below, the probabilities of the two sentences we created with bigram and trigram are calculated and examined.In the second part, I interpreted the results and here I calculated the probability values. If an unprecedented value appears in the probability of MLE, the result is zero. This came across in the trigram.
  - **Sentence (Generated with bigram)** ”The barn boy they changed sides grandfather man should be” MLE probabilty:
  - Unigram: 2.6704318685987897e-34
  - Bigram: 1.962457996576393e-59
  - Trigram: 0.0
  - **Sentence (Generated with trigram)** ”It looks very foolish folk who asked what she did” Smoothed probabilty:
  - Unigram: 2.645540129487184e-34
  - Bigram: 3.480372871516l2e-59
  - Trigram: 1.6032214826235658e-73