National University of Computer and Emerging Sciences

Natural Language Processing (NLP)

"Assignment 2"

STUDENTS NAME	ROLL NUMBER
Nasir Iqbal	17I-0519

DEGREE PROGRAM: BS(CS)

Section: B

SUBMITTED TO: Sir Muhammad Bin Arif

Contents

Introduction

In this report we are going to compare ***n*-grams** of two corpus. An ***n*-gram** is a contiguous sequence of *n* items from a given sample of text or speech. The items can be phonemes, syllables, letters, words or base pairs according to the application. The *n*-grams typically are collected from a text or speech corpus. In our implementation I have used two corpus of novel which will be discussed in below sections. The code Implementations are explained in the below section.

Corpus Selections

Data play very important role in Machine learning and NLP tasks. In order to implement ***n*-gram** model we need some input text data. While searching for data, I have found two corpora where corpus 1 is extracted from Alchemist Novel of paulo Coelho and corpus 2 is extracted from Eleven Minutes Novel. Corpus1 consist of 2614 sentences and 41407 tokens. On the other hand, Corpus2 consists of 3097 sentences and 74188 tokens. We will be using these two corpora for language modeling.

Data Preprocessing

First of all I have loaded the two corpus in two NumPy arrays. Once the data is loaded, I have passed it to the the preprocess data function which clean the data from special characters and return the clean version of both the data. And then, I have tokenize and segment both the text data.

Data Preprocessing

Once we preprocess the data, we get some statistical parameters about the data such as the number of unique tokens which can be used while defining the vocabulary size in a model. Next we create the following language models on the training corpus -

1. Unigram
2. Bigram
3. Trigram

We have find the top 10 bigrams, trigrams, of both the corpuses without smoothing. And then We remove those which contain only articles, prepositions, determiners, for example, '*of the*', '*in a*', etc. These are called stop words and can be removed by using an inbuilt list from NLTK. Finally we compare unigrams, bigrams and trigrams of both the corpuses using seaborn library.

Figure 1,2,3 shows unigram, bigram, trigram comparisons respectively.

Corpus 1 corpus 2

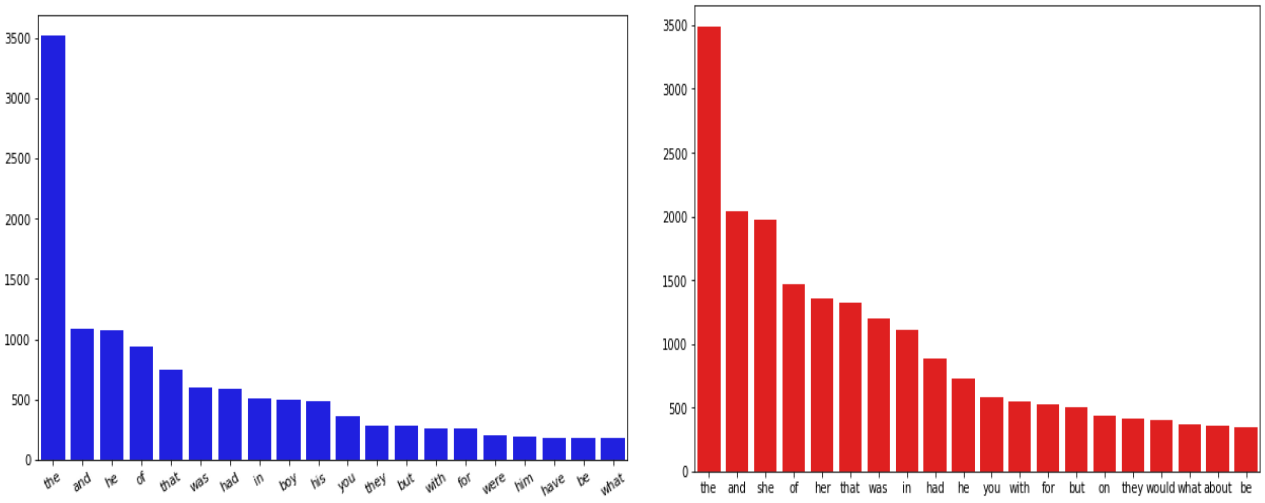
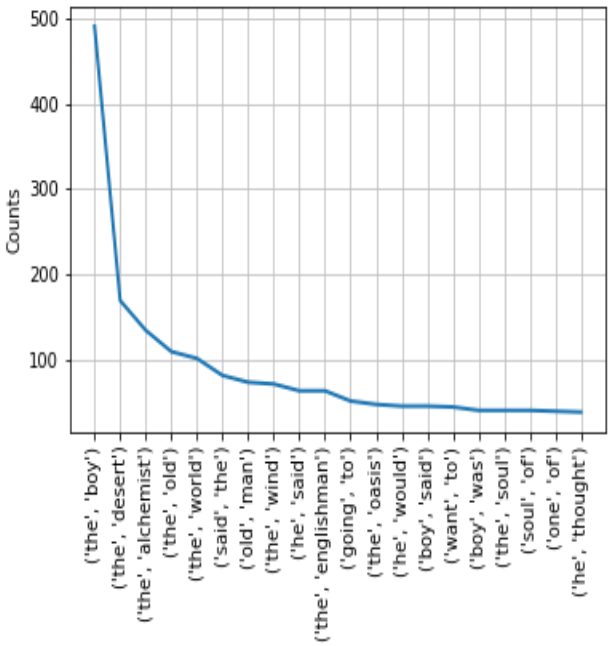


Figure1

Corpus 1



corpus 2

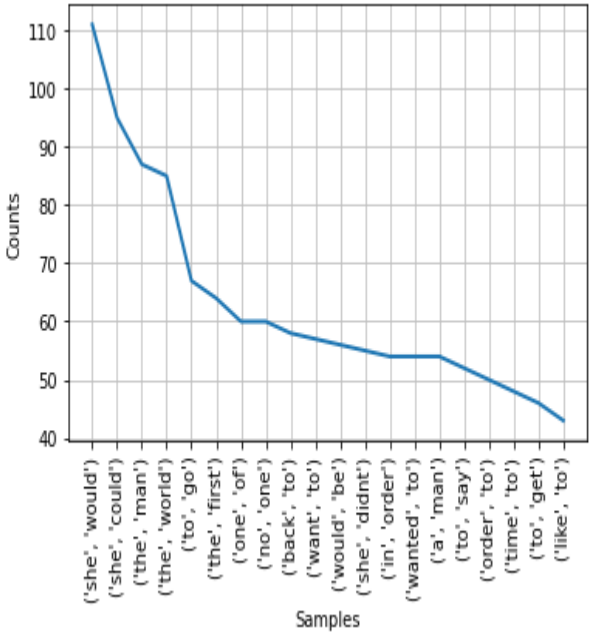
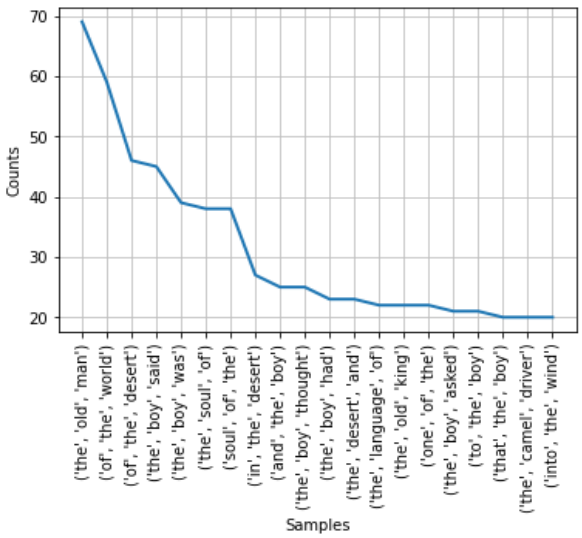


Figure 2

Corpus 1



corpus 2

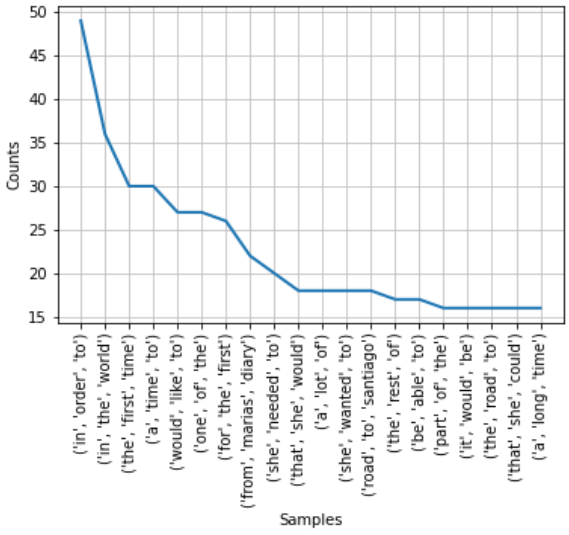


Figure 3

References

1) https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-language-model-nlp-python-code/

2) https://medium.com/swlh/language-modelling-with-nltk-20eac7e70853

3) https://seaborn.pydata.org/generated/seaborn.barplot.html