**Title of the Project:**

NEXT WORD PREDICTION

**Input data:**

Data scraped from web.

**Domain:**

Environmental science.

**Libraries used:**

* nltk
* re
* collections

**Pre-processing tasks:**

1. Remove numbers and percentages and degrees.
2. Remove links/URL
3. Remove special characters and all punctuations except period (.)
4. Remove non-English text (if any)
5. Normalization: case folding
6. Normalization: removal of additional whitespaces.

**Dataset Description:**

URLs scraped: 57

No of words: 100961

No of unique words: 10342

**Block Diagram:**

**DATA COLLECTION**

**DATA PREPROCESSING**

**MODEL CREATION**

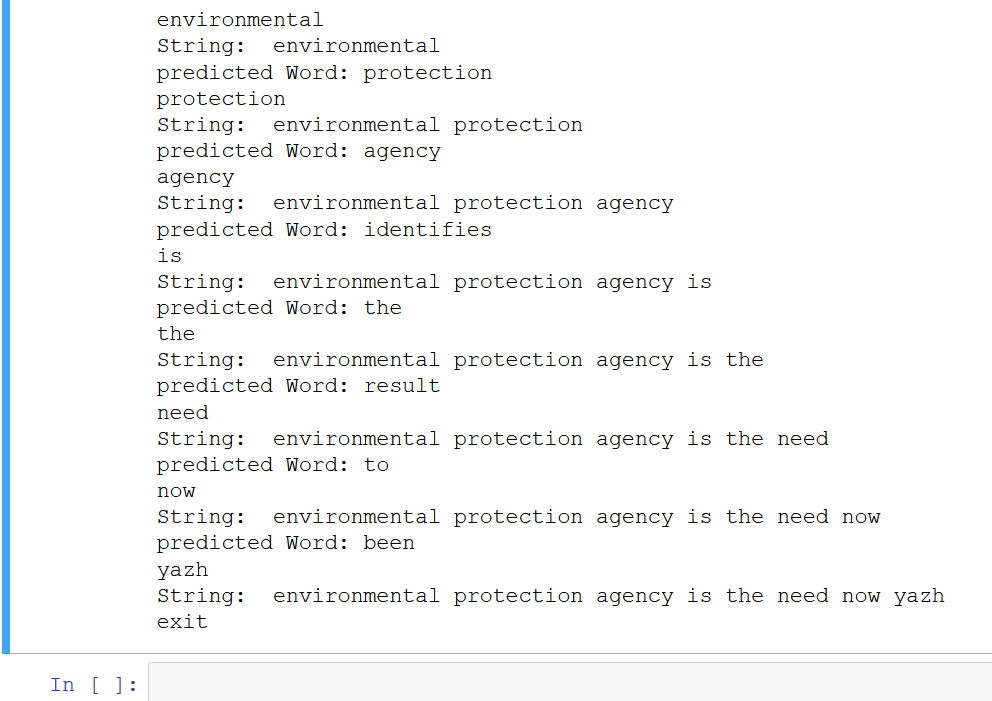
*(bigram, trigram and**quadgram)*

**PREDICTION**

**Description:**

* The first and foremost task we do is data collection. We didn’t take an existing corpus directly. Rather, we scraped data from the URLs related to environmental sciences and created a data set. The second step is data preprocessing. In this step, we do a lot of tasks like removing numbers, percentages, links/URL, special characters and all punctuations except period (.), non-English text (if any) We also perform normalization tasks such as case folding and removal of additional whitespaces.
* Now we have a clean data set. So, we move on to the third step, wherein we create the language model. We created bigram, trigram and quadgram models which calculate the probabilities of the next words possible based on the previous one word, two words and three words respectively. We choose the word with the maximum probability as the next word and display it to the user.
* If the length of the input is >=3, we first try quadgram model. If empty result set is returned by the quadgram model, then go for trigram model. If that result set is also empty, we move on to bigram model.
* And, it should be noted that our model will not predict any word if the typed sequence of words is not found in our data set.

**Actual output (Screen shot):**



**Description of output:**

* If the user types a word, it shows a suggestion as “predicted Word” part.
* The user can select the suggestion or type a different word. The user input is kept on appended in “String” part.
* If no predictions are available, then it won’t show anything. When we give “exit”, the program will exit.