Lambton College

In Toronto

Gurpreet Singh (C0817627)

Ravi Singh Khipal (C0819623)

Prasanth Moothedath Padmakumar (C0796752)

Submitted To:

Parisa Naraei

1. Find the most common algorithms the NLTK library covers and very briefly summarize your findings.

➔ Natural Language Processing can be simply termed as a way to deliver/convert the human based languages in order to help computer machines understand and process accordingly. It basically helps in performing automated task by machines. It enables robots to interpret and comprehend human language in order to do repetitive activities automatically.
Natural language processing (NLP) is a branch of computer science that focuses on making natural human language understandable to computer algorithms. There are multiple algorithms used in processing of NLP like Tokenization Stopwords, Stemming, Lemmatization, Parts of speech tagging, but we are focused on a specific one, Tokenization.
Tokenization: Tokenizing allows you to easily break up text by word or phrase. This will allow you to work with smaller chunks of text that are still somewhat cohesive and comprehensible even when taken out of context. It's the initial step toward transforming unstructured data into structured data that can be analyzed more easily.
When analyzing text, you will tokenize by word and tokenize by sentence. Here are the benefits of both forms of tokenization:

- By Word: Words are the building blocks of natural language. They are the smallest unit of meaning that can nevertheless be understood on their own. Tokenizing your text by word helps you to detect terms that appear often. For example, if you were to examine a set of employment advertisements, you may notice that the term "Python" appears frequently. That might indicate a significant demand for Python skills, but you'd have to dig deeper to find out.
- By Text: Tokenizing by sentence allows you to evaluate how words connect to one another and see additional context.

2. **Choose** 12 functions in NLTK and elaborate on each.

➔ Before using all the below listed NLTK functions function we need to perform tokenizing first.
Count
Count is one of the simplest functions in the list of NLTK functions, it outputs the frequency of a word as single integer value. The function name is count and the parameter passed is the particular word you want to find the frequency of.

Concordance

Concordance plot also helps to understand the frequency of a word, but here we will get an idea of that context the particular word is used. It shows the few contents before and after the occurrence. The number of characters to the left and right will be same. The function name is concordance, and the parameter is the word you want to search for.

Dispersion Plot

Dispersion plot helps in visualizing lexical dispersion, which means the frequency and positions of words present in a text corpus. In Fig 1 we see those words such as game, player, score, oil and Man are plotted in Dispersion plot. The number of vertical blue line against each of the words give an idea about frequency of the and the number along horizontal axis gives the word offset value. The function name is dispersion plot and the parameter is the word you want to search for.

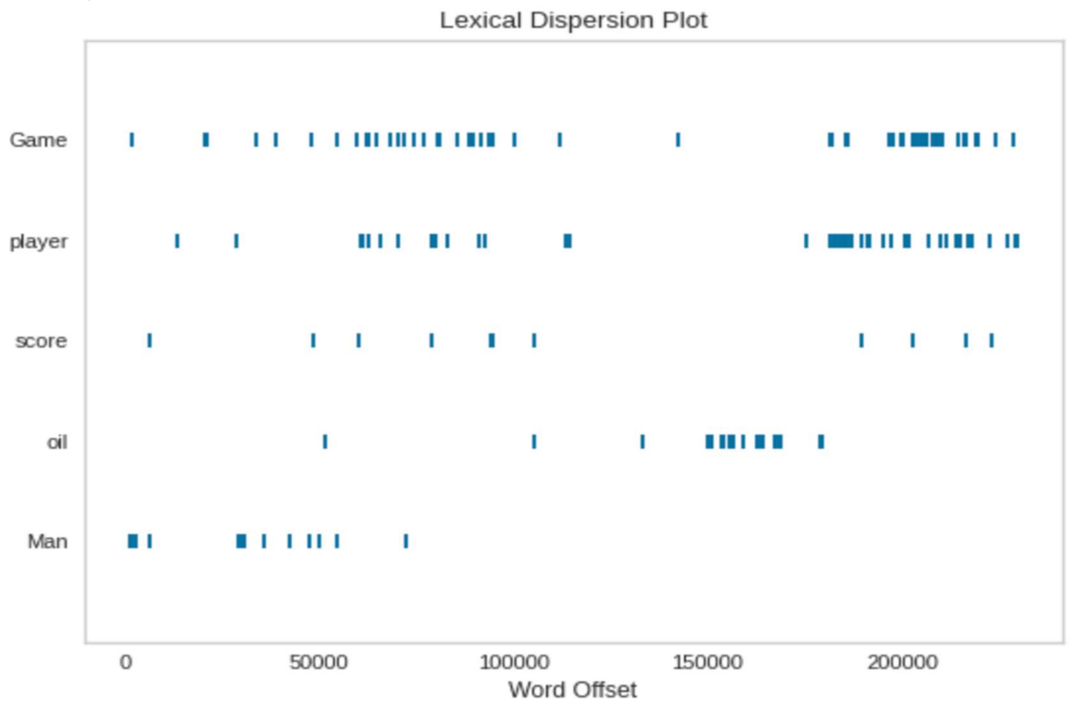
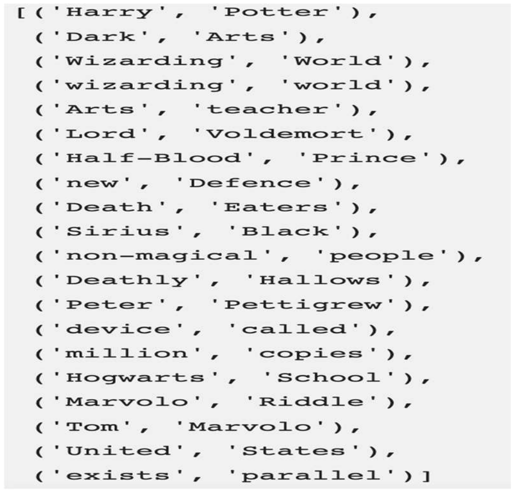


Figure 1. Dispersion plot output

<u>Collocation List</u>

It is one of a potentially useful function among the list which returns a list of set of words that are seen to be occurring together often.



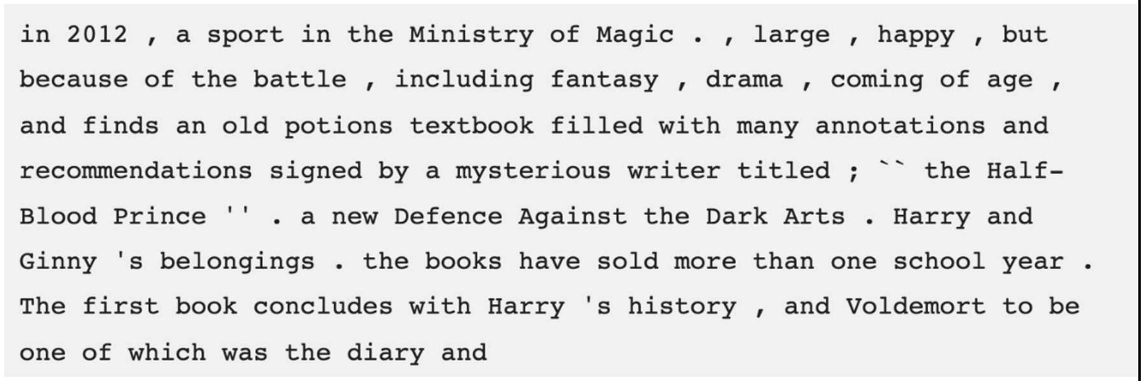
```
[('Harry', 'Potter'),
 ('Dark', 'Arts'),
 ('Wizarding', 'World'),
 ('wizarding', 'world'),
 ('Arts', 'teacher'),
 ('Lord', 'Voldemort'),
 ('Half-Blood', 'Prince'),
 ('new', 'Defence'),
 ('Death', 'Eaters'),
 ('Sirius', 'Black'),
 ('non-magical', 'people'),
 ('Deathly', 'Hallows'),
 ('Peter', 'Pettigrew'),
 ('device', 'called'),
 ('million', 'copies'),
 ('Hogwarts', 'School'),
 ('Marvolo', 'Riddle'),
 ('Tom', 'Marvolo'),
 ('United', 'States'),
 ('exists', 'parallel')]
```

Figure 2. Collocation list output

In figure 2 we can see the sample output of the function. By default, it checks for 2 words which are collated often.

<u>Generate</u>

Generate function can be used to generate random text, each time function is executed it will give a different output. In figure 3 we can see a sample output of the generate function.

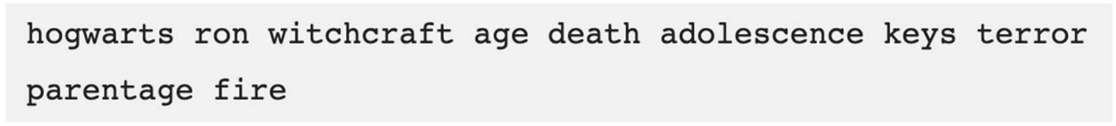


```
in 2012 , a sport in the Ministry of Magic . , large , happy , but
because of the battle , including fantasy , drama , coming of age ,
and finds an old potions textbook filled with many annotations and
recommendations signed by a mysterious writer titled ; `` the Half-
Blood Prince '' . a new Defence Against the Dark Arts . Harry and
Ginny 's belongings . the books have sold more than one school year .
The first book concludes with Harry 's history , and Voldemort to be
one of which was the diary and
```

Figure 3. Generate function’s output

<u>Similar</u>

Similar function helps to find words which shares a range of contexts which are similar to that of the input word. The function name is similar and parameter is that particular word.



```
hogwarts ron witchcraft age death adolescence keys terror
parentage fire
```

Figure 4. Similar function output for the parameter “magic”.

Common context :

By using the function common contexts, we may focus just on the contexts that two or more words share in common.

Index:

The index function returns the first occurrence of a word in the text. Remember that the first token starts at index 0.

Vocab:

In NLTK, the vocab function returns the total number of occurrences a word has been in a given text. For example, we can use the following code to check the occurrence of the word:

```
Tokens = nltk.word_tokenize(text1)
Text1 = nltk.Text(counter1)
Len(text1.vocab())
```

Stemming:

Stemming is a linguistic normalisation technique that reduces words to their base word or removes derivational affixes. For example, the words connection, connected, and linking word are all shortened to "connect."

```
Ps = PorterStemmer()
Stemmed_words = []
For w in filtered_sent:
Stemmed_word.append(ps.stem(word) )
```

Stopwords:

Stopwords are seen as textual noise. Stop words in text include is, am, are, this, a, an, the, and so on.To remove stopwords in NLTK, construct a list of stopwords and filter your token list from these words.

POS Labeling:

The fundamental goal of POS tagging is to determine the grammatical group of a given word. Depending on the context, whether it is a NOUN, PRONOUN, ADJECTIVE, VERB, ADVERBS, etc. POS Tagging searches for connections inside the phrase and tags the word accordingly

References:

- https://machinelearningknowledge.ai/cool-nltk-functions-you-did-not-know-exist/
- https://www.scikit-yb.org/en/latest/api/text/dispersion.html#:~:text=of%20a%20corpus.-,DispersionPlot%20allows%20for%20visualization%20of%20the%20lexical%20dispersion%20of%20words,of%20the%20corpus%20it%20appears.