Basics of Programming – Project Write-Up

Anushka Dhomeja

2021HCS7255

Link to code file on github:

https://github.com/anushka-dhbh/BasicsOfProgramming

The objective of this python project was to:

- Diversify personal knowledge of python programming by using different libraries and packages for python
- Explore functions of the NLTK library in order to study different aspects of language
- Learn the basics of GUI using Tkinter

Packages used in the project:

NLTK

- The natural language toolkit package for python that enables natural language processing in a statistical manner, and analysis of semantics
- Has multiple different in-built functions as well as different corpora containing different types of language information and texts

UDHR

- Is the Universal Declaration of Human Rights in over 300 languages
- One of the corpora available via NLTK, which can be used to create programs to study, for example, the frequency distribution of words in these languages, or even develop a function that can identify the language of a given word.

WordNet

- A lexical database comprising of 155,287 words and 117,659 synonym sets, or synsets. The nouns, verbs, adverbs and adjectives are all categorised into synsets as cognitive synonyms, by creating a network of words that are related to each other semantically
- Available in different languages via Open Multilingual WordNet

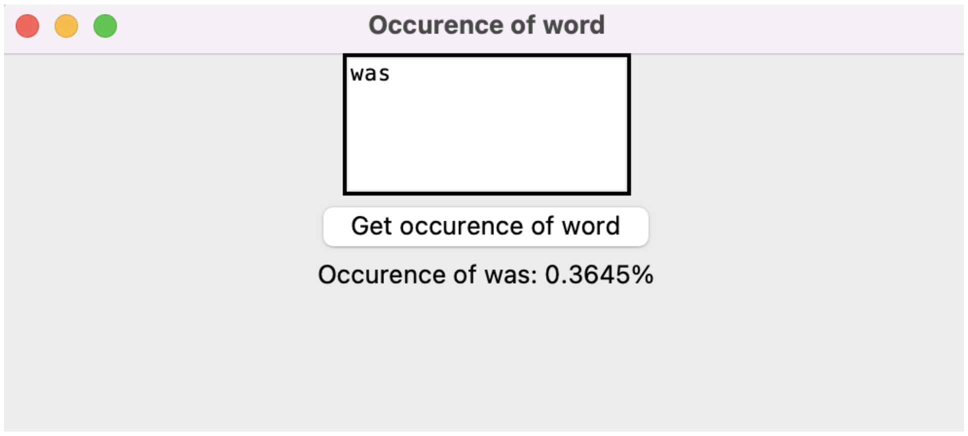
Tkinter

- Enables use of GUI for python

Functions built for the product

1. A function percent (word, text) that calculates how often a given word occurs in a text, and expresses the result as a percentage
    a. Creating a single argument function for Tkinter to analyse the percentage occurence of a given word in text7 of the NLTK book corpus, which is an excerpt of the Wall Street Journal

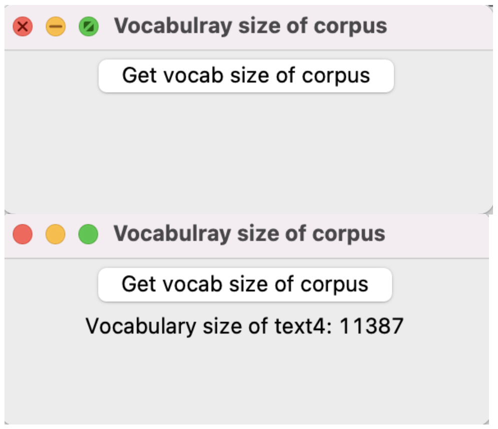
    b. Creating a tkinter frame that asks for a word as an input and returns the percentage occurrence of that word in text7 of the NLTK book corpus, which looks like this when the code is run and the word "was" is given as a string input, and the "Get occurrence of word" button is clicked, giving the output as a percentage value



```
was
```

```
Occurence of was: 0.3645%
```

2. Defining a function called vocab_size(text) that has a single parameter for the text, and which returns the vocabulary size of the text.
    a. Creating a Tkinter GUI frame that has a button which reveals the vocabulary size of the text:



```
Vocabulary size of text4: 11387
```

For further projects the interface could include an input that takes a word and allows the user to analyse and copare the vocabulary size of any text that the user inputs, for example by draggin and dropping, thus allowing the user to analyse the range and diversity of vocabulary used in different types of texts for example a research article vs a rom-com novel.

3. A program that finds all words that occur at least three times in the Brown Corpus.

4. A function find_language() that takes a string as its argument, and returns a list of languages that have that string as a word. Use the udhr corpus and limit your searches to files in the Latin-1 encoding.

5. A function supergloss(s) that takes a synset s as its argument and returns a string consisting of the concatenation of the definition of s, and the definitions of all the hypernyms and hyponyms of s.

6. NLTK has multiple similarity measures, including path similarity which returns a score to denote how similar the senses of two words are. This similarity is measured based on the shortest path between the two words in the taxonomy net that includes hypernyms and hyponyms. This will give a score between 0 and 1. Other such similarity measures are the Leacock-Chodorow, Wu-Palmer, Resnik, Jiang-Conrath and Lin similarities. Each return a score to denote how similar two-word senses but are based on different grounds such as information content along with other elements. This function has been created to analyse the score rankings of a set of word pairs given by Miller and Charles, 1998, in decreasing order. The different types of similarity measures can be used as they are available in NLTK.

7. Write a program to print the 50 most frequent bigrams (pairs of adjacent words) of a text, omitting bigrams that contain stopwords.

Conclusion

These functions are a few illustrations of different programmes that can be used to analyse data from text, enabling the user to draw statistical conclusions and formulate hypotheses pertaining to the contextual use of language. NLTK can be used for natural language processing (or NLP), the applications of which are integral to digital language translation, chatbots, targeted advertising, automated recruitment, voice assistants such as Amazon Alexa, predictive text, and more importantly, artificial intelligence.

By implementing codes from scratch, using guides from the NLTK book chapters 1 and 2, I was able to better understand the working of some of the functions of the NLTK library that are used for text pre-processing in NLP.

In terms of the scope of this project, the real-world applications were considerably limited due to my introductory level abilities with programming, however I hope to implement more extended functions, such as some that would aide in decreasing bias of cv-based recruitment systems by AI. The challenges faced in this project also pertained to my limited exposure to programming.