College of Computer Science and Engineering
Department of Computer Science and Artificial Intelligence

CCAI-413: Natural Language Processing
Lab#2 Morphological analysis

Objectives

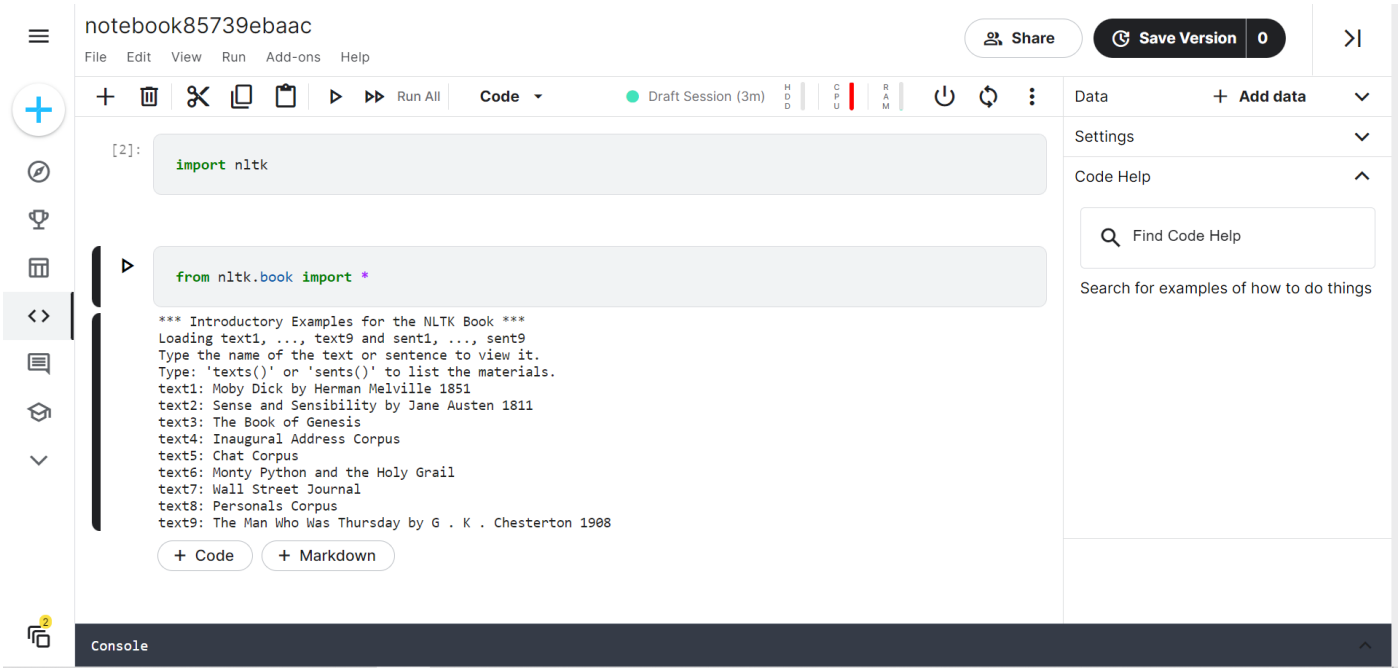
- Apply different text search techniques
- Apply text tokenization
- Apply text stemming
- Apply Arabic text stemming

Lab Tool(s)

https://www.kaggle.com/

Searching Text

Type the following commands to import NLTK and the book data at the Python prompt



The function *concordance ()* shows the occurrence of a given word with some context. The following figure presents how the command is used to search about the word “lived”:

```
[5]: text1.concordance("lived")
```

```
Displaying 13 of 13 matches:
us an old idolator at heart , he yet lived among these Christians , wore their
s dined in the cabin , and nominally lived there ; still , being anything but s
 might more properly be said to have lived out of the cabin than in it . For wh
m , he was still an alien to it . He lived in the world , as the last of the Gr
ld , as the last of the Grisly Bears lived in settled Missouri . And as when Sp
ng himself in the hollow of a tree , lived out the winter there , sucking his o
sun . In the sixth Christian century lived Procopius , a Christian magistrate o
 be as good as the days that Lazarus lived after his resurrection ; a supplemen
e gloomily muttered . " And you have lived in this world hard upon one hundred
ng vessel -- that these men actually lived for several months on the mouldy scr
declare to you , that for the time I lived as in a musky meadow ; I forgot all
iting mockery of grey hairs , have I lived enough joy to wear ye ; and seem and
 and keel did point to . The rigging lived . The mast - heads , like the tops o
```

To find out what other words appear in the same context, we can use the *similar ()* function.

```
text1.similar("lived")
```

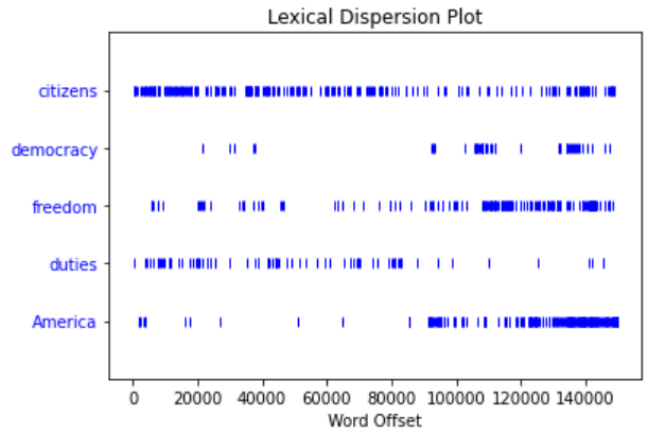
```
sailed was all of is alone s could must been thought ye go aloft who
had came were sleeps can
```

+ Code + Markdown

- *Your Turn!* Try searching for other words

We can also determine the location of a word in the text: how many words from the beginning it appears. This positional information can be displayed using a ***dispersion plot***. Each stripe represents an instance of a word, and each row represents the entire text.

```
text4.dispersion_plot(["citizens", "democracy", "freedom", "duties", "America"])
```



+ Code + Markdown

Text Tokenization

Tokenization is the process of transforming the text into a list of words and punctions, called tokens. NLTK provides a function for tokenization as follows *nltk.word_tokenize()*.

```
text="Tokenization is the process of transforming the text into a list of words, called tokens
tokens= nltk.word_tokenize(text) # Transforming text into words (tokens)
print (tokens)
```

```
['Tokenization', 'is', 'the', 'process', 'of', 'transforming', 'the', 'text', 'into', 'a', 'list', 'of', 'word
s', ',', 'called', 'tokens', '.']
```

+ Code + Markdown

Type the following command to find the length of the tokens:

```
len(tokens) # to find the length of the tokens
```

```
[4]: 17
```

Type the following command to print the first N tokens:

```
tokens[:3] # to print the first three tokens
```

```
[5]: ['Tokenization', 'is', 'the']
```

Text Stemming

Words can be written in different forms (e.g., studying, studies, and study). Stemming is the process of removing prefixes and suffixes of a word. NLTK provides a stemmer called *PorterStemmer()*.

```
porter = nltk.PorterStemmer()
[porter.stem(t) for t in tokens]
```

```
[4]: ['token',
      'is',
      'the',
      'process',
      'of',
      'transform',
      'the',
      'text',
      'into',
      'a',
      'list',
      'of',
      'word',
      ',',
      'call',
      'token',
      '.']
```

+ Code + Markdown

Arabic Text Stemming

Tashaphyne is a python library that provides Arabic stemmer. To use Tashaphyne you have to install the library using the following command:

```
In [1]: !pip install Tashaphyne
```

```
Requirement already satisfied: Tashaphyne in c:\users\bushra\anaconda3\lib\site-packages (0.3.4.1)
Requirement already satisfied: pyarabic in c:\users\bushra\anaconda3\lib\site-packages (from Tashaphyne) (0.6.10)
```

Then we should import the stemmer and create its object:

```
In [2]: from tashaphyne.stemming import ArabicLightStemmer
        ArListem = ArabicLightStemmer()
```

Below is an example of using the Arabic stemmer on the word "أفتضاربانني":

```
In [6]: word = 'أفتضاربانني'
        stem = ArListem.light_stem(word)
        print (ArListem.get_stem())
```

```
ضارب
```

```
# extract root
print (ArListem.get_root())
```

ضرب

```
# get prefix
print (ArListem.get_prefix())
```

أفت

In [10]:
```
# get suffix
print (ArListem.get_suffix())
```

انني

References:

- Bird and Klein, O'Reilly Media, Natural Language Processing with Python, 2nd Edition, 2017.
- https://pypi.org/project/Tashaphyne/