

Example 1 - Tokenisation

Natural language toolkit (NLTK) is the most popular library for natural language processing (NLP) which is written in Python and has a big community behind it.

NLTK also is very easy to learn; it's the easiest natural language processing (NLP) library that you'll use.

In this NLP Tutorial, we will use the Python NLTK library.

1. Create a new project

- Make sure that you link the project to the Python interpreter (different IDEs do this in a different way, but most will ask while you are creating the new project)
- It's a good idea to try running a simple hello world example

```
print("hello world")
```

2. Install NLTK.

- If you are using PyCharm you can click on the interpreter of the project (bottom right), choose interpreter settings, click the + button, and install NLTK.
- If you are using Visual Studio code you can run the following from the terminal.

```
pip install nltk
```

- If it fails it might be because of right restrictions. Try the following:

```
pip install --user nltk
```

After installing nltk, run the following code, and when a window pops up, choose to install all. This will install dictionaries and other information nltk needs to help you process text.

```
import nltk  
nltk.download()
```

3. The subject of a piece of text

Our aim is to analyze a piece of text to see what the text is most probably about.

So first we read the content of a website, and display it.

We also split it up word by word (splits on spaces). We shall call these **tokens**.

```
input = " The term domestic dog refers to any of  
several hundred breeds of dog in the world today. While  
these animals vary drastically in appearance, every dog  
from the Chihuahua to the Great Dane—is a member of the  
same species, Canis familiaris. This separates domestic  
dogs from wild canines, such as coyotes, foxes, and  
wolves."  
  
print (input)  
  
tokens = input.split()  
print (tokens)
```

Now let's calculate the frequency distribution of those tokens using Python NLTK.

There is a function in NLTK called FreqDist() which does the job. But first, import nltk at the very top of the script.

```
import nltk
```

Then:

```
freq = nltk.FreqDist(tokens)
for key, val in freq.items():
    print (str(key) + ':' + str(val))
```

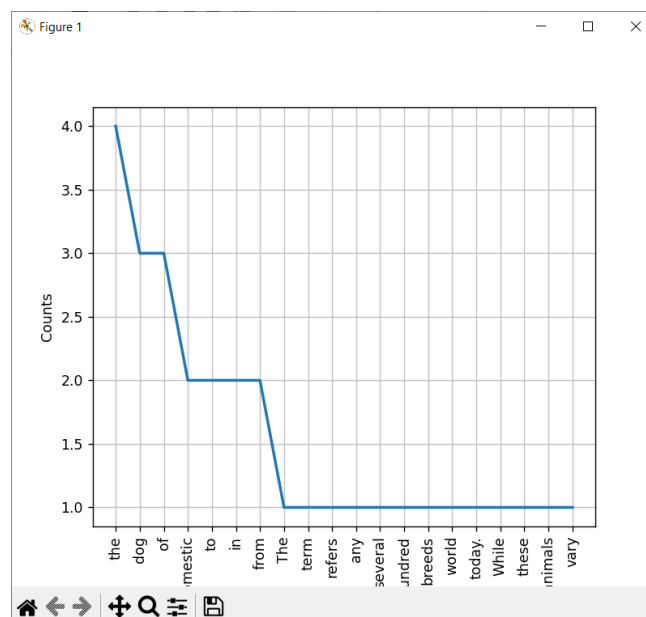
Search the output. What is the most frequent token?

You can plot a graph for those tokens using plot function like this:

```
freq.plot(20, cumulative=False)
#20 refers to number of gridlines along the x axis
```

If you experience an error in PyCharm mentioning **FigureCanvasAgg** you might need to go to File..Settings...Tools..Python scientific and disable the checkbox *"Show plots in tool window"*.

Observe the output. So what is the text about? Is "dog" the word with the highest frequency? Why?



This leads us to our next example...stop words!