WK4_Natural_Language_Processing_Task-Copy1

January 24, 2020

1 Natural Language Processing Task

In this programming task, we'll learn how to do some basic natural processing tasks, e.g. tokenisation, stemming and named entity recognition. We'll get to work with two fictitious clinical pieces of text: a biopsy report and a medical note.

```
In [2]: # OBJECTIVE:
        # - Learn how to do basic NLP tasks, i.e., tokenization, stemming, and named entity re

        # Will work with two ficitious clinical pieces of text: a biopsy report and a medical
```

1.1 Part 1: Importing packages needed

1.1.1 Importing NLTK

NLTK (Natural Language Toolkit) is a very popular suite of libraries and other resources for natural language processing. Let's tell Python that we're going to be using NLTK, with the use of the import command.

```
In [1]: # Using NLTK - Natural Language Toolkit libraries

        import nltk
```

We will also import a few more NLTK utilities for tokenisation, stemming, part of speech tagging and named entity recognition.

```
In [3]: # Will import a few more NLTK utilities for:
        # - Tokenisation, stemming, part of speech tagging and named entity recognition

        from nltk.corpus import stopwords
        nltk.download('stopwords')
        nltk.download('punkt')
        nltk.download('averaged_perceptron_tagger')
        nltk.download('maxent_ne_chunker')
        nltk.download('words')
```

```
[nltk_data] Downloading package stopwords to /home/jovyan/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
```

```
[nltk_data] Downloading package punkt to /home/jovyan/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /home/jovyan/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data] Downloading package maxent_ne_chunker to
[nltk_data]     /home/jovyan/nltk_data...
[nltk_data]   Unzipping chunkers/maxent_ne_chunker.zip.
[nltk_data] Downloading package words to /home/jovyan/nltk_data...
[nltk_data]   Unzipping corpora/words.zip.
```

Out[3]: True

1.1.2 Importing docx2txt

docx2txt is a Python-based utility which converts text within docx files into plain text. We'll import this in order to be able to work with the two clinical docx files.

In [4]:
```
# docx2txt - a Python-based utility which converts text within docx files into plain t

import docx2txt
```

Part 2: Processing a Biopsy Report

1.1.3 Loading the data

The Biopsy Report is a fictitious example for the purpose of this programming task. It contains biopsy details about a mastecotmy specimen from a patient.

The Biopsy Report is a Microsoft Word document (in a docx file format). In order for it to be processed by the NLTK package, the document requires to be converted into plain text. In this case, we are using the *process* method from the docx2txt Python package to do this. We are calling the resulting plain text "text".

In [5]:
```
# Biopsy Report
# - Contains details about a mastecotmy speciment from a patient.

# Def: Mastecotmy
# - Surgical removal of breast

# Use docx2txt.process() to process Biopsy report as text.

text = docx2txt.process('./readonly/Biopsy_Report.docx')
```

By running the code in the following cell, you'll see that text is a string.

In [6]:
```
# Convert document to plain text:

type(text)
```