TOKENIZATION
&
LEMMATIZATION

By:

M. Ashish Reddy

DSWP -> Batch-5

TOKENIZATION

- Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens.

EXAMPLE:

Natural Language Processing

['Natural', 'Language', 'Processing']

- The tokens could be words, numbers or punctuation marks. In tokenization, smaller units are created by locating word boundaries.

what are word boundaries?

- These are the ending point of a word and the beginning of the next word.
- These tokens are considered as a first step for stemming and lemmatization

Why is Tokenization required in NLP?

- **This is important because the meaning of the text could easily be interpreted by analyzing the words present in the text.**
- Let's take an example. Consider the below string:
  - "This is a cat."
- What do you think will happen after we perform tokenization on this string? We get ['This', 'is', 'a', cat'].
- There are numerous uses of doing this. We can use this tokenized form to:

1. Count the number of words in the text
2. Count the frequency of the word, that is, the number of times a particular word is present

Methods to Perform Tokenization in Python

1. Tokenization using Python’s split() function
2. Tokenization using Regular Expressions (RegEx)
3. Tokenization using NLTK
4. Tokenization using the spaCy library

1. Tokenization using Python's split() function

```
text = """Founded in 2002, SpaceX's mission is to enable humans to become a spacefaring civilization and a multi-planet
species by building a self-sustaining city on Mars. In 2008, SpaceX's Falcon 1 became the first privately developed
liquid-fuel launch vehicle to orbit the Earth."""
# Splits at space
text.split()
```

```
Output : ['Founded', 'in', '2002,', 'SpaceX's', 'mission', 'is', 'to', 'enable', 'humans',
          'to', 'become', 'a', 'spacefaring', 'civilization', 'and', 'a', 'multi-planet',
          'species', 'by', 'building', 'a', 'self-sustaining', 'city', 'on', 'Mars.', 'In',
          '2008,', 'SpaceX's', 'Falcon', '1', 'became', 'the', 'first', 'privately',
          'developed', 'liquid-fuel', 'launch', 'vehicle', 'to', 'orbit', 'the', 'Earth.']
```

2. Tokenization using Regular Expressions (RegEx)

```
1  import re
2  text = """Founded in 2002, SpaceX’s mission is to enable humans to become a spacefaring civilization and a multi-planet
3  species by building a self-sustaining city on Mars. In 2008, SpaceX’s Falcon 1 became the first privately developed
4  liquid-fuel launch vehicle to orbit the Earth."""
5  tokens = re.findall("[\w']+", text)
6  tokens
```

```
Output : ['Founded', 'in', '2002', 'SpaceX', 's', 'mission', 'is', 'to', 'enable',
          'humans', 'to', 'become', 'a', 'spacefaring', 'civilization', 'and', 'a',
          'multi', 'planet', 'species', 'by', 'building', 'a', 'self', 'sustaining',
          'city', 'on', 'Mars', 'In', '2008', 'SpaceX', 's', 'Falcon', '1', 'became',
          'the', 'first', 'privately', 'developed', 'liquid', 'fuel', 'launch', 'vehicle',
          'to', 'orbit', 'the', 'Earth']
```

3. Tokenization using NLTK

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

```
1  from nltk.tokenize import word_tokenize
2  text = """Founded in 2002, SpaceX’s mission is to enable humans to become a spacefaring civilization and a multi-planet
3  species by building a self-sustaining city on Mars. In 2008, SpaceX’s Falcon 1 became the first privately developed
4  liquid-fuel launch vehicle to orbit the Earth."""
5  word_tokenize(text)
```

```
Output: ['Founded', 'in', '2002', ',', 'SpaceX', '’', 's', 'mission', 'is', 'to', 'enable',
         'humans', 'to', 'become', 'a', 'spacefaring', 'civilization', 'and', 'a',
         'multi-planet', 'species', 'by', 'building', 'a', 'self-sustaining', 'city', 'on',
         'Mars', '.', 'In', '2008', ',', 'SpaceX', '’', 's', 'Falcon', '1', 'became',
         'the', 'first', 'privately', 'developed', 'liquid-fuel', 'launch', 'vehicle',
         'to', 'orbit', 'the', 'Earth', '.']
```

4. Tokenization using the spaCy library

```
pip install -U spacy
python -m spacy download en
```

```
from spacy.lang.en import English

# Load English tokenizer, tagger, parser, NER and word vectors
nlp = English()

text = """Founded in 2002, SpaceX's mission is to enable humans to become a spacefaring civilization and a multi-planet
species by building a self-sustaining city on Mars. In 2008, SpaceX's Falcon 1 became the first privately developed
liquid-fuel launch vehicle to orbit the Earth."""

#  "nlp" Object is used to create documents with linguistic annotations.
my_doc = nlp(text)

# Create list of word tokens
token_list = []
for token in my_doc:
    token_list.append(token.text)
token_list
```

LEMMATIZATION

- ***A method that switches any kind of a word to its base root mode is called Lemmatization.***
- In other words, Lemmatization is a method responsible for grouping different inflected forms of words into the root form, having the same meaning.

Difference between lemmatization and stemming

- For example, Lemmatization clearly identifies the base form of ‘troubled’ to ‘trouble” denoting some meaning whereas, Stemming will cut out ‘ed’ part and convert it into ‘troubl’ which has the wrong meaning and spelling errors.

- **‘troubled’ -> Lemmatization -> ‘troubled’, and error**
- **‘troubled’ -> Stemming -> ‘troubl’**

S.No	***Stemming***	***Lemmatization***
1	Stemming is faster because it chops words without knowing the context of the word in given sentences.	Lemmatization is slower as compared to stemming but it knows the context of the word before proceeding.
2	It is a rule-based approach.	It is a dictionary-based approach.
3	Accuracy is less.	Accuracy is more as compared to Stemming.
4	When we convert any word into root-form then stemming may create the non-existence meaning of a word.	Lemmatization always gives the dictionary meaning word while converting into root-form.
5	Stemming is preferred when the meaning of the word is not important for analysis. Example: Spam Detection	Lemmatization would be recommended when the meaning of the word is important for analysis. Example: Question Answer
6	For Example: “Studies” => “Studi”	For Example: “Studies” => “Study”

Example:

```
# import these modules
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

print("rocks :", lemmatizer.lemmatize("rocks"))
print("corpora :", lemmatizer.lemmatize("corpora"))

# a denotes adjective in "pos"
print("better :", lemmatizer.lemmatize("better", pos ="a"))
```

OUTPUT:

```
rocks : rock
corpora : corpus
better : good
```