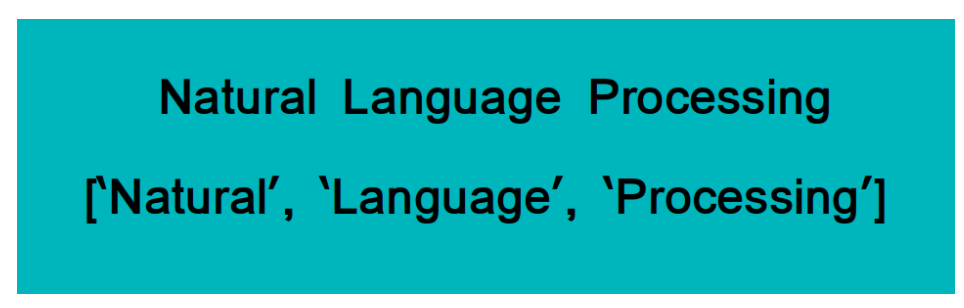
**What is Tokenization in NLP?**

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.

In Python tokenization basically refers to splitting up a larger body of text into smaller lines, words or even creating words for a non-English language.

The tokens could be words, numbers or punctuation marks.

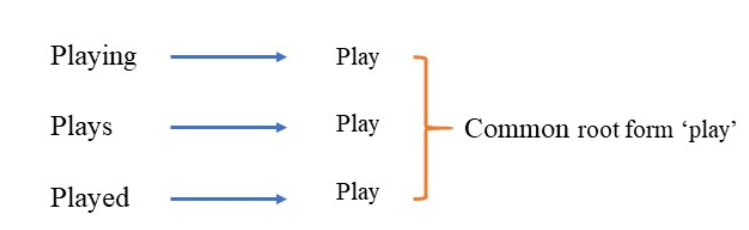
**Example :**



**What is Stemming in NLP?**

Stemming is basically removing the suffix from a word and reduce it to its root word.

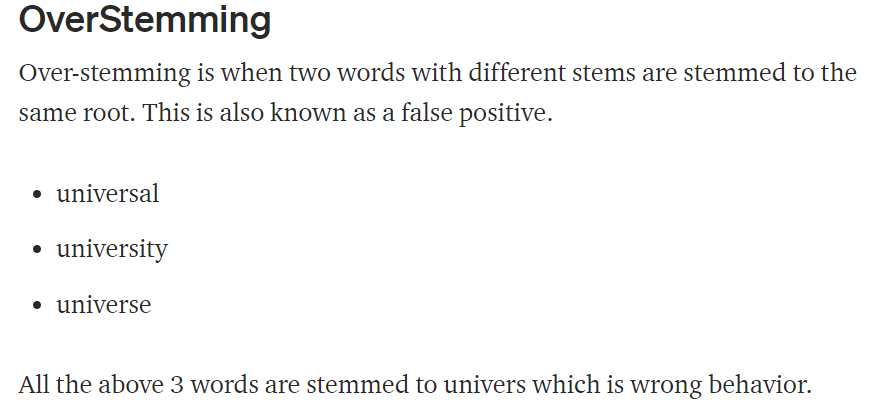
**Example:**

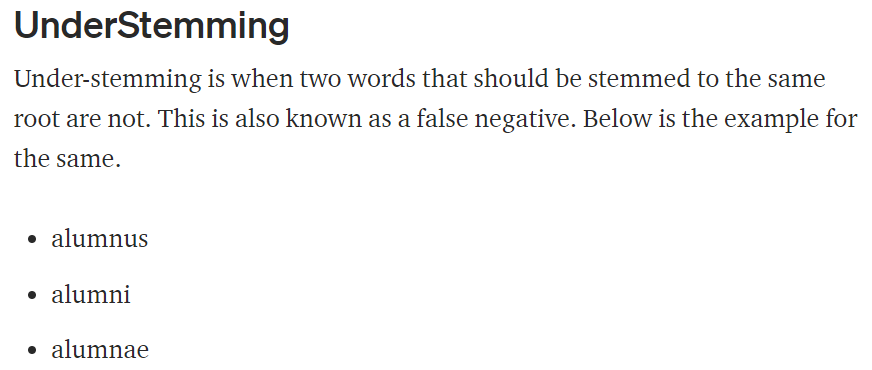
**Example 2:**

Histor

History

Historical





**Porter Stemmer**

sentence = 'Provision Maximum owed caring on go gone going was this'

output = provis maximum owe care on go gone go wa thi

**Snowball Stemmer**

sentence = 'Provision Maximum owed caring on go gone going was this'

output = provis maximum owe care on go gone go was this

**Lancaster Stemmmer**

sentence = 'Provision Maximum owed caring on go gone going was this'

output = provid maxim ow car on go gon going was thi

**Lemmatization**

Lemmatization is the process of converting a word to its base form. lemmatization considers the context and converts the word to its meaningful base form.

For example, lemmatization would correctly identify the base form of ‘caring’ to ‘care’.

**Bag of Words:**

Bag of words is a Natural Language Processing technique of text modelling. In technical terms, we can say that it is a method of feature extraction with text data. This approach is a simple and flexible way of extracting features from documents.

A bag of words is a representation of text that describes the occurrence of words within a document. We just keep track of word counts and disregard the grammatical details and the word order. It is called a “bag” of words because any information about the order or structure of words in the document is discarded.

Example:

Sent 1 – good boy

Sent 2 – good girl

Sent 3 – boy girl good

1. Convert to lower case
2. Apply Stop words

Sent 1 – He is a good boy

Sent 2 – She is a good girl

Sent 3 – boy and girl are good

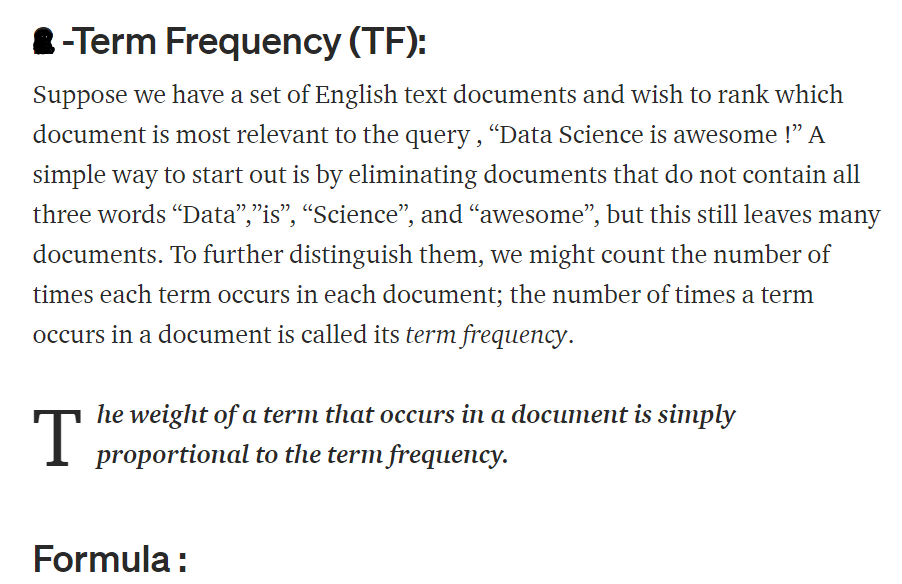
|  |  |
| --- | --- |
| Words | Frequency |
| good | 3 |
| boy | 2 |
| girl | 2 |

**Converting to Vector**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Feature 1 | Feature 2 | Feature 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Good | Boy | Girl | O/P |
| Sent 1 | 1 | 1 | 0 |  |
| Sent 2 | 1 | 0 | 1 |  |
| Sent 3 | 1 | 1 | 1 |  |

**TF – IDF – [Term Frequency – Inverse Document Frequency]:**



No of reptations of words in sentence

No of Words in sentence

Example:

Sent 1 – good boy

Sent 2 – good girl

Sent 3 – boy girl good

1. Convert to lower case
2. Apply Stop words

Sent 1 – He is a good boy

Sent 2 – She is a good girl

Sent 3 – boy and girl are good

|  |  |
| --- | --- |
| Words | Frequency |
| good | 3 |
| boy | 2 |
| girl | 2 |

**Converting to Vector:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **TERM FREQUENCY** |  |  |
|  | Sent 1 | Sent 2 | Sent 3 |
| good | 1 / 2 | 1 / 2 | 1 / 3 |
| boy | 1 /2 | 0 | 1 / 3 |
| girl | 0 | 1 / 2 | 1 / 3 |

**Inverse Document Frequency (IDF):**

Inverse Document Frequency (IDF) is a weight indicating how commonly a word is used. The more frequent its usage across documents, the lower its score. The lower the score, the less important the word becomes.



log (No of Sentences / No of Sentences Containing words)

**Converting to Vector:**

|  |  |
| --- | --- |
| Words | IDF |
| good | log (3/3) = 0 |
| boy | log (3/2) = 0.176091259 |
| girl | log (3/2) = 0.176091259 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Feature 1 | Feature 2 | Feature 3 | O / P |
|  |  | Good | Boy | Girl |  |
| Sent 1 |  | 0 | 1 /2 \* log (3/2) = 0.08804563 | 0 |  |
| Sent 2 |  | 0 | 0 | 1 /2 \* log (3/2) = 0.08804563 |  |
| Sent 3 |  | 0 | 1 /3 \* log (3 /2) = 0.058697086 | 1 /3 \* log (3/2) = 0.058697086 |  |

**TF \* IDF:**