**Introduction To NLP**

**All the code and easy understanding in : (**[**Google Colab**](https://colab.research.google.com/drive/1ZX3dcjbauz8jw75qA57KHRYBnYK9XApD?usp=sharing)**)**

Writng a basc Sentenc is easy Rigt? I apologize for the spelling errors I made, but this is all because the ‘autocorrect' was off 😠. Ever consider the possibility that you're speaking with a bot? over the phone, or on Telegram and Discord? If you're wondering why you weren't aware of anything Heh, heh 😁.

When it comes to checking claims of plagiarism or copyright, things have gotten considerably simpler. Visit any of the sites and use the software to quickly verify it. There are 6500 different languages spoken worldwide, so what would you do if you were a writer who wanted your work to be translated into another language?

Have you ever wondered how many people remark on different social media and e-commerce websites? Wouldn't it be better if someone or something could analyze all of these comments and give the producers a sense of the users' attitudes so that they could better understand their customers?

Let's examine this from a societal standpoint. Everyone today utilizes social media, correct? According to a survey, several effects could occur there, such as detecting nasty remarks, suicidal behavior, harsh conversations, and even crime 😮. If and only if the machine could grasp our language, everything could be kept an eye on and prevented. Are you able to come up with a solution to the aforementioned issues?

You made a good approximation. Natural Language Processing, a subfield of artificial intelligence, enables machines to comprehend "Language" and do all the tasks listed above in a "Human-Like" fashion. Well, sometimes machines are more articulate than we are 😅.

As the population grows and social media becomes more and more popular, zettabyte-scale amounts of data (also known as "Big Data") are produced (21 zeroes). As a result, it has become vital to utilize NLP to evaluate this data to better understand human behavior because it is largely present in an unstructured form (>80%). This can also assist to save a lot of time...

Some more applications of NLP :

1. Sentiment Analysis
2. Chatbot
3. Speech Recognition (Siri, Alexa)
4. Machine Translation
5. Spell Checking
6. Keyword Searching
7. Information extraction
8. Advertisement matching

After learning about the uses and significance of NLP in the modern marketplace, you must be concerned that learning this Technology will be very challenging. But don't worry, we'll make studying more enjoyable and uncomplicated in this "Practical Handbook on NLP." where you'll feel secure enough to create models of the real world at the end of it. So, let’s keep reading...

NLP is divided into 7-10 steps just like how a child would learn a language for example a paragraph.

1. Segmentation - Breaking the big paragraph into segments. (Like each with full stops)
2. Tokenizing - Take those sentences and break them into words.
3. Stop Words - Removing the non-important words like - ‘and’ ‘the’
4. Stemming - verb forms, Like we should tell the child that walks, walking, and walking all have similarities like (walk + ).
5. Lemmatization - Then understanding the base words like Am, Is, are
6. Speech Tagging - Understanding Nouns, Verbs, etc.
7. Named Entity Tagging - Also the child should know the important names of capital, animals, and state right?
8. Classification - As a Machine is still only a machine but to make it is intelligent enough to make it understand the language.

**NATURAL LANGUAGE with Python and NLTK - Tokenizing Words and Sentences**

Is there a toolkit that can teach us all the basics so we can do all these above-mentioned steps?

The answer is yes !!! .

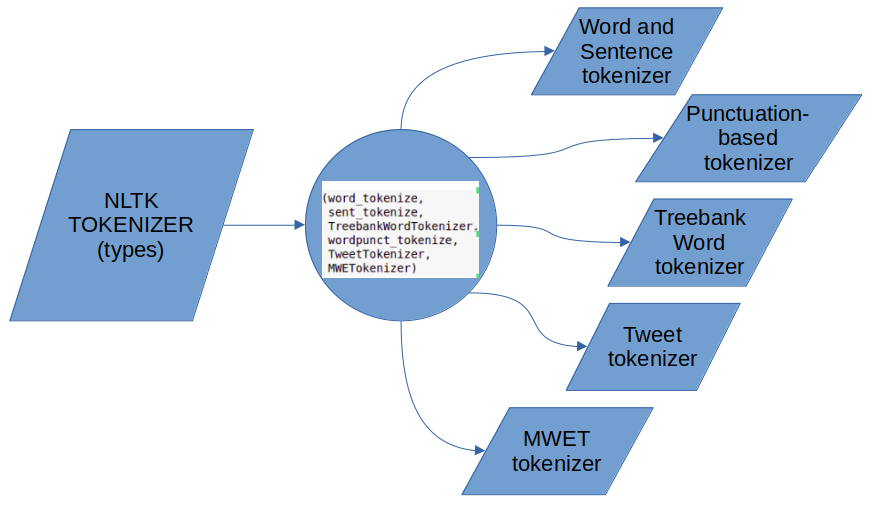
NLTK - Natural Language Toolkit ([Documentation](https://www.nltk.org/)) is a python toolkit for Natural Language Processing. It includes everything that you need to know to get started in NLP and we will perform the steps mentioned above with it in **(**[**Google Colab**](https://colab.research.google.com/drive/1ZX3dcjbauz8jw75qA57KHRYBnYK9XApD?usp=sharing)**)**.

Things to be remembered in the Google Colab →

1. Segmentation in the Colab file - Think it as the my\_text string is the segment of a bigger paragraph like a part of my whole portfolio
2. Tokens - These are the broken down words or sentences from a big paragraph into simpler language-assignable elements or comprehensible components.
3. The importance of Tokens - Tokenization because this is also used while any programming language gets executed.

The initial stage in every NLP pipeline is tokenization. It significantly affects the remainder of your pipeline. Natural language text and unstructured data are divided into discrete data units through the process of tokenization. A document can be directly represented by a vector of token occurrences in that text.

This instantly converts unstructured text or strings into numerical data types appropriate for machine learning. Additionally, a computer can use them directly to start up helpful reactions and actions. In a machine learning pipeline, they could also be utilized as features to initiate more sophisticated actions or judgments.



Types of NLTK Tokenizer

The importance of the Tokenizer pipeline task in any NLP task or project. Keep practicing the techniques.

Accessing the NLTK Tokenizer code and practice - **(**[**Google Colab**](https://colab.research.google.com/drive/1ZX3dcjbauz8jw75qA57KHRYBnYK9XApD?usp=sharing)**)**

Let's go to the second pipeline - Stop Words.

**NATURAL LANGUAGE with Python and NLTK - STOP WORDS**

When analyzing natural language, stop words are commonly disregarded. Despite the fact that these are some of the most frequently used terms across all languages, the text does not learn anything from them (along with articles, prepositions, pronouns, conjunctions, etc.). In English, stop words include "the," "a," "an," "so," and "what."

Every language used by humans has numerous stop words. These terms can be made more focused on the key information by removing the low-level information. In other words, the removal of such statements has no impact on the model we create to accomplish our goal.

Because they enable us to focus on the important words rather than the words that are overused in a language, stop words are crucial in many applications. The terms "how" and "to" are so prevalent in the context of search engines that the search engine will uncover a lot more pages that contain them than pages that provide information about creating information retrieval applications. For instance, the search engine will turn up far more pages that contain the words "how to construct information retrieval applications" than those that provide knowledge on doing so.

Here are a few instances of jobs where stop words may be used:

* Deleting words from the feature space can be done with supervised machine learning.
* Before creating clusters, stop words are removed during clustering.
* keeping stop words out of your information index
* Stop words are excluded from text summarizing scores and are not taken into account when calculating summary ratings.

Accessing the NLTK Stop Words code and practice - **(**[**Google Colab**](https://colab.research.google.com/drive/1ZX3dcjbauz8jw75qA57KHRYBnYK9XApD?usp=sharing)**)**

**NATURAL LANGUAGE with Python and NLTK - STEMMING**

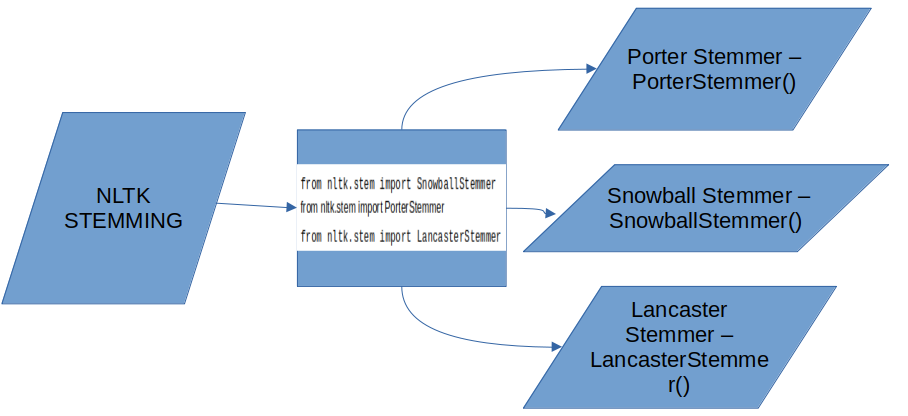
A method known as stemming reduces word inflection to their base forms as part of the preparation of text, words, and documents for text normalization.

Inflection is the process through which a word is changed to express several grammatical categories, including tense, case, voice, aspect, person, number, gender, and mood, according to Wikipedia. The presence of several inflected forms within the same text adds redundancy to the NLP process, even though a word may have a variety of inflected forms.

For instance, the word "connect" serves as the stem of the three phrases connections, connected, and connects.

In the English language, there are several ways to say the same thing. These variations in a text corpus produce redundant data for building NLP or machine learning models. These models may not perform as expected.

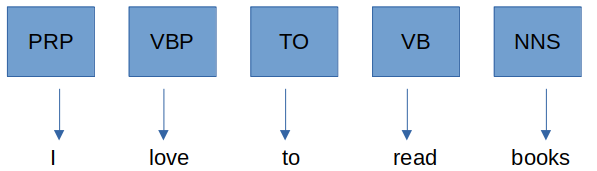
Text must be normalized by removing duplicates and stemming words to their simplest forms.



How to use Stemming can be seen in the Google Colab File.

**NATURAL LANGUAGE with Python and NLTK - POS-TAGGING**

According to its meaning and context, the text for a particular piece of a speech is marked up using the POS Tagging (Parts of Speech Tagging) technique. It is in charge of reading texts in a language and assigning a specific token to each word (Parts of Speech). It is also known as grammatical tagging.



How to use Tagging can be seen in the Google Colab File.

**NATURAL LANGUAGE with Python and NLTK - CHUNKING**

An NLP technique called "chunking" is used to turn small informative units into bigger ones. Chunking is often used to put together collections of "noun phrases." POS tagging and regular expressions are used to create the sentence structure. Also known as shallow parsing, the resulting group of words is referred to as a "chunk."

Between the roots and the leaves, there may only be one or two levels in shallow parsing; in deep parsing, there are many levels. Light parsing or chunking are other names for shallow parsing.

There are no set guidelines for chunking; nevertheless, you can combine the guidelines as necessary.

For instance, you must tag the sentence's noun, verb (past tense), adjective, and coordinating junction. You can apply the following rule:

chunk:{<NN.?>\*<VBD.?>\*<JJ.?>\*<CC>?}

Chunking is employed to identify entities. The part of the sentence that the machine uses to determine the value for any intention

**NATURAL LANGUAGE with Python and NLTK - CHINKING**

The act of "chinking" is simply the separation of the "chink" from the "chunk." These patterns are regular expressions that have been altered and tailored to match Part-of-Speech (POS) tags and their sequences.

The syntax is quite similar; the only difference is that the chink}{ following the chunk must be indicated with {} rather than.