Text Preprocessing - NLP LECTURE 3  
Basic Preprocessing techniques

Author : Harshavardhan

INSTRUCTOR: Nitish Sir, CampusX

horizontal line

# Text Representation

Text preprocessing in NLP refers to the series of steps taken to clean, transform, and prepare raw text data before feeding it into a machine learning or natural language processing model. The goal of text preprocessing is to standardize the text, remove noise, and create a consistent and meaningful representation for analysis and modeling.

## Techniques covered in this lecture:

* Lower Casing
* Removing HTML Tags
* Removing URLs
* Removing Punctuation
* Chat words treatment
* Spelling Correction
* Removing Stopwords
* Handling Emojis
* Tokenization
* Stemming
* Lemmatization

## Importing the csv file



## 

## 

## 

## 

## 

## Lowercasing

Lowercasing refers to the process of converting all letters in a piece of text to their lowercase form. The significance of lowercasing in simple words is as follows:

* **Uniformity**: Lowercasing makes all the text consistent by treating different cases (uppercase and lowercase) of the same word as identical. This ensures that "apple," "Apple," and "APPLE" are considered the same word.
* **Word** **Comparisons**: Lowercasing enables easier word comparisons and matching, making it simpler to find occurrences of specific words regardless of their case.



## Removing HTML tags

Removing HTML tags in simple words involves eliminating any markup code or formatting elements present in the text data. These tags are used in web pages to structure and style content, but they may not carry meaningful information for certain NLP tasks. Removing HTML tags ensures that only the plain text content remains, making it more suitable for text analysis and processing.



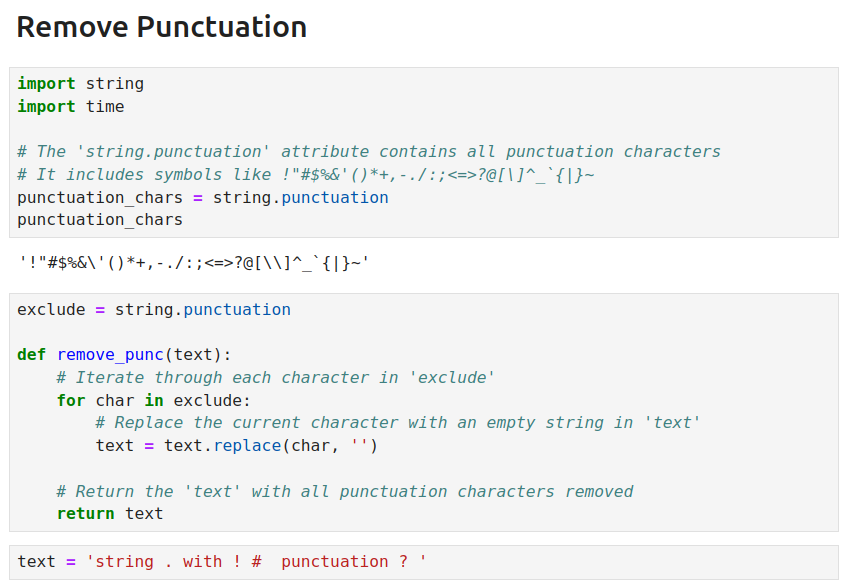
## Removing URL’s

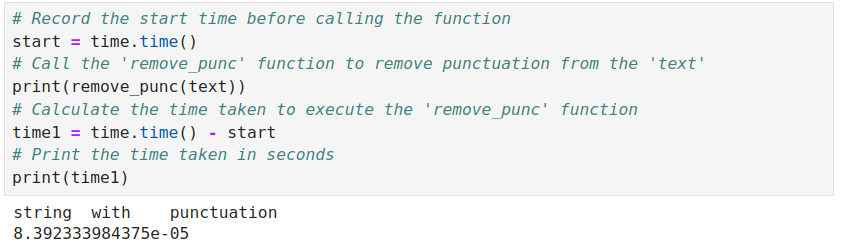
In NLP preprocessing, removing URLs from the text data is an essential step to clean and prepare the text for further analysis or modeling. URLs, which are web links, often do not carry significant meaning for most NLP tasks and can introduce noise or disrupt the analysis.

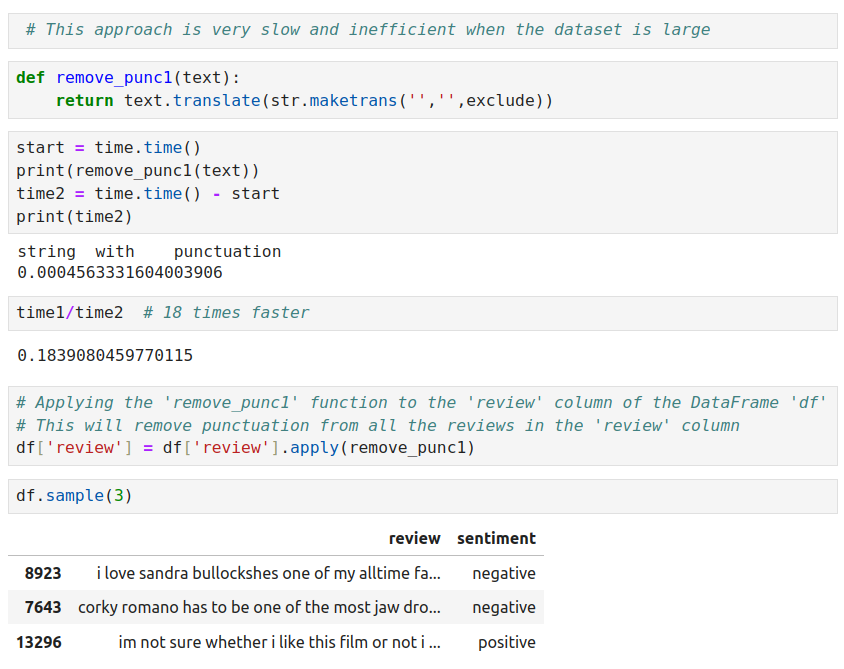


## Removing Punctuations

In NLP preprocessing, removing punctuations from the text data is a common practice to clean and simplify the text for further analysis and modeling. Punctuations are characters like periods, commas, exclamation marks, question marks, and other symbols used to indicate pauses, sentence boundaries, or emphasize certain expressions. While they are essential for grammatical correctness in human language, they often do not contribute much to the meaning or context in many NLP tasks.





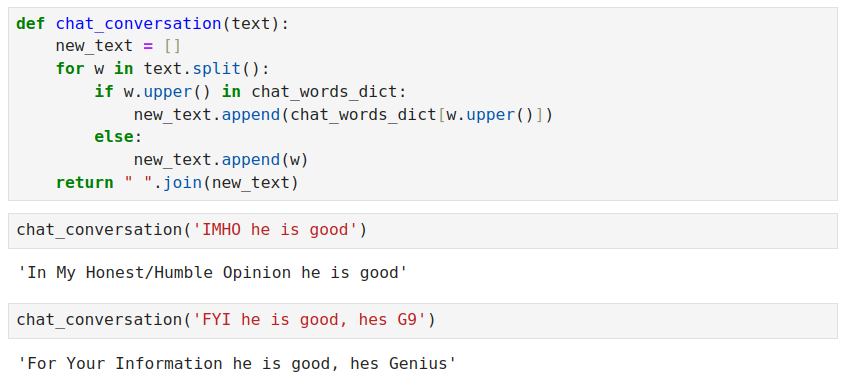


## Chat word Treatment

In NLP preprocessing, chat word treatment refers to the process of handling informal or abbreviated words commonly used in chats, social media, and online communication. These chat words may include acronyms, emojis, emoticons, and slang, which can be challenging for NLP models to interpret correctly.

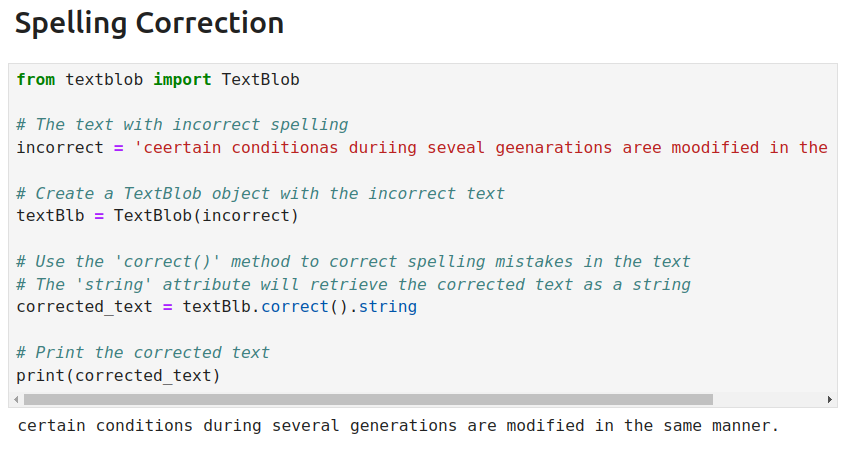






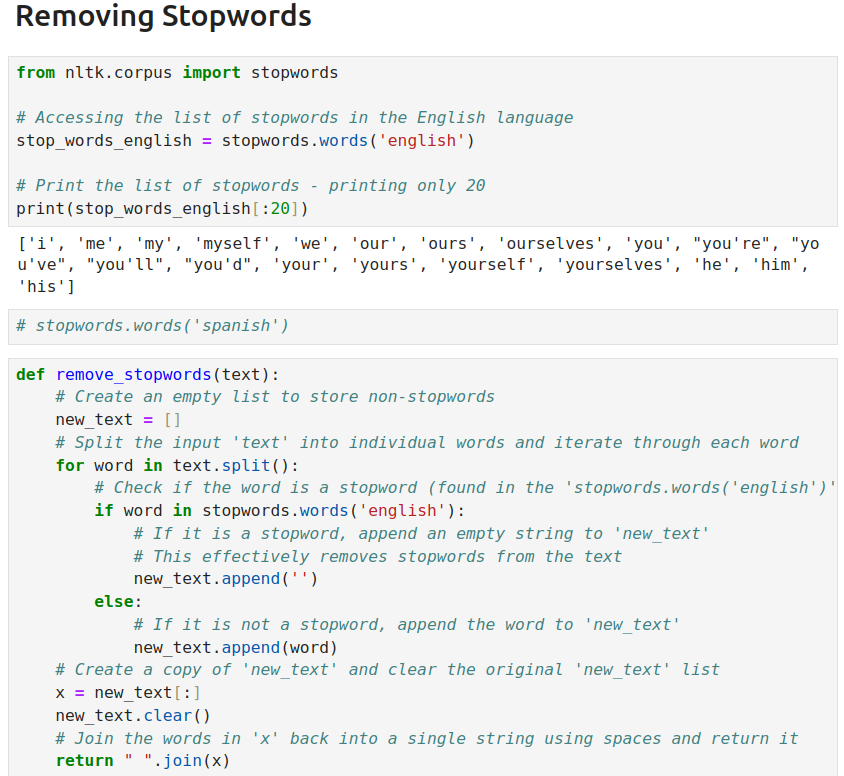
## Spelling Correction

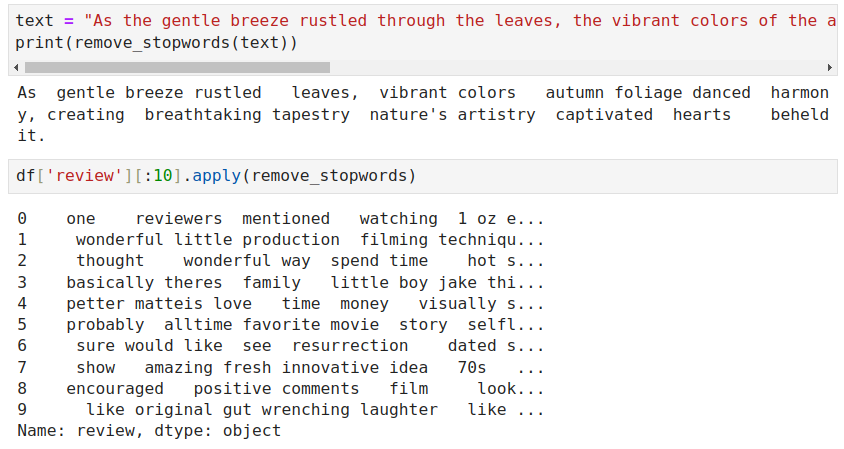
Spelling correction is a crucial step in NLP preprocessing, aimed at fixing spelling errors in the text data. Spelling errors can occur due to typos, keyboard mistakes, or other human errors during data input or text generation processes. These errors can negatively impact the performance of NLP models and lead to incorrect analysis and interpretations.



## Removing stopwords

In NLP preprocessing, removing stop words is a common practice to clean and prepare text data for analysis or modeling. Stop words are common words that appear frequently in the language but often do not carry significant meaning or contribute much to the context of the text. Examples of stop words include "the," "and," "in," "is," "of," "to," etc.





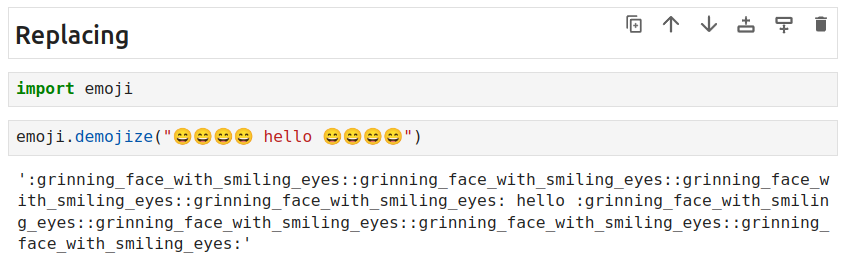
## Handling emojis

Handling emojis in NLP preprocessing involves managing and processing these visual representations of emotions, expressions, or objects that are commonly used in online communication. Emojis can be challenging for NLP models to interpret because they do not have direct linguistic meanings but can carry important contextual information in text data.

Here's how handling emojis is important and how it is done:

* Emoji Removal: In certain NLP tasks, emojis may not contribute to the analysis and could be treated as noise. Removing emojis can simplify the text and improve model performance.
* Sentiment Analysis: Emojis are commonly used to express emotions. For sentiment analysis tasks, mapping emojis to sentiment scores or categories can enhance the model's understanding of emotional context.



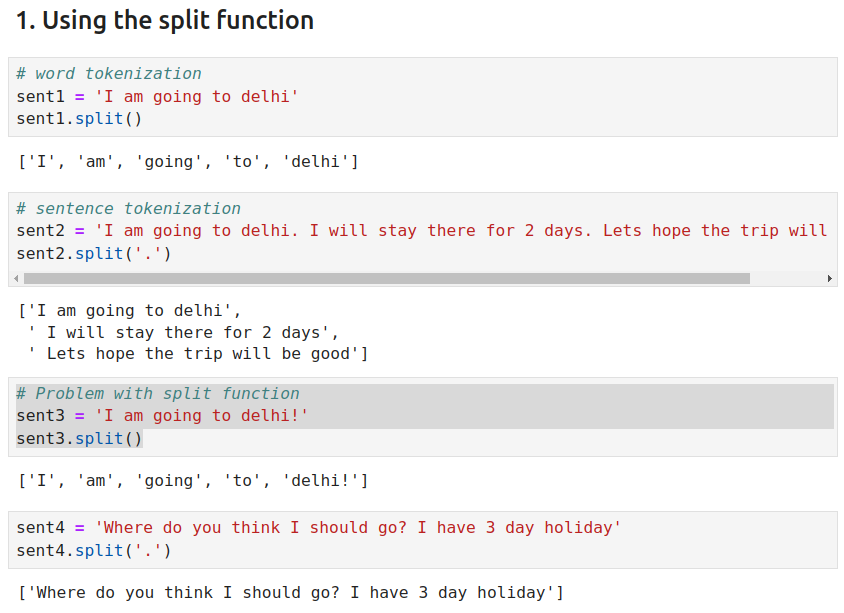


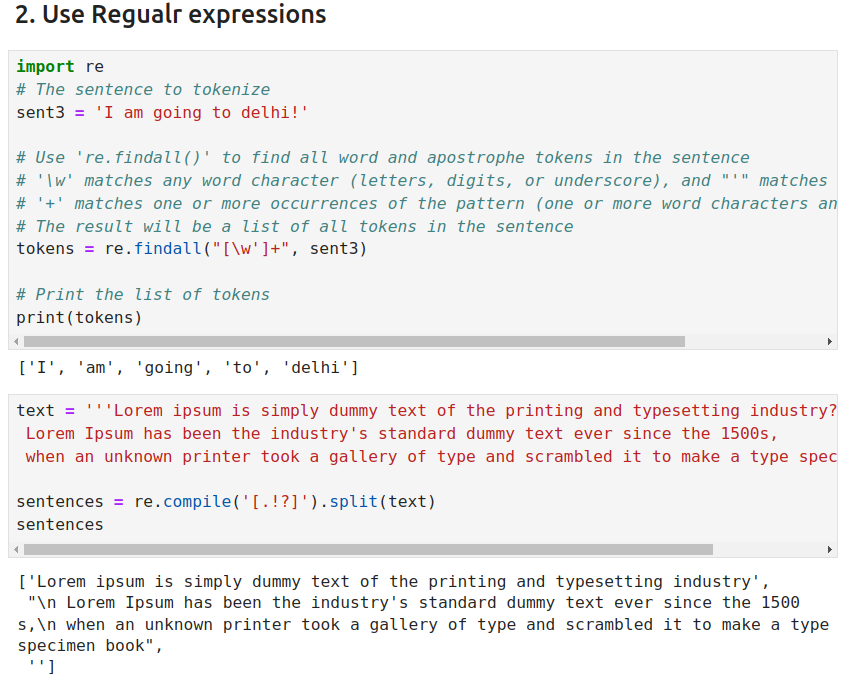
## Tokenization

Tokenization in NLP preprocessing is the process of breaking down a text or a sentence into smaller units called tokens. These tokens are typically words or subwords, and tokenization is a fundamental step in preparing text data for various NLP tasks.

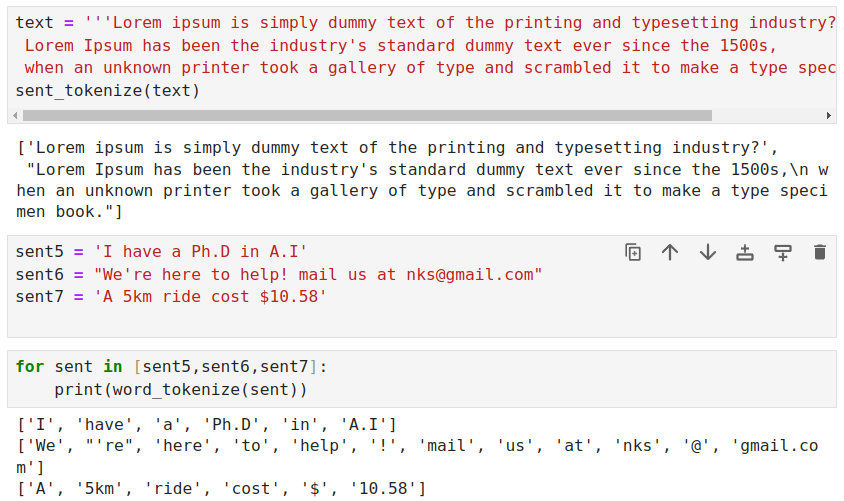
Tokenization can be performed at different levels:

* Word Tokenization: Dividing the text into individual words or word-level tokens. For example, "Tokenization is essential for NLP" becomes ['Tokenization', 'is', 'essential', 'for', 'NLP'].
* Subword Tokenization: Splitting words into subword units, such as morphemes or character n-grams. This is useful for languages with complex morphology or to handle out-of-vocabulary words.











# Stemming

In grammar, inflection is the modification of a word to express different grammatical categories such as tense,case,voice,aspect,person,number,gender,and mood.

eg. Walk - walk,walking,walked,walker,walks

Stemming is the process of reducing inflection in words to their root forms such as mapping a group of words to the same stem even if the stem itself is not a valid word in the Language.

* its mostly used in information retrieval systems (google search)

Stemmer : Algorithms using which we can perform stemming, examples are

* Porter stemmer (for english)
* Snowball Stemmer(for other language)



## Lemmatization

Lemmatization, unlike stemming,reduces the inflected words properly ensuring that the root word belongs to the language. In Lemmatization the root word is called Lemma. A lemma (plural lemmas or lemmata) is the canonical form, dictionary form, or citation form of a set of words.

* almost same as stemming, but the root word here is a valid word
* It takes a little longer time when compared to stemming
* if we don't have to show the output to the user then we can use stemming
* else we can use lemmatization

Lemmatization is done using a lexical dictionary instead of an algorithm. The WORDNET lexical dictionary is used here

