**Natural Language Processing**

# Natural Language Processing

NLP is the technology that is used by machine to understand, analyses, manipulate and interpret

Human’s language.

NPL is a part of Computer Science, Artificial Intelligence and Human Language

## Application of NLP

Alexa, spelling checking, Email spam, and language translator

## Components of NLP

Natural Language Processing (NLP)

Natural Language Understanding (NLU) Natural Language Generation (NLG)

## NLP Challenges

Contextual words and Phrases and Homonyms

Synonyms 🡪 different ways

Irony and Sarcasm

Ambiguity

Errors in text or speech

Colloquialisms and slang

Domain-specific language

Low-resource language

Lack of research and development

## NLP Libraries

Scikit-Learn NLTK SpaCy

TextBlob AllenNLP

# NPL Pipeline

Feature Engineering

Pre-Processing

Text Cleaning

Data Collection

Deployment

Monitoring and Model Updating

Evaluation

Modelling

## Data Acquisition

Company Database Web Scraping Server or System

API Survey

## Text Pre-processing

* **Text Cleaning:** In text cleaning, we do HTML tag removing, emoji handling, Spelling checking, etc.
* **Basic Preprocessing:** In basic programming we do tokenization (Word or sent tokenization, stop word removal, removing digit, lower casing).
* **Advance Preprocessing:** In this step we do POS tagging, Parsing, and Conference resolution.

## Featured Engineering

* One Hot Encoder
* Bag of Word (BOW)
* N-grams
* Tf-Idf
* Word2vec

## Modelling / Model Building

**Approaches to building model:**

* Heuristic Approach
* Machine Learning Approach
* Deep Learning Approach
* Cloud API

## Model Evaluation

* **Intrinsic Evaluation:** In this evaluation, we use multiple metrics to check our model such as Accuracy, Recall, Confusion Metrics, Perplexity, etc.
* **Extrinsic Evaluation:** This evaluation is done after deployment. This is the business-centric approach.

# Understanding Textual Data

## Elements of text

**Hierarchy of Text:** No. of words, No. of Alphabet

**Tokenization:** Separation of words in sentence and word.

**Vocabulary:** Counting of unique words and repeating words.

**Punctuation:** Removing ?, <, >, ;, :, “ ”

**Part of Speech:** what is word like verb, none, adverb

**Roots of a Word:** study, studies, studying, play, playing, pla

**Base of a Word:** play, played, and playing

**Stop Word:** Non important, no need of words like a, is , now

# Text Pre-processing

Lowercasing Tokenization

Remove HTML tags Stop words removal

Remove URLs N-Grams

Removing Punctuation Word Sense Disambiguation

Chat word Treatment Count Vectorizer

Spelling Correction Lemmatization

TF-IDF Vectorizer

Hashing Vectorizer

# Tokenization

Tokenization is used in natural language processing to split paragraphs and sentences into smaller units that can be more easily assigned meaning.

The first step of the NLP process is gathering the data (a sentence) and breaking it into understandable parts (words).

One of the steps to be perform in the NLP. It convert unstructured text into a proper format of data.

**Input Text**

One of the steps to be perform in the NLP.

**Sentence**

**Tokenization**

It convert unstructured text into a proper format of data.

**Word**

perform

be

One

Of

To

Steps

**Tokenization**

convert

It

unstructured

NLP

the

in

of

format

proper

a

into

text

data

## Practical - Word and Sentence Tokenization

var = "Natural language processing (NLP) is a subfield of computer science and especially artificial intelligence. It is primarily concerned with providing. computers with the ability to process data encoded in natural language and is thus closely related to information retrieval, knowledge representation and computational linguistics, a subfield of linguistics."  
from nltk.tokenize import word\_tokenize, sent\_tokenize

sent = sent\_tokenize(var)

sent

for i in sent:

print(i)

word = word\_tokenize(var)

word

# Stop Words Removal

The words which are generally filtered out before processing a natural language are called stop words.

These are actually the most common words in any language (like articles, prepositions, pronouns, conjunctions, etc) and does not add much information to the text. Examples of a few stop words in English are “the”, “a”, “an”, “so”, “what”.

**Text with stop words After removing Stop words**

I believe this would help ‘I’, ‘believe’, ‘would’, ‘help’

The A reader understand ‘reader’, ‘understand’, ‘stop’

How stop words works ‘words’, ‘works’

As well as realize its importance ‘well’, ‘realize’, ‘importance’

## Practical – Stop words removal

from nltk.corpus import stopwords

from string import punctuation

stop = stopwords.words("english")

stop\_word\_list = list(punctuation)+stop

for i in word: word after tokenization

if i not in stop\_word\_list:

print(i)

# Stemming and Lemmatization

## Stemming

Stemming is a technique used to extract the based form of the words by removing affixes from them.

Changing chang

Stem

Changed chang

Change chang

Studying studi

Stem

Studies studi

Study studi

## Practical – Stemming

from nltk.stem import LancasterStemmer, RegexpStemmer, PorterStemmer, SnowballStemmer

l = LancasterStemmer()

r = RegexpStemmer('ing')

p = PorterStemmer()

s = SnowballStemmer('english')

l.stem('changing')

r.stem('changing')

p.stem('changing')

s.stem('changing')

## Lemmatization

Lemmatization techniques is like stemming.

The output we will get after lemmatization is called ‘lemma’.

After lemmatization, we will be getting a valid word that means the same thing.

Studying Study

Lemmatization

Studies Study

Study Study

## Practical – Lemmatization

from nltk.stem import WordNetLemmatizer

wl = WordNetLemmatizer()

wl.lemmatize('mice')

# N – Grams

N – grams are continuous sequences of words or symbols or tokens in a document.

They can be defined as the neighboring sequences of items in a document.

## Practical -- n-grams

x = 'i am Haroon Rasheed i am computer science student i am good boy i am best'

from nltk.tokenize import word\_tokenize

w = word\_tokenize(x)

from nltk.collocations import BigramCollocationFinder, TrigramCollocationFinder, ngrams

b = BigramCollocationFinder.from\_words(w)

b.ngram\_fd

t = TrigramCollocationFinder.from\_words(w)

t.ngram\_fd

n = ngrams(w,4)

for i in n: print(i)

# Count Vectorizer

Count Vectorizer means breaking down a sentence or any text into words by performing preprocessing tasks like converting all words to lowercase. Thus removing special characters.

## Practical – Count Vectorizer

l = ['i am Haroon Rasheed', 'i am computer science student', 'i am good boy', 'i am best']

import pandas as pd

df = pd.DataFrame({'name': l})

df

from sklearn.feature\_extraction.text import CountVectorizer

cv = CountVectorizer()

new\_df = cv.fit\_transform(df['name']).toarray()

new\_df

cv.vocabulary\_

# Word Sense Disambiguation

Word Sense Disambiguation is an important method of NLP by which the meaning of a word is determined, which is used in a particular context.

## Practical – Word Sense Disambiguation

y = "A computer mouse (plural mice; also mouses) is a hand-held pointing device that detects two-dimensional motion relative to a surface. This motion is typically translated into the motion of the pointer (called a cursor) on a display, which allows a smooth control of the graphical user interface of a computer."

from nltk.wsd import lesk

from nltk.tokenize import word\_tokenize

l = lesk(word\_tokenize(y), 'mouse')

l.definition()