# Natural Language Processing

techtrunk

www.techtrunk.in

## Objectives

Introduction to NLP

Applications of NLP

Tokenization

Stemming & Lemmatization

Count Vector

TF IDF

Co occurrence Matrix

Bag of Words

Text Classification

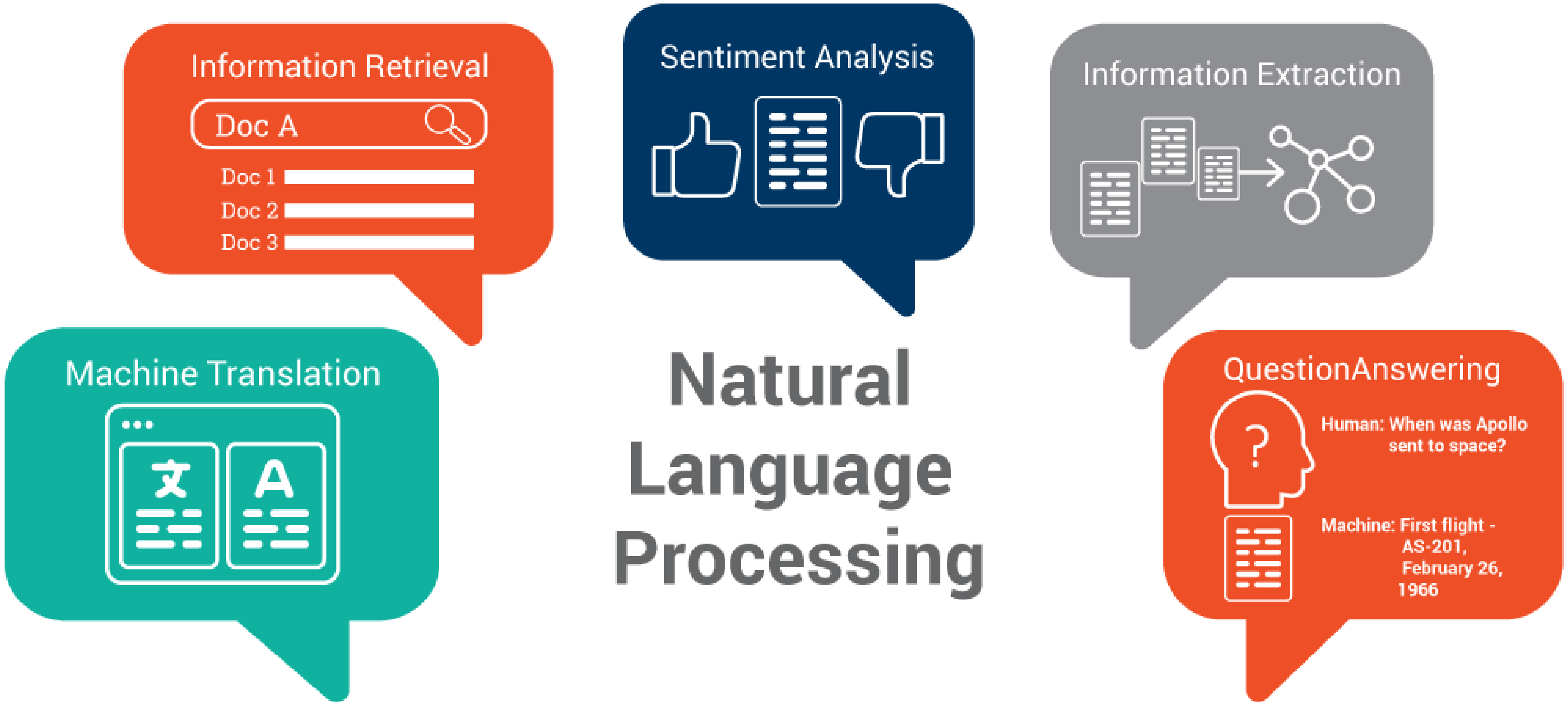
# What is Natural Language Processing?

The process of computer analysis of input provided in a human language (natural language), and conversion of this input into a useful form of representation.

The field of NLP is primarily concerned with getting computers to perform useful and interesting tasks with human languages.

The field of NLP is secondarily concerned with helping us come to a better understanding of human language.

# Applications of NLP



# Applications of NLP

Machine Translation

Database Access

Information Retrieval

- Selecting from a set of documents the ones that are relevant to a query

Text Categorization

- Sorting text into fixed topic categories

Extracting data from text

- Converting unstructured text into structure data

Spoken language control systems

Spelling and grammar checkers

## Components of NLP

**Natural Language Understanding**

Mapping the given input in the natural language into a useful representation.

Different level of analysis required:

- ***morphological analysis,***
- ***syntactic analysis,***
- ***semantic analysis,***
- ***discourse analysis***, …

**Natural Language Generation**

Producing output in the natural language from some internal representation.

Different level of synthesis required:

- ***deep planning***
- ***syntactic generation***

## Forms of Natural Language

The input/output of a NLP system can be:

- written text
- speech

To process written text, we need:

- lexical, syntactic, semantic knowledge about the language
- discourse information, real world knowledge

To process spoken language, we need everything required to process written text, plus the challenges of speech recognition and speech synthesis.

## Why NL Understanding is hard?

Natural language is extremely rich in form and structure

One input can mean many different things. Ambiguity can be at different levels.

Many input can mean the same thing.

Interaction among components of the input is not clear.

## Knowledge of Language

**Phonology** – concerns how words are related to the sounds that realize them.

**Morphology** – concerns how words are constructed from more basic meaning units called morphemes. A morpheme is the primitive unit of meaning in a language.

**Syntax** – concerns how can be put together to form correct sentences and determines what structural role each word plays in the sentence and what phrases are subparts of other phrases.

**Semantics** – concerns what words mean and how these meaning combine in sentences to form sentence meaning. The study of context-independent meaning.

## Knowledge of Language

**Pragmatics** – concerns how sentences are used in different situations and how use affects the interpretation of the sentence.

**Discourse** – concerns how the immediately preceding sentences affect the interpretation of the next sentence. For example, interpreting pronouns and interpreting the temporal aspects of the information.

**World Knowledge** – includes general knowledge about the world. What each language user must know about the other’s beliefs and goals.

## Simple Text Preprocessing

## What exactly are regular expressions?

Strings with a special syntax

Allow us to match patterns in other strings

Applications of regular expressions:

- Find all web links in a document
- Parse email addresses, remove/replace unwanted characters

## Why preprocess?

- Helps make for better input data
  - When performing machine learning or other statistical methods
- Examples:
  - Tokenization to create a bag of words
  - Lowercasing words
- Lemmatization/Stemming
  - Shorten words to their root stems
- Removing stop words, punctuation, or unwanted tokens
- Good to experiment with different approaches

## Preprocessing example

Input text: Cats, dogs and birds are common pets. So are fish.

Output tokens: cat, dog, bird, common, pet, fish

## What is tokenization?

Turning a string or document into **tokens** (smaller chunks)

One step in preparing a text for NLP

Many different theories and rules

You can create your own rules using regular expressions

Some examples:

- Breaking out words or sentences
- Separating punctuation
- Separating all hashtags in a twee

## Why tokenize?

Easier to map part of speech

Matching common words

Removing unwanted tokens

"I don't like Sam's shoes."

"I“, "do“, "n’t”, "like“, "Sam", “’s”, "shoes", "."

## Other nltk tokenizers

**sent_tokenize:** tokenize a document into sentences

**regexp_tokenize:** tokenize a string or document based on a regular expression pattern

**TweetTokenizer:** special class just for tweet tokenization, allowing you to separate hashtags, mentions and lots of exclamation points!!!

## Stemming

Stemming and Lemmatization are the basic text processing methods for English text. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root.

## Lemmatization

- Lemmatisation (or lemmatization) in linguistics, is the process of grouping together the different inflected forms of a word so they can be analysed as a single item.

- Lemmatisation is closely related to stemming. The difference is that a stemmer operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech. However, stemmers are typically easier to implement and run faster, and the reduced accuracy may not matter for some applications.

## Part of Speech Tagging

- Part-of-speech tagging is one of the most important text analysis tasks used to classify words into their part-of-speech and label them according the tagset which is a collection of tags used for the pos tagging.

- Part-of-speech tagging also known as word classes or lexical categories.

## Different types of Word Embeddings

The different types of word embeddings can be broadly classified into two categories-

Frequency based Embedding

Prediction based Embedding

## Frequency based Embedding

There are generally three types of vectors that we encounter under this category.

Count Vector

TF-IDF Vector

Co-Occurrence Vector

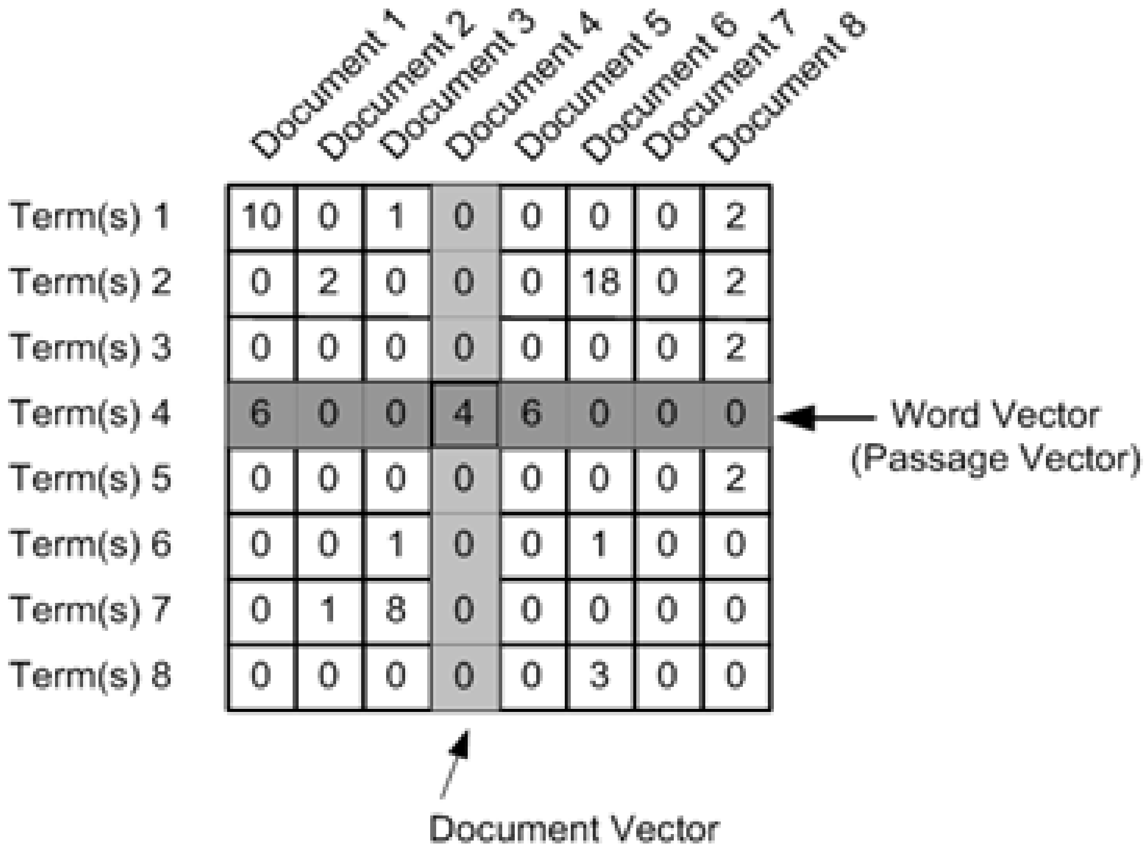
## Count Vector

D1: He is a lazy boy. She is also lazy.
D2: Neeraj is a lazy person.

The dictionary created may be a list of unique tokens(words) in the corpus =[‘He’,’She’,’lazy’,’boy’,’Neeraj’,’person’]
Here, D=2, N=6

| | He | She | lazy | boy | Neeraj | person |
|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 2 | 1 | 0 | 0 |
| D2 | 0 | 0 | 1 | 0 | 1 | 1 |

## Count Vector



| | Document 1 | Document 2 | Document 3 | Document 4 | Document 5 | Document 6 | Document 7 | Document 8 |
|---|---|---|---|---|---|---|---|---|
| Term(s) 1 | 10 | 0 | 1 | 0 | 0 | 0 | 0 | 2 |
| Term(s) 2 | 0 | 2 | 0 | 0 | 0 | 18 | 0 | 2 |
| Term(s) 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Term(s) 4 | 6 | 0 | 0 | 4 | 6 | 0 | 0 | 0 |
| Term(s) 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 |
| Term(s) 6 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Term(s) 7 | 0 | 1 | 8 | 0 | 0 | 0 | 0 | 0 |
| Term(s) 8 | 0 | 0 | 0 | 0 | 0 | 3 | 0 | 0 |

## What is tf-idf?

Term frequency - inverse document frequency

Allows you to determine the most important words in each document

Each corpus may have shared words beyond just stopwords

These words should be down-weighted in importance

Example from astronomy: "Sky"

Ensures most common words don't show up as key words

Keeps document specific frequent words weighted high

## Tf-Idf Formula

$$w_{ij} = tf_{i,j} * \log(\frac{N}{df_i})$$

$w_{ij}$ = tf-idf weight for token i in document j

$tf_{i,j}$ = number of occurences of token i in document j

$df_i$ = number of documents that contain token i

$N$ = total number of documents

## Co-Occurrence Vector

Co-occurrence – For a given corpus, the co-occurrence of a pair of words say w1 and w2 is the number of times they have appeared together in a Context Window.

Context Window – Context window is specified by a number and the direction.

# Co-Occurrence Vector

Corpus = He is not lazy. He is intelligent. He is smart.

| | He | is | not | lazy | intelligent | smart |
|---|---|---|---|---|---|---|
| **He** | 0 | 4 | 2 | 1 | 2 | 1 |
| **is** | 4 | 0 | 1 | 2 | 2 | 1 |
| **not** | 2 | 1 | 0 | 1 | 0 | 0 |
| **lazy** | 1 | 2 | 1 | 0 | 0 | 0 |
| **intelligent** | 2 | 2 | 0 | 0 | 0 | 0 |
| **smart** | 1 | 1 | 0 | 0 | 0 | 0 |

## Bag-of-words

Basic method for finding topics in a text

Need to first create tokens using tokenization

... and then count up all the tokens

The more frequent a word, the more important it might be

Can be a great way to determine the significant words in a text

## Text Classification

Text classification is one of the classical problem of NLP. Notorious examples include – Email Spam Identification, topic classification of news, sentiment classification and organization of web pages by search engines.

Text classification, in common words is defined as a technique to systematically classify a text object (document or sentence) in one of the fixed category. It is really helpful when the amount of data is too large, especially for organizing, information filtering, and storage purposes.

# Summary

This module covered the following topics:

Introduction to NLP

Applications of NLP

Tokenization

Stemming

Lemmatization

Count Vector

TF IDF

Co occurrence Matrix

Bag of Words

Text Classification