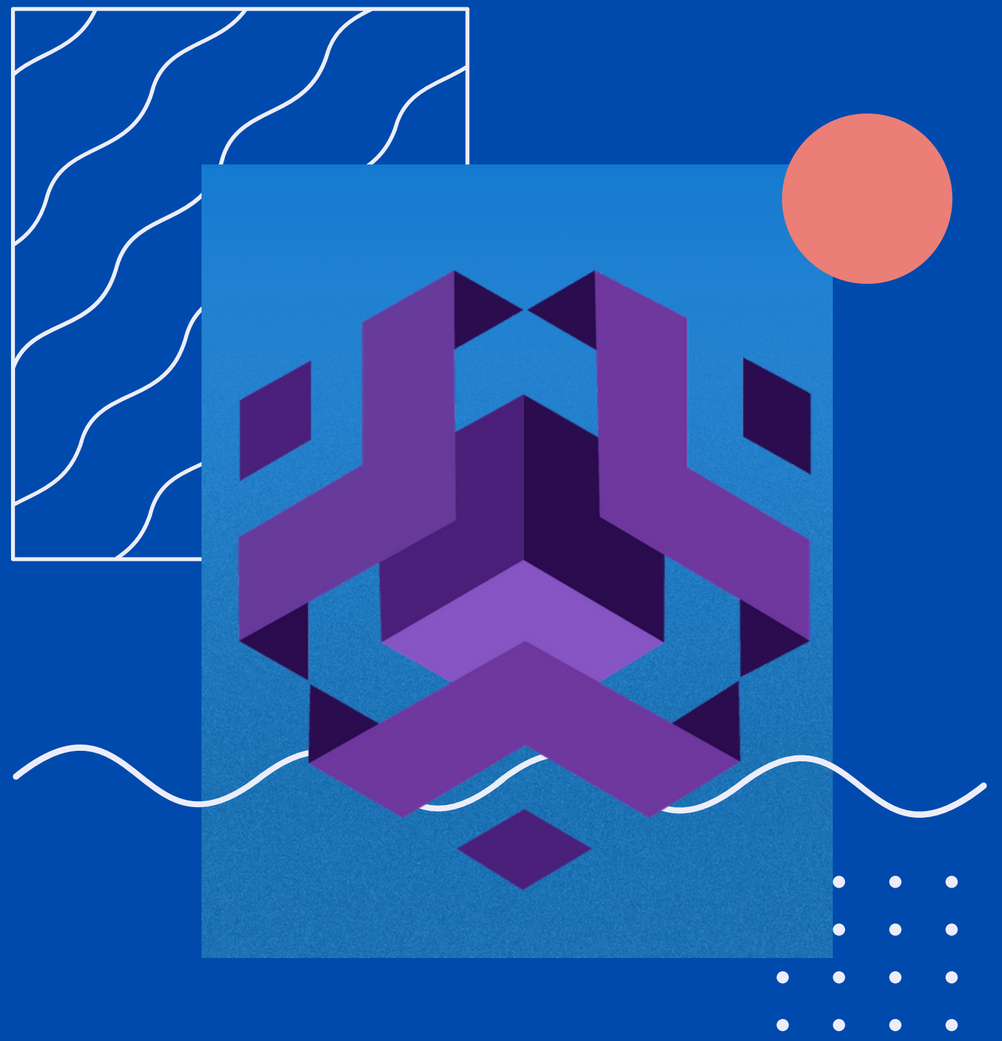
NLP Day 5

A WEBINAR BY
DATA SCIENCE COMMUNITY SRM

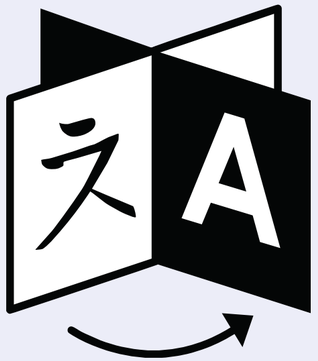


Your Speakers

ROHAN MATHUR

AKSHAT ANAND

Topics For This Session



WHAT IS NLP?

WHY NLP?

HOW NLP PIPELINES WORK

TEXT PROCESSING

COUNTING WORDS

FEATURE EXTRACTION

MODELLING

PART-OF-SPEECH TAGGING

NAMED ENTITY RECOGNITION

BAG OF WORDS

TF-IDF

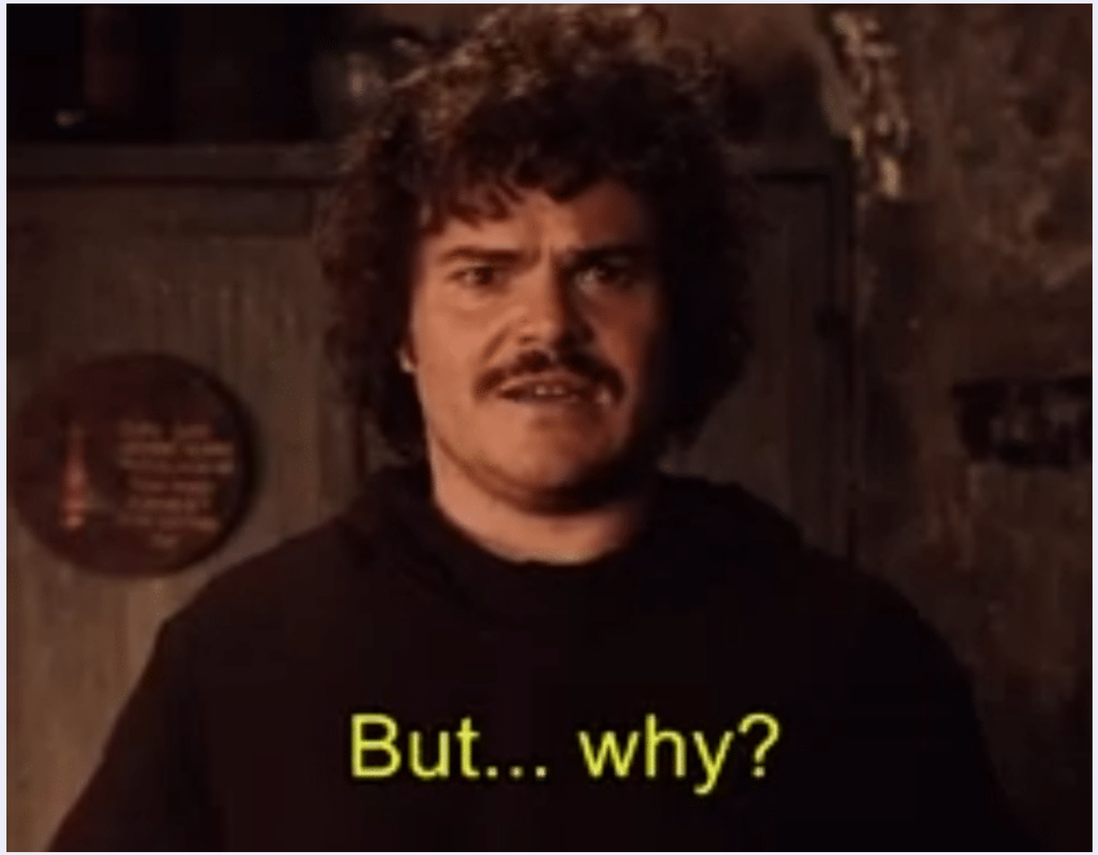
COSINE SIMILARITY

JACCARD SIMILARITY

SUMMARY

LET'S GET STARTED

Why NLP?



WHAT IS NLP?

Natural - Existing in or derived from nature; not made or caused by humankind.

Language - the principal method of human communication, consisting of words used in a structured and conventional way and conveyed by speech, writing, or gesture.

Processing - perform a series of mechanical or chemical operations on (something) in order to change or preserve it.

Voice Assistants

"WHAT TIME IS THE NEXT BUS FROM THE CITY AFTER THE 5:00 PM BUS ?"

"WHEN WILL COLLEGE REOPEN?"

"I AM A 3RD YEAR CSE STUDENT WHICH CLASSES DO I HAVE TODAY ?"

"WHO IS DONALD KNUTH ?"

INFORMATION EXTRACTION

The Raw Info

EXTRACTION OF MEANING FROM EMAIL :-

WE HAVE DECIDED TO MEET TOMORROW AT 10:00 AM IN THE LAB

Extracted Info

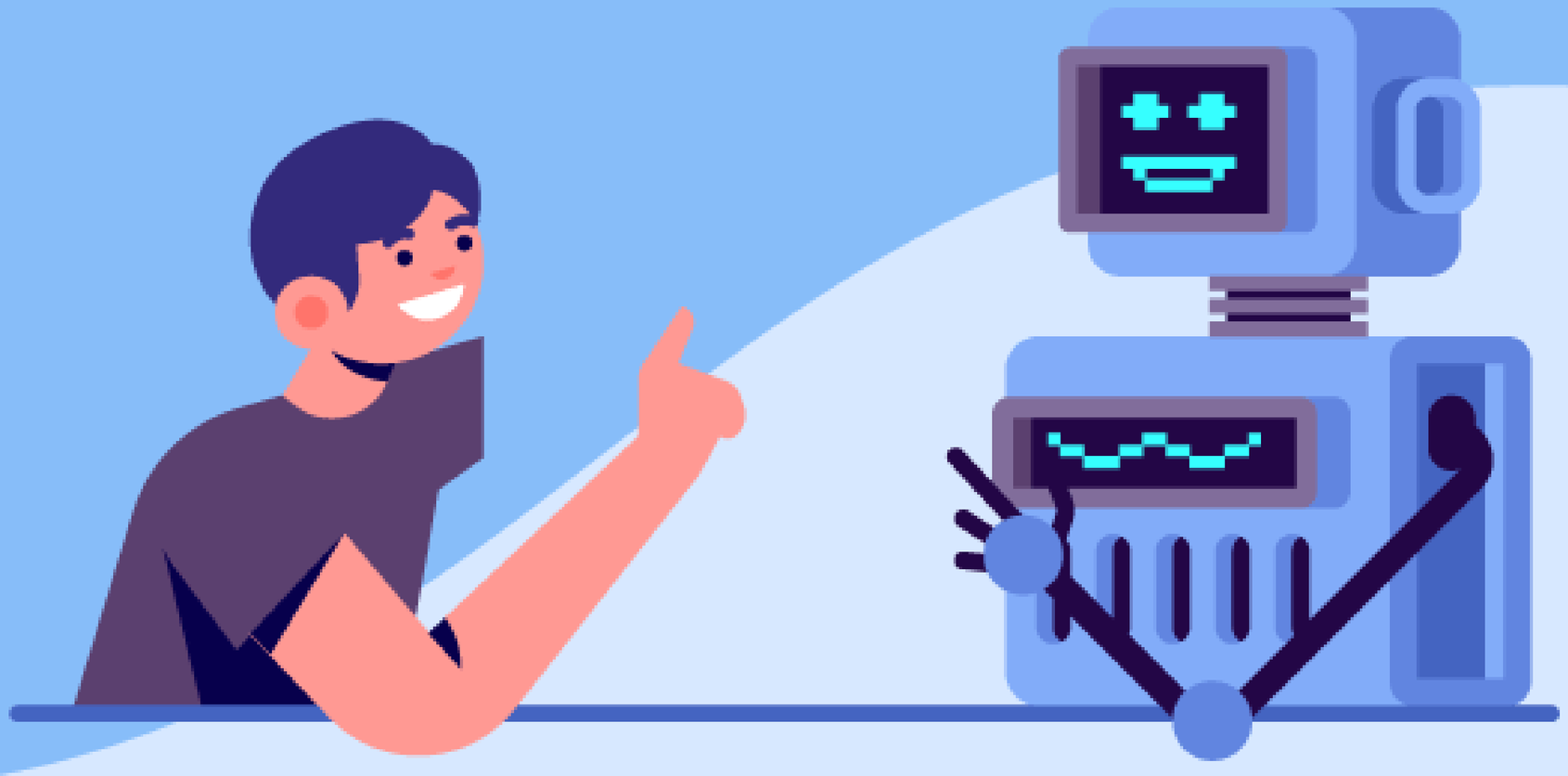
TO DO :

MEETING TIME : 10:00 AM, 22/3/2012

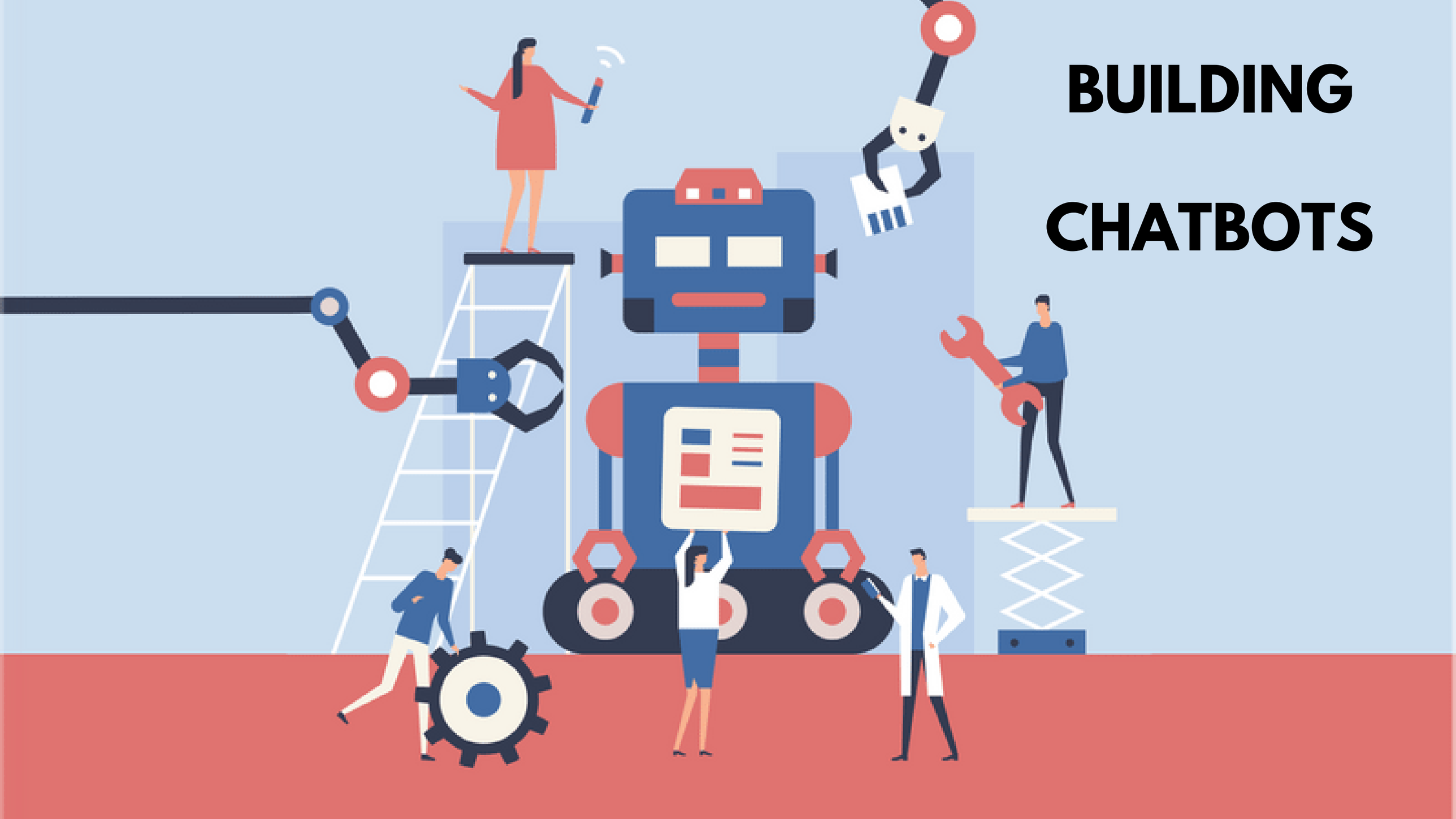
VENUE : LAB

SENTIMENT ANALYSIS

MACHINE TRANSLATION

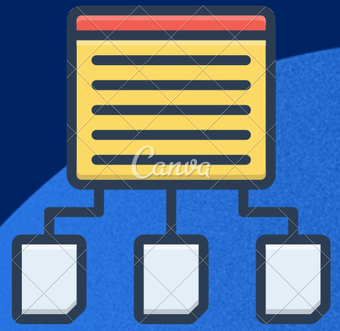


BUILDING CHATBOTS

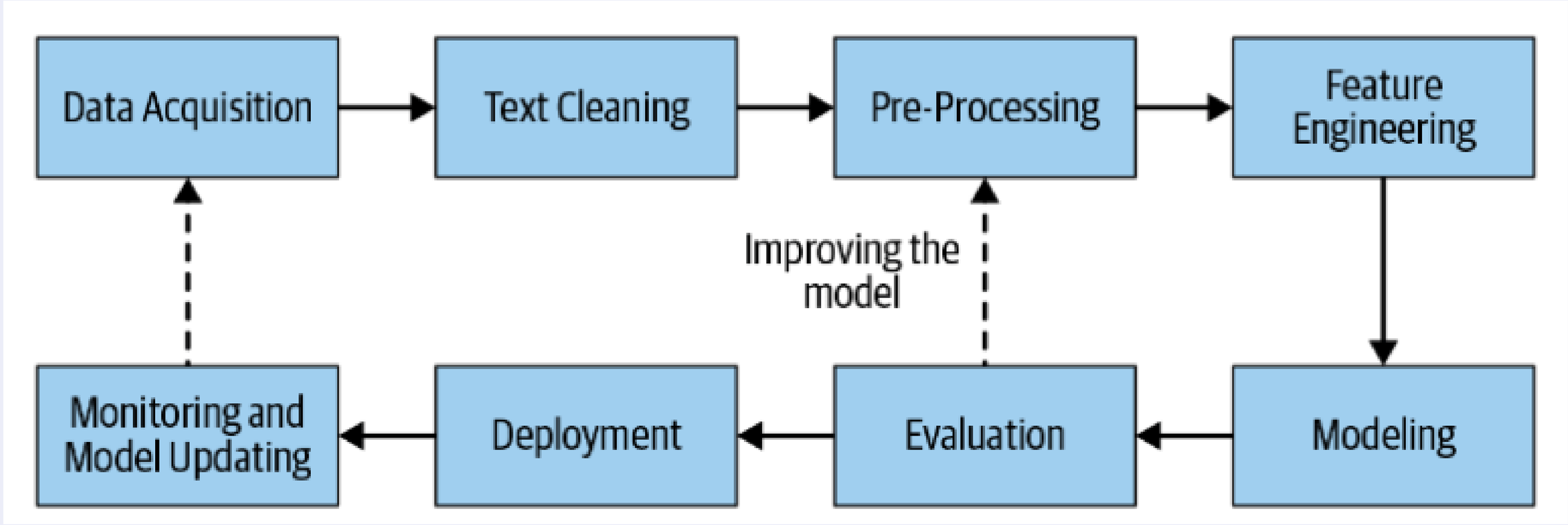


OTHER USES

- Voice Assistants
- Grammar Checkers
- Auto-Correct
- Text Summarization
- Context Analysis
- And Many More ...



NLP PIPELINES(AN OVERVIEW)



Text Preprocessing

TEXT PREPROCESSING

Why Preprocessing?

To preprocess your text simply means to bring your text into a form that is **predictable and analyzable** for your task.

Data preprocessing is an **essential step** in building a Machine Learning model and depending on **how well the data has been preprocessed**; the results are seen.

Steps for Preprocessing :-

- Tokenization
- Lower casing
- Stop words removal
- Stemming
- Lemmatization

STEPS FOR PREPROCESSING

- **Tokenization :** Splitting the sentence into words.

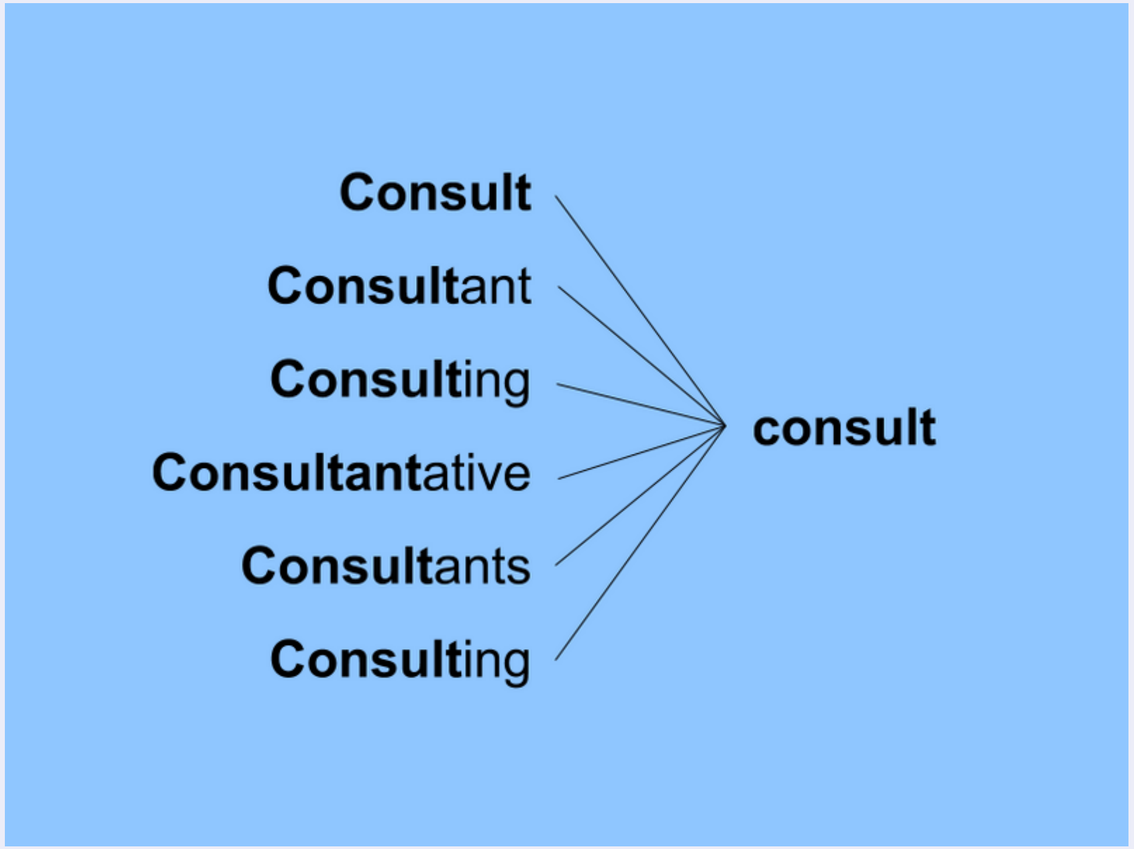
- **Lower Casing :** Converting a word to lowercase Eg. “RoHaN” ---> “rohan”
  Words like “Book” and “book” mean the same thing.
  But represented differently when making our models!

- **Stop Words Removal** : Stop words are very commonly used words (a, an, the, etc.) in the documents.
  These words do not really signify any importance as they do not help in distinguishing two documents.

STEPS FOR PREPROCESSING

- **Stemming :** It is a process of transforming a word to its root form. The stem sometimes does not make sense as it is not a word in English. This is a disadvantage of stemming. This is tackled by --

- **Lemmatization** : Unlike stemming, lemmatization reduces the words to a word existing in the language.

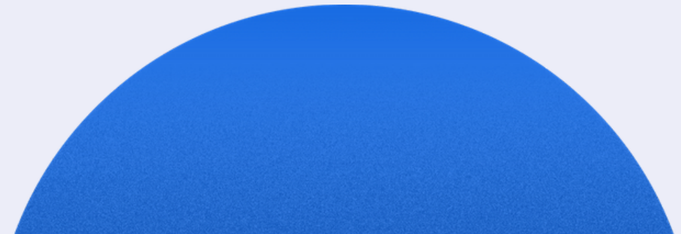


FEATURE EXTRACTION

Feature extraction step means to **extract and produce feature representations** that are appropriate for the type of NLP task you are trying to accomplish and the type of model you are planning to use.

But the main problem in working with language processing is that machine learning algorithms cannot work on the raw text directly. So, we need some feature extraction techniques to **convert text into a matrix (or vector)** of features.

Feature extraction **increases the accuracy** of learned models by extracting features from the input data.



Counting Words

By word frequency we indicate the **number of times each token** occurs in a text.

Let's implement a simple function that is often used in Natural Language Processing: **Counting Word Frequencies.**

As I was waiting, a man came out of a side room, and at a glance I was sure he must be Long John. His left leg was cut off close by the hip, and under the left shoulder he carried a crutch, which he managed with wonderful dexterity, hopping about upon it like a bird. He was very tall and strong, with a face as big as a ham—plain and pale, but intelligent and smiling. Indeed, he seemed in the most cheerful spirits, whistling as he moved about among the tables, with a merry word or a slap on the shoulder for the more favoured of his guests.

— Excerpt from Treasure Island, by Robert Louis Stevenson...

This **feature further** on can be extended to a statistical language model that is called -> **N - Gram Model**

MODELLING

NLP is all about ‘modelling’

A model is a file that has been trained to recognize certain types of patterns. You train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data.

Once you have trained the model, you can use it **to reason over data that it hasn't seen before**, and **make predictions** about those data.

There are pretrained models that can be used. Some are -

- **Google’s BERT -** With this, anyone in the world can train their own question answering models in a few hours on a single GPU.
- **Open AI GPT Series -** which stands for Generative Pre-trained Transformers, is an autoregressive language model that uses deep learning to produce human-like text.


MAYBE WE SHOULD
JUST TAKE A BREAK

MEANWHILE...

DSCOMMUNITY_SRM

DATA SCIENCE
COMMUNITY SRM

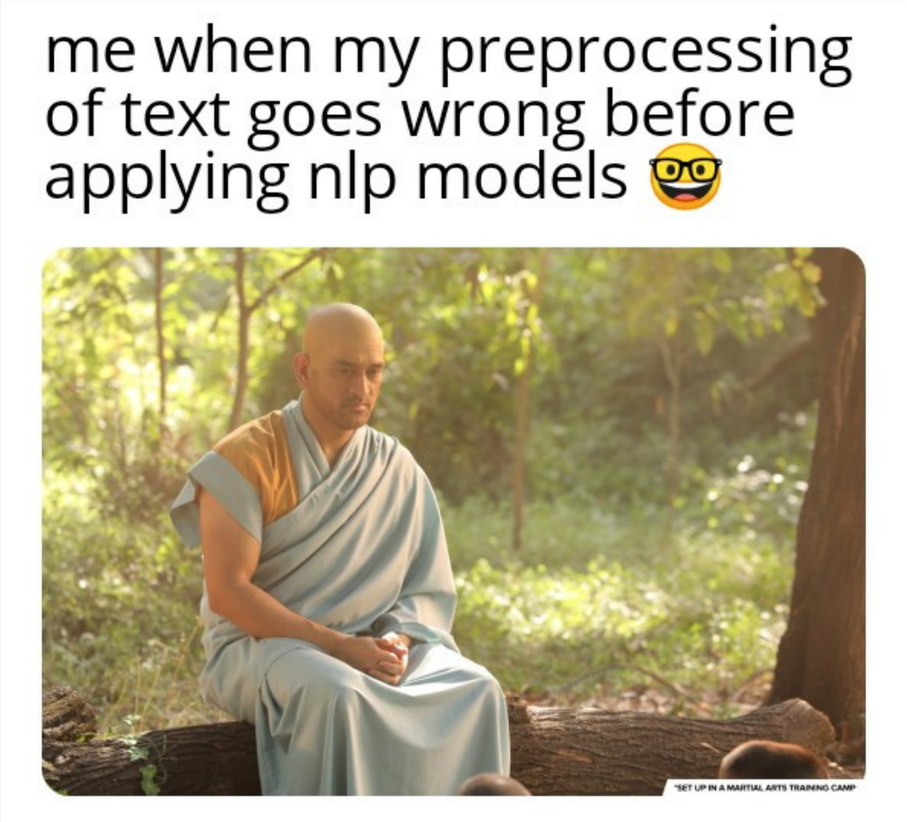
DSCOMMUNITY_SRM

DATASCIENCECOMMUNITY
@GMAIL.COM

DATA SCIENCE
COMMUNITY SRM


TIME
FOR
CHANGE

LET'S KEEP IT GOING



PART-OF-SPEECH TAGGING

Part-of-speech tagging using a **predefined grammar** like this is a simple, but the limited solution. It can be very tedious and error-prone for a large corpus of text since you have to account for all possible sentence structures and tags!

There are other more advanced forms of POS tagging that can **learn sentence structures and tags** from given data, including Hidden Markov Models (HMMs) and Recurrent Neural Networks (RNNs).

N-Gram

N-Grams is a set of 1 or more consecutive sequence of items that occur next to each other.
As an instance, if the sentence is "I am Akshat Anand, then:
1-Gram would be: -----> "I" , "am" , "Akshat" , "Anand"
2-Gram would be: -----> "I am" , "am Akshat" , "Akshat Anand"
3-Gram would be: -----> "I am Akshat" , "am Akshat Anand"

can you guess what would be for 4 - Gram?

One use case would be -> **Autocomplete feature** of the search engines

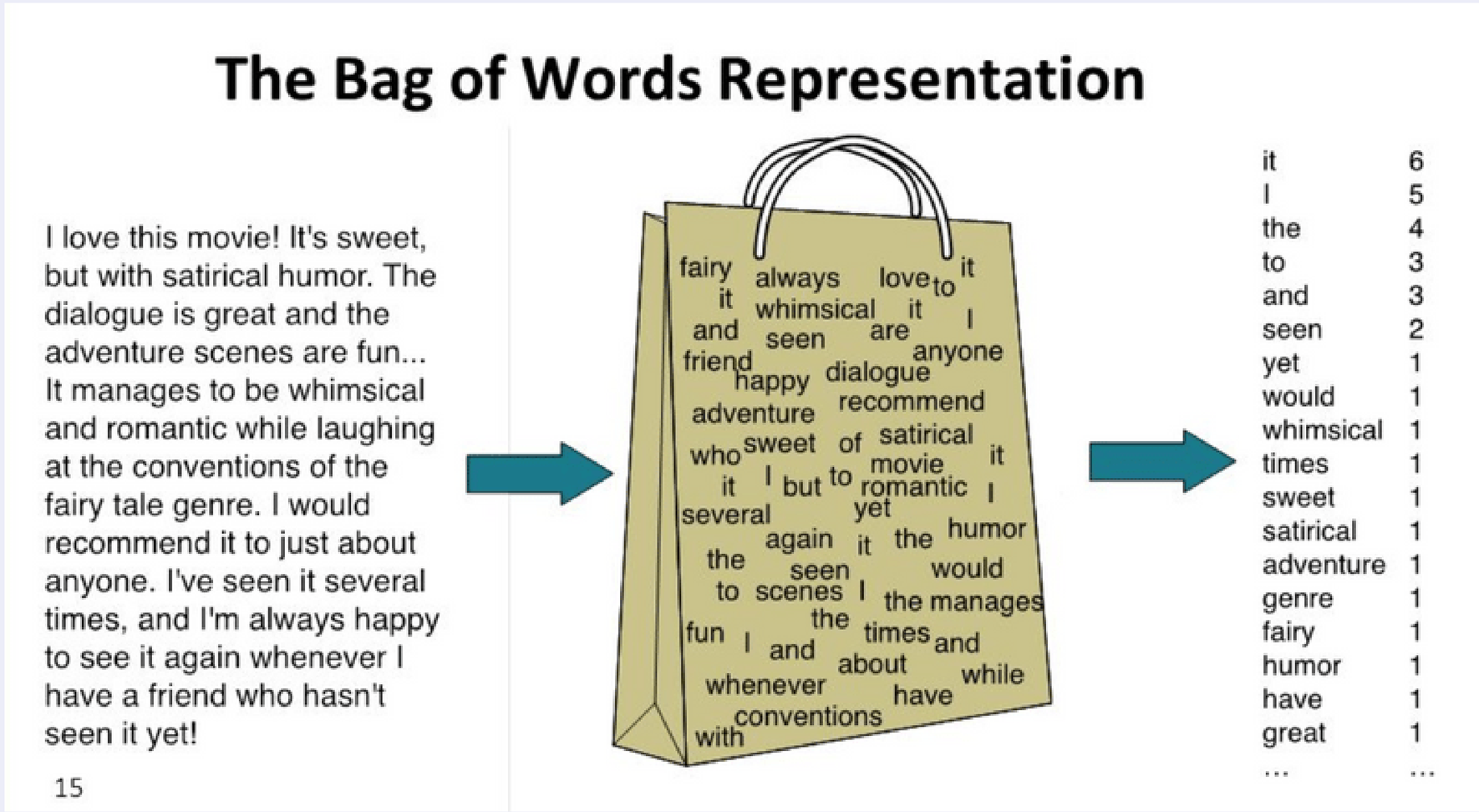
NAMED ENTITY RECOGNITION

It is probably the first step towards information extraction that seeks to locate and **classify named entities** in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. NER is used in many fields in Natural Language Processing (NLP), and it can help to **answer many real-world questions**, such as:

- Which companies were mentioned in the news article?
- Were specified products mentioned in complaints or reviews?
- Does the tweet contain the name of a person? Does the tweet contain this person's location?

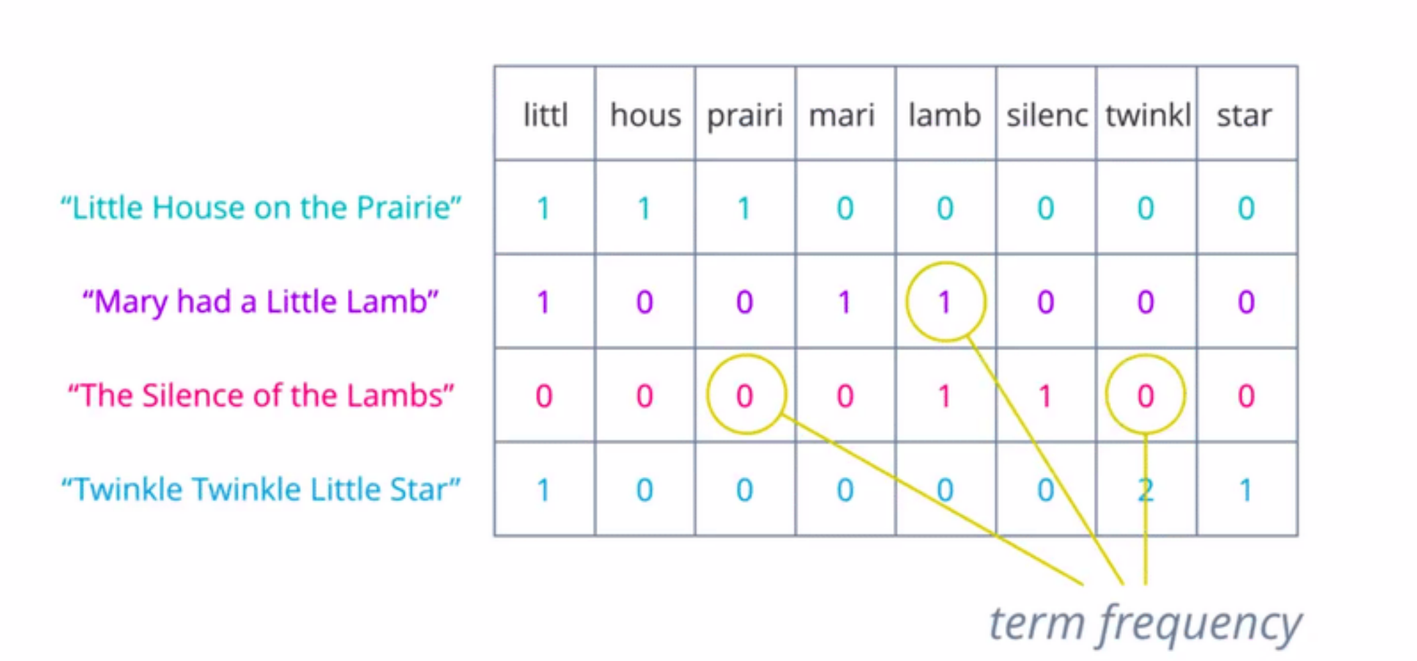
BAG OF WORDS (BOW)

This model which simply counts **how many times a word** appears in a document.



BAG OF WORDS (BOW)

It forms a document matrix of unique words with values ranging as 0 and 1, and the values in the matrix are known as term frequency.



BAG OF WORDS (BOW)

The demerit of using dot product over cosine similarity is that it calculates the portion of overlap it is not affected by values that are not in common a better way is to use cosine similarity for similarity it will be 1 and for dissimilarity it will be -1.

	littl	hous	prairi	mari	lamb	silenc	twinkl	star
a "Little House on the Prairie"	1	1	1	0	0	0	0	0
b "Mary had a Little Lamb"	1	0	0	1	1	0	0	0

$$\mathbf{a}\cdot\mathbf{b} = \sum a_0 b_0 + a_1 b_1 + \ldots + a_n b_n = 1$$ *dot product*

$$\cos(\theta) = \frac{\mathbf{a}\cdot\mathbf{b}}{\|\mathbf{a}\|\cdot\|\mathbf{b}\|} = \frac{1}{\sqrt{3}\times\sqrt{3}} = \frac{1}{3}$$ *cosine similarity*

TF-IDF (TERM FREQUENCY/ INVERSE DOCUMENT FREQUENCY)

TF-IDF is a **statistical measure** that evaluates how relevant a word is to a document in a collection of documents.

TF-IDF **weighs the importance of words** in a document. For example, “the” is commonly used in any documents so that TF-IDF does not consider “the” important to characterize documents.

TF-IDF (TERM FREQUENCY/ INVERSE DOCUMENT FREQUENCY)

The document frequency is divided by each value of the document term matrix which eventually helps in finding the unique words in the given sentence.

	littl	hous	prairi	mari	lamb	silenc	twinkl	star
“Little House on the Prairie”	1/3	1	1	0	0	0	0	0
“Mary had a Little Lamb”	1/3	0	0	1	1/2	0	0	0
“The Silence of the Lambs”	0	0	0	0	1/2	1	0	0
“Twinkle Twinkle Little Star”	1/3	0	0	0	0	0	2	1

Evaluation Metrics

COSINE SIMILARITY

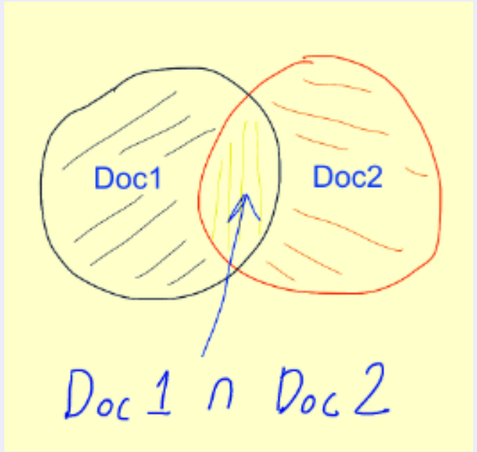
Cosine similarity is a useful measure if you want to consider duplicates when comparing the textual documents.
We can compute cosine angle between the two documents to estimate how similar the documents are.
The key to note is that the **smaller the angle, the bigger the cosine value** and the **more similar the two documents.**

The cosine similarity equation will result in a value **between 0 and 1** as the term frequencies are always positive.

$$\text{Cosine}(Doc1, Doc2) = \frac{\sum_{i=1}^{n} Doc1_i \times Doc2_i}{\sqrt{\sum_{i=1}^{n} Doc1_i^2} \times \sqrt{\sum_{i=1}^{n} Doc2_i^2}}$$

JACCARD SIMILARITY

Jaccard similarity is all about finding the **commonality via intersection** of the data sets. We can compute the Jaccard similarity coefficient score.
It is computed by **finding the intersection** between two sets and then dividing the size of intersection by the size of the union of the two sets.



- We can find the intersection of two documents using : doc1.intersection(doc2) as long as both are sets
- We can find the union of two documents by using union = doc1.union(doc2) as long as both are sets

Formula $\frac{\text{Len(Intersection)}}{\text{Len(Union)}}$

SUMMARIZE

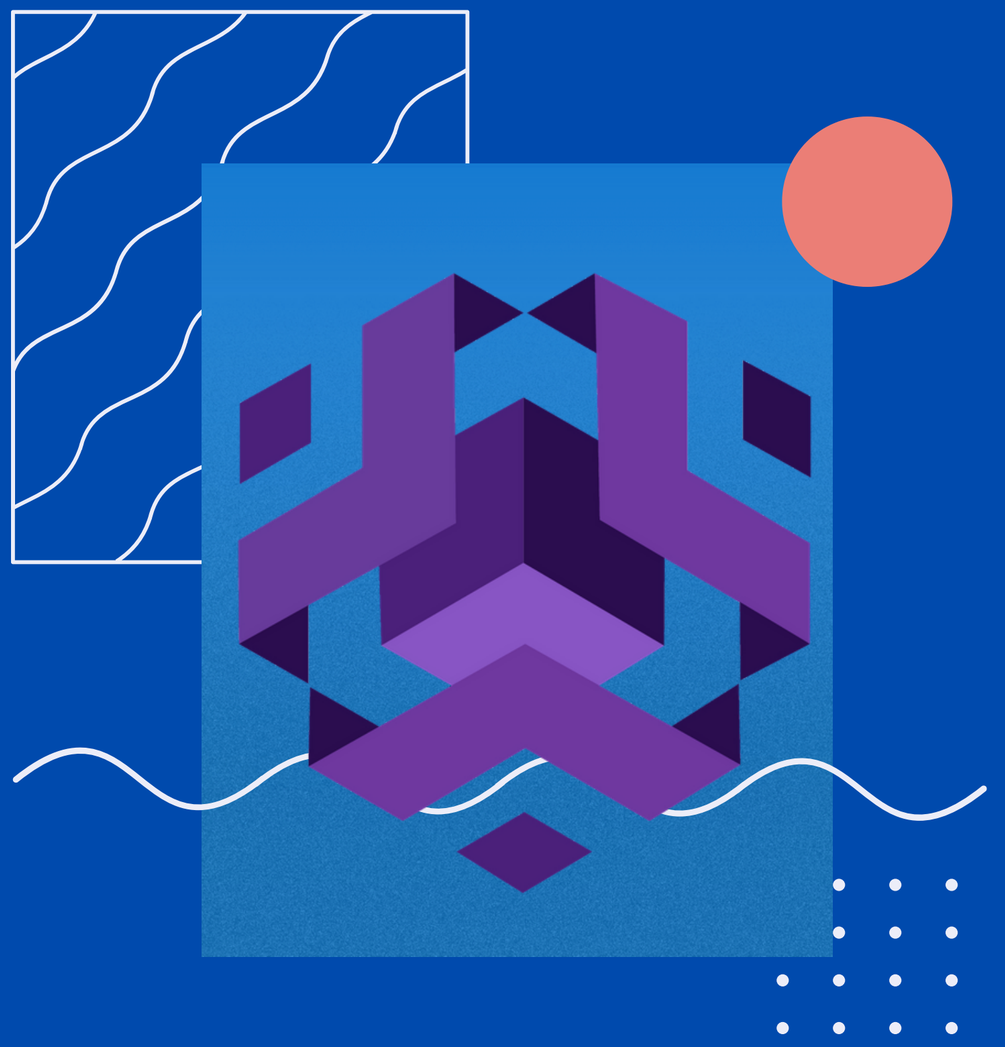
NLP Pipeline: Step-by-step

- Converting text to lowercase
- Removing Special Characters
- Stemming and Lemmatization
- Stop Words
- Bringing it all together — Building a Text Normalizer
- Modelling
- N gram
- PoS
- BoW
- TF-IDF
- Cosine and Jaccard similarity

Upcoming event

NLP Day 6

-TOMORROW



by

TARUSHI PATHAK

HARSH SHARMA

QUIZ

Thank you