Introduction to NLP

MUKESH KUMAR

AGENDA

- What is NLP?
- Tokenization
- Vectorization
- Text Processing
- Working with web Data
- Wrod2Vec Embeddings
- POS tagging

Computer understand?

Humans communicate?

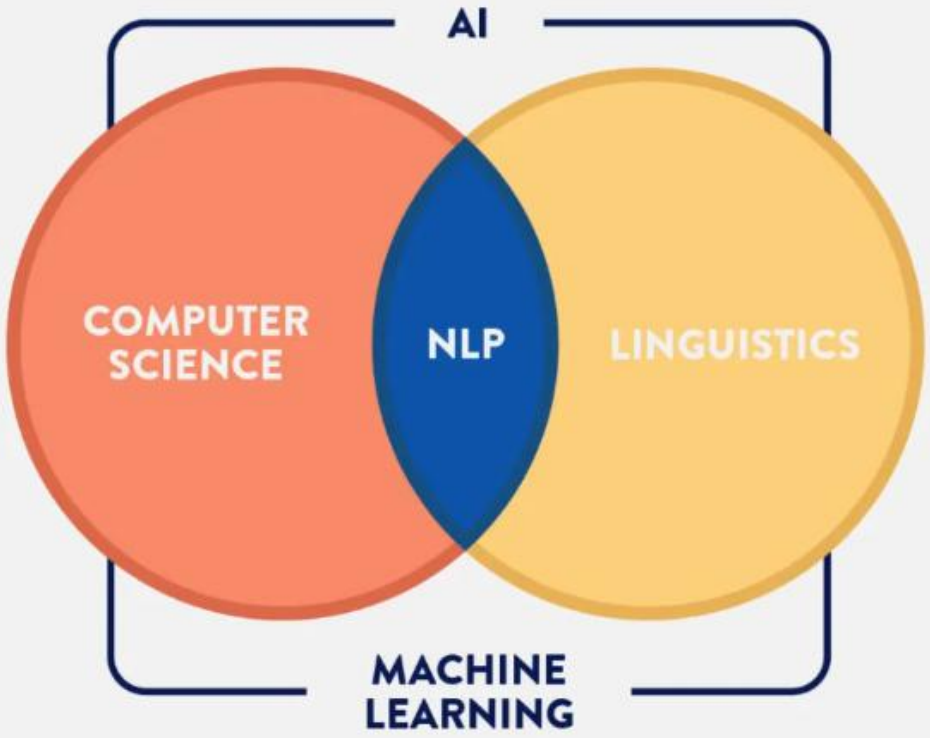
- Goal of NLP is to give machine ability to read , listen and understand language like humans do

NLP

- **Natural Language Processing (NLP)** is a subfield of artificial intelligence that focuses on the interaction between computers and humans through natural language

WHAT IS NATURAL LANGUAGE PROCESSING?

NLP is the ability for computers to understand human language. NLP is an interdisciplinary field of computer science and linguistics



NLP CHALLENGES

- I ate an apple
- I have an apple watch

- He approached many banks for loan
- Delhi is situated on the banks of yamuna river

Hard to build rules for languages

So we build ML models

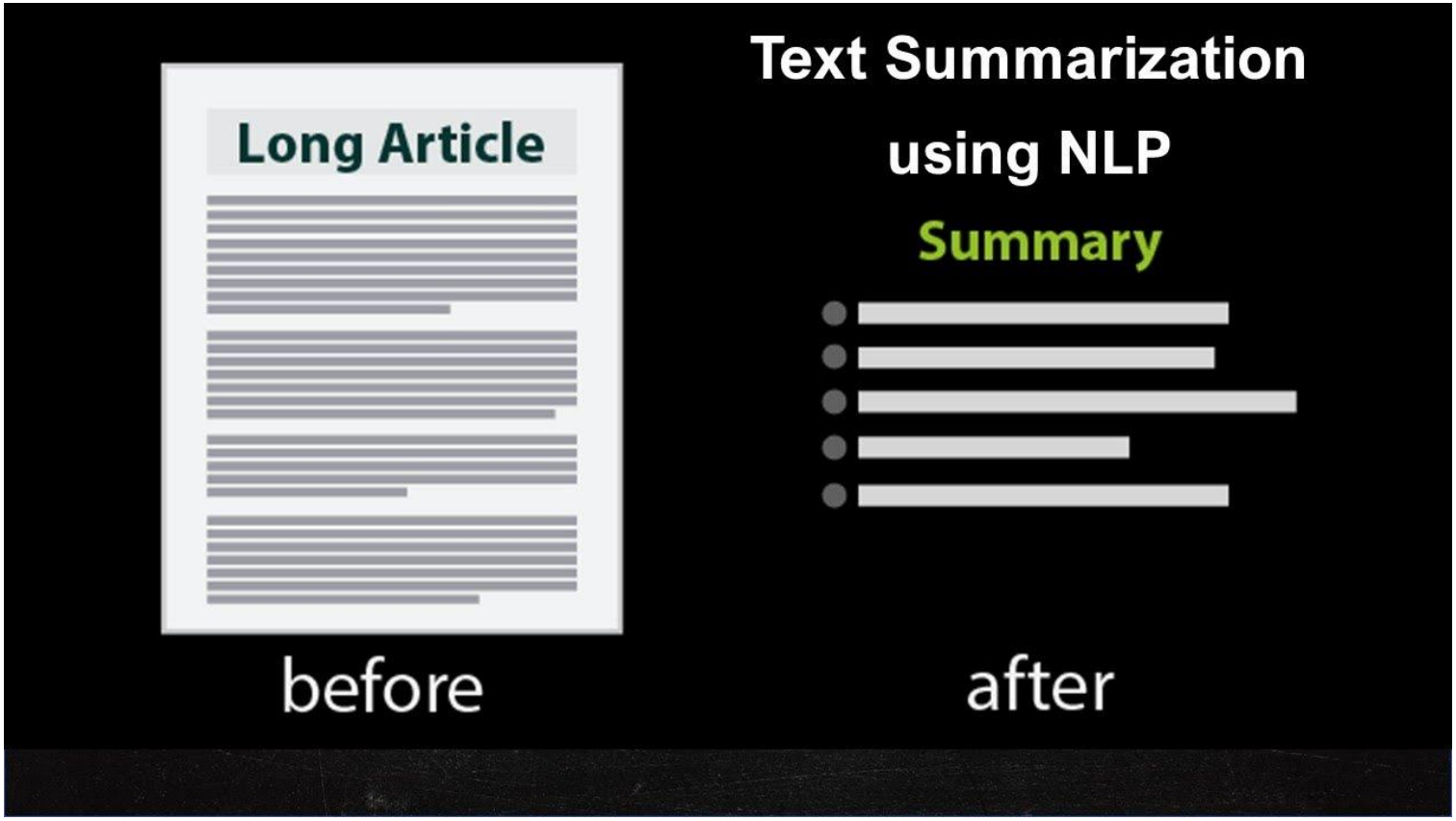
NLP APPLICATIONS

Digital Assistants

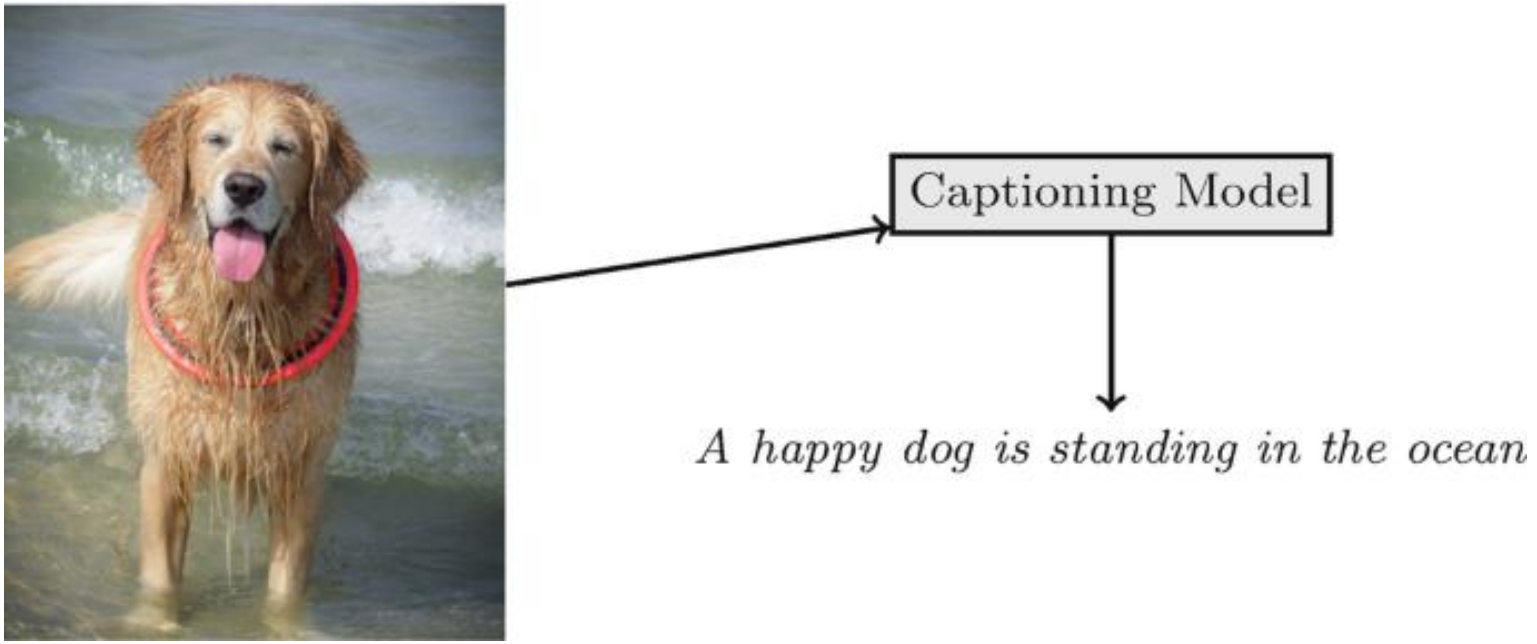
Grammarly


grammarly
Your the one who sho
SPELLING
Your
You're
You are
It appears that the
ssessive

Text Summarization



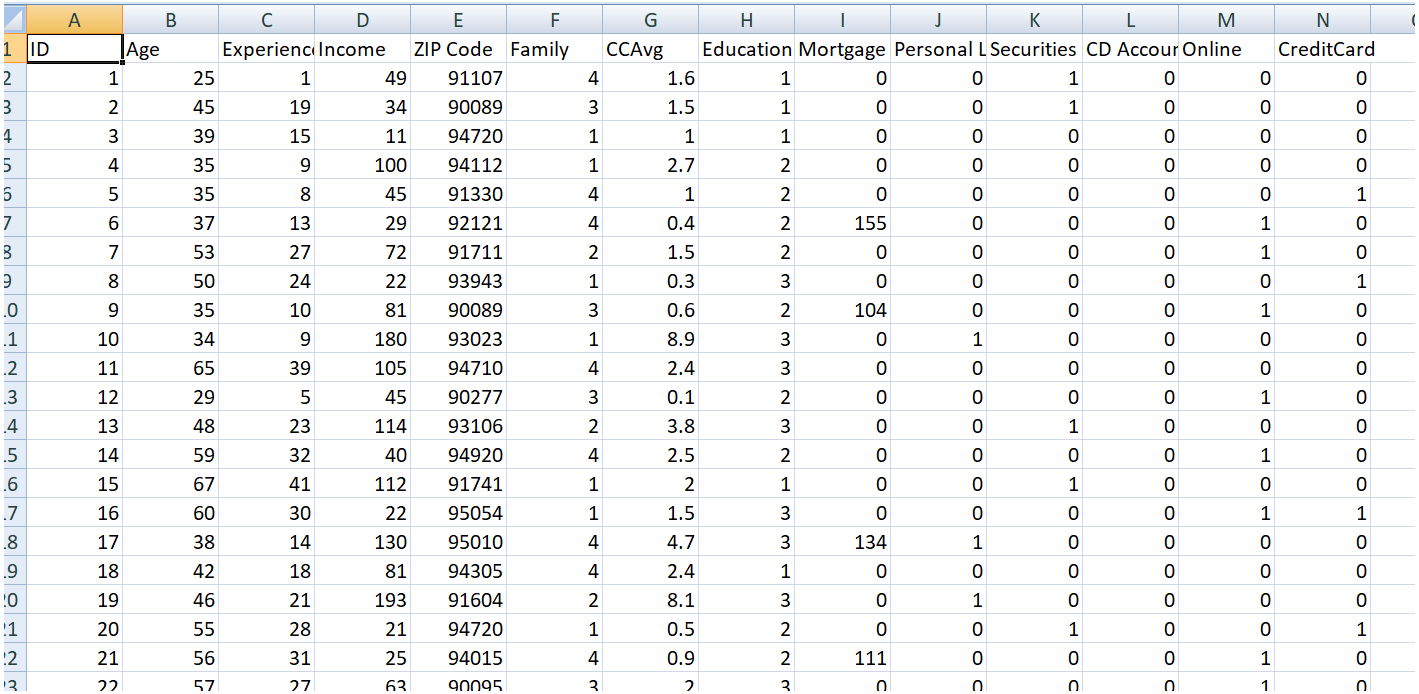
Describe a picture



BUILDING NLP MODELS

Structured Data

- Features
- Ordered

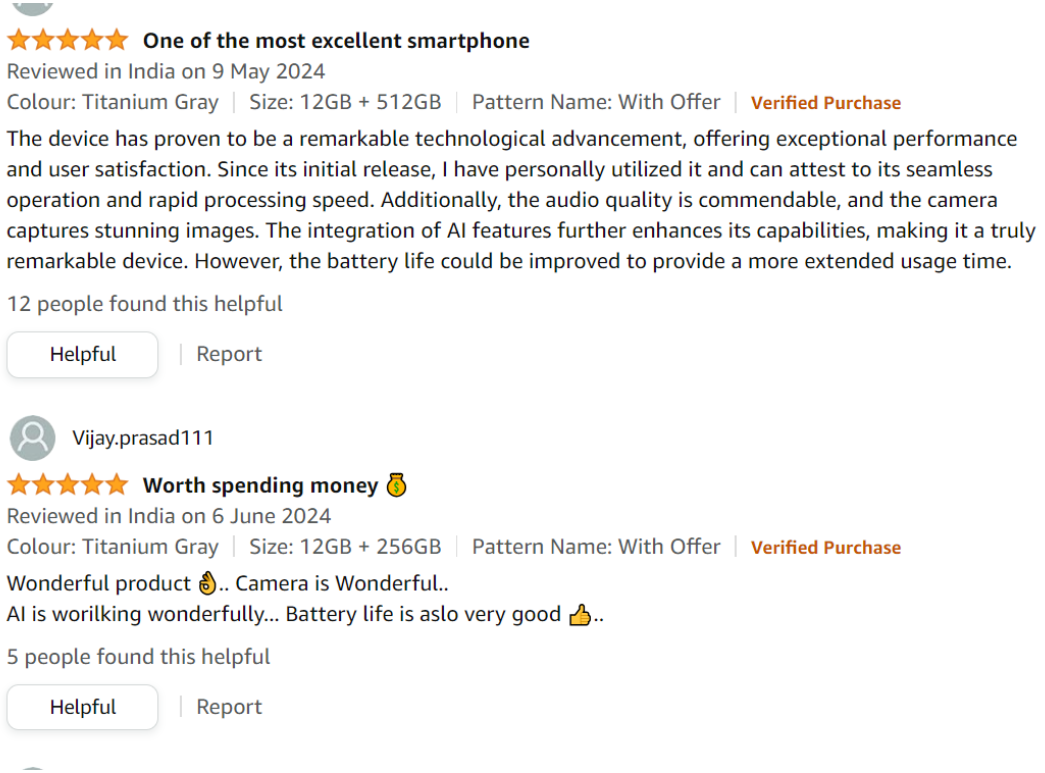


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ID	Age	Experienc	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal L	Securities	CD Accour	Online	CreditCard
2	1	25	1	49	91107	4	1.6	1	0	0	1	0	0	0
3	2	45	19	34	90089	3	1.5	1	0	0	1	0	0	0
4	3	39	15	11	94720	1	1	1	0	0	0	0	0	0
5	4	35	9	100	94112	1	2.7	2	0	0	0	0	0	0
6	5	35	8	45	91330	4	1	2	0	0	0	0	0	1
7	6	37	13	29	92121	4	0.4	2	155	0	0	0	1	0
8	7	53	27	72	91711	2	1.5	2	0	0	0	0	1	0
9	8	50	24	22	93943	1	0.3	3	0	0	0	0	0	1
10	9	35	10	81	90089	3	0.6	2	104	0	0	0	1	0
11	10	34	9	180	93023	1	8.9	3	0	1	0	0	0	0
12	11	65	39	105	94710	4	2.4	3	0	0	0	0	0	0
13	12	29	5	45	90277	3	0.1	2	0	0	0	0	1	0
14	13	48	23	114	93106	2	3.8	3	0	0	1	0	0	0
15	14	59	32	40	94920	4	2.5	2	0	0	0	0	1	0
16	15	67	41	112	91741	1	2	1	0	0	1	0	0	0
17	16	60	30	22	95054	1	1.5	3	0	0	0	0	1	1
18	17	38	14	130	95010	4	4.7	3	134	1	0	0	0	0
19	18	42	18	81	94305	4	2.4	1	0	0	0	0	0	0
20	19	46	21	193	91604	2	8.1	3	0	1	0	0	0	0
21	20	55	28	21	94720	1	0.5	2	0	0	1	0	0	1
22	21	56	31	25	94015	4	0.9	2	111	0	0	0	1	0
23	22	57	27	63	90095	3	2	3	0	0	0	0	1	0

Textual data

- What are the features?
- Doesn’t have features like

Structured data



★★★★★ **One of the most excellent smartphone**

Reviewed in India on 9 May 2024

Colour: Titanium Gray | Size: 12GB + 512GB | Pattern Name: With Offer | **Verified Purchase**

The device has proven to be a remarkable technological advancement, offering exceptional performance and user satisfaction. Since its initial release, I have personally utilized it and can attest to its seamless operation and rapid processing speed. Additionally, the audio quality is commendable, and the camera captures stunning images. The integration of AI features further enhances its capabilities, making it a truly remarkable device. However, the battery life could be improved to provide a more extended usage time.

12 people found this helpful

Helpful | Report

Vijay.prasad111

★★★★★ **Worth spending money** 💰

Reviewed in India on 6 June 2024

Colour: Titanium Gray | Size: 12GB + 256GB | Pattern Name: With Offer | **Verified Purchase**

Wonderful product 👌.. Camera is Wonderful..
AI is worilking wonderfully... Battery life is aslo very good 👍..

5 people found this helpful

Helpful | Report

Features for textual data

- Words
- Characters
- Combination of words(n-grams)

Document #1

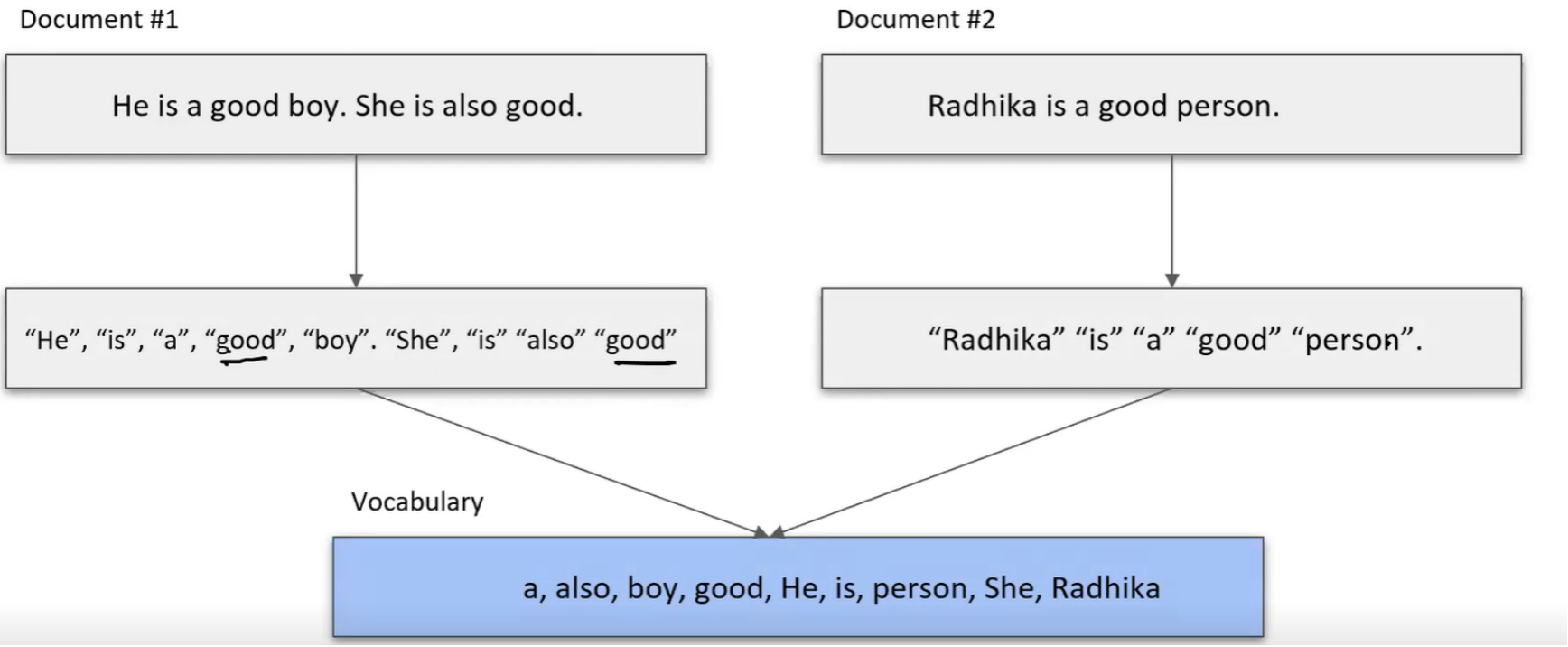
He is a good boy. She is also good.

Document #2

Radhika is a good person.

- Features are:
- Breakdown the text into individual tokens(words, sentences, group of words)
- Find the unique tokens
- Unique tokens form the vocabulary or dataset

- This is called tokenization


Document #1
He is a good boy. She is also good.
Document #2
Radhika is a good person.
“He”, “is”, “a”, “good”, “boy”. “She”, “is” “also” “good”
“Radhika” “is” “a” “good” “person”.
Vocabulary
a, also, boy, good, He, is, person, She, Radhika

What is Tokenization

- Tokenization is a process of breaking a text into smaller parts or chunks. Whether it’s breaking paragraphs into sentences, sentence into words or word into characters.

- Creating Vocabulary is the ultimate goal of Tokenization.

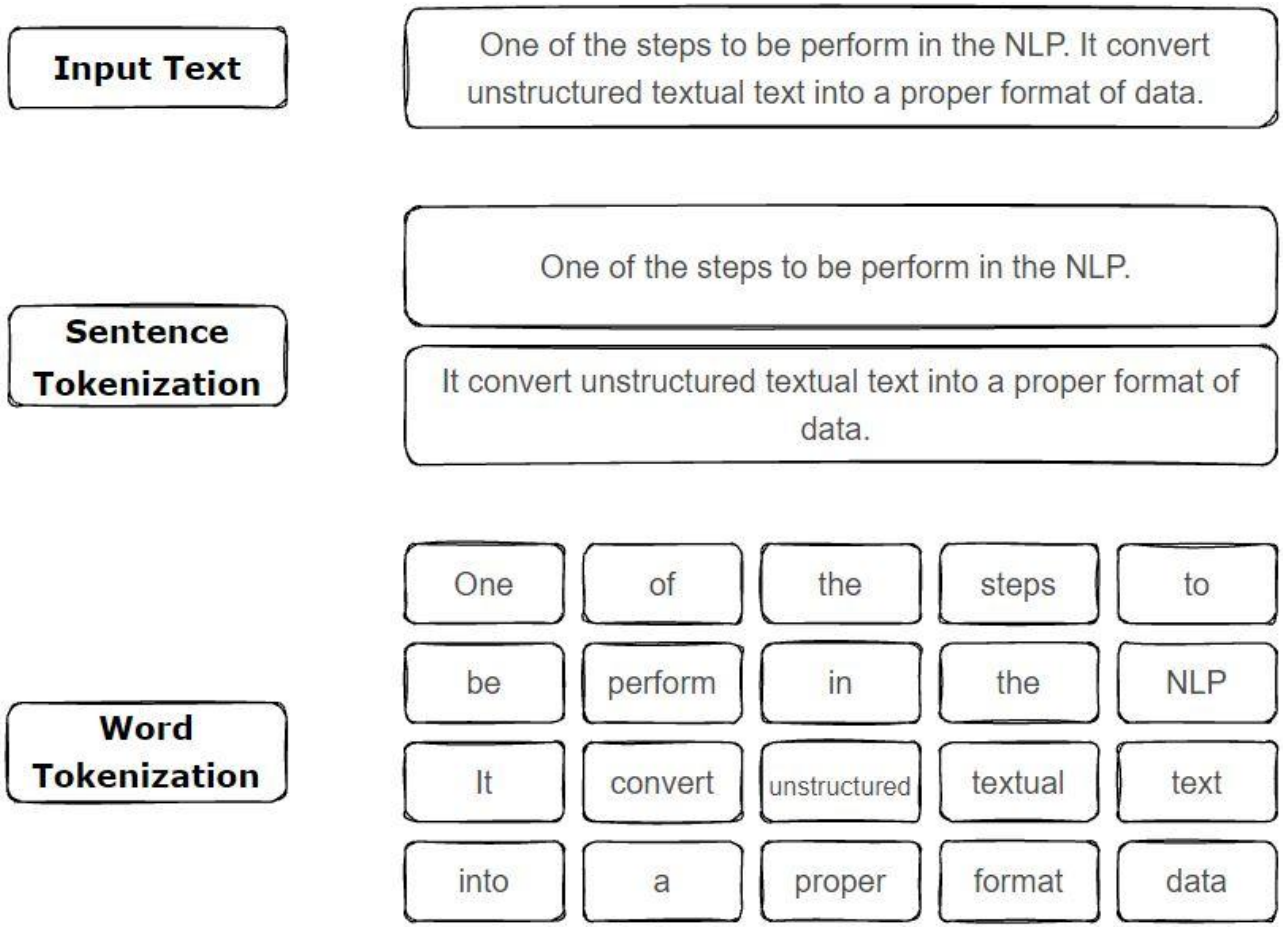
Types of Tokenization

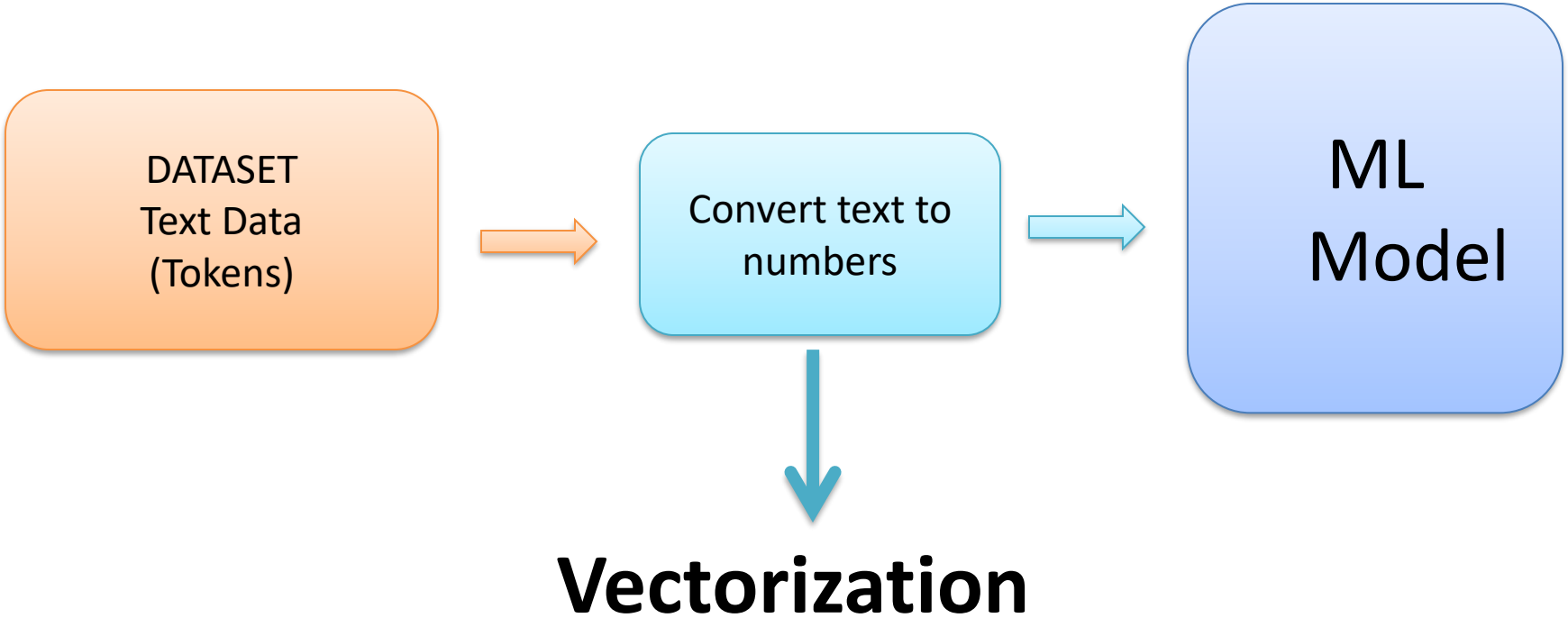
- Therefore, if you split the text data (or document) into words, it's called **Word Tokenization**.
- If the document is split into sentences, then it is called **Sentence Tokenization**.
- Similarly, splitting the document into individual characters is known as **Character Tokenization**.

- There are numerous uses of tokenization. We can use this tokenized form to:

- Count the number of words in the text

- Count the frequency of the word, that is, the number of times a particular word is present


Input Text
One of the steps to be perform in the NLP. It convert unstructured textual text into a proper format of data.
Sentence Tokenization
One of the steps to be perform in the NLP.
It convert unstructured textual text into a proper format of data.
Word Tokenization
One
of
the
steps
to
be
perform
in
the
NLP
It
convert
unstructured
textual
text
into
a
proper
format
data


DATASET
Text Data
(Tokens)
Convert text to numbers
ML Model
Vectorization

Vocabulary

a, also, boy, good, He, is, person, She, Radhika

	a	also	boy	good	He	Is	person	She	Radhika
Index	0	1	2	3	4	5	6	7	8

Assign index for each word in vocabulary

- Order, alphabetical , frequency based

Convert text to numbers

- Bag Of Words
- Count Vectorizer
- TF-IDF Vector

Bag of words

- A simple feature extraction approach in NLP
- Ignores grammar/Structure
- Represents each document by measuring presence of vocabulary words

Document #1

He is a good boy. She is also good.

Document #2

Radhika is a good person.

Vocabulary

a, also, boy, good, He, is, person, She, Radhika

	a	also	boy	good	He	Is	person	She	Radhika
Index	0	1	2	3	4	5	6	7	8
Document# 1				.					

Check if each word in Vocabulary appears in Document #1

Final dataset

	a	also	boy	good	He	Is	person	She	Radhika
Index	0	1	2	3	4	5	6	7	8
Document #1	1	1	1	1	1	1	0	1	0
Document #2	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**	**1**

Frequency of words is also noted

COUNT VECTORIZER

Document #1

He is a good boy. She is also good.

Document #2

Radhika is a good person.

Vocabulary

a, also, boy, good, He, is, person, She, Radhika

	a	also	boy	good	He	Is	person	She	Radhika
Index	0	1	2	3	4	5	6	7	8
Document #1	1	1	1	2	1	2	0	1	0

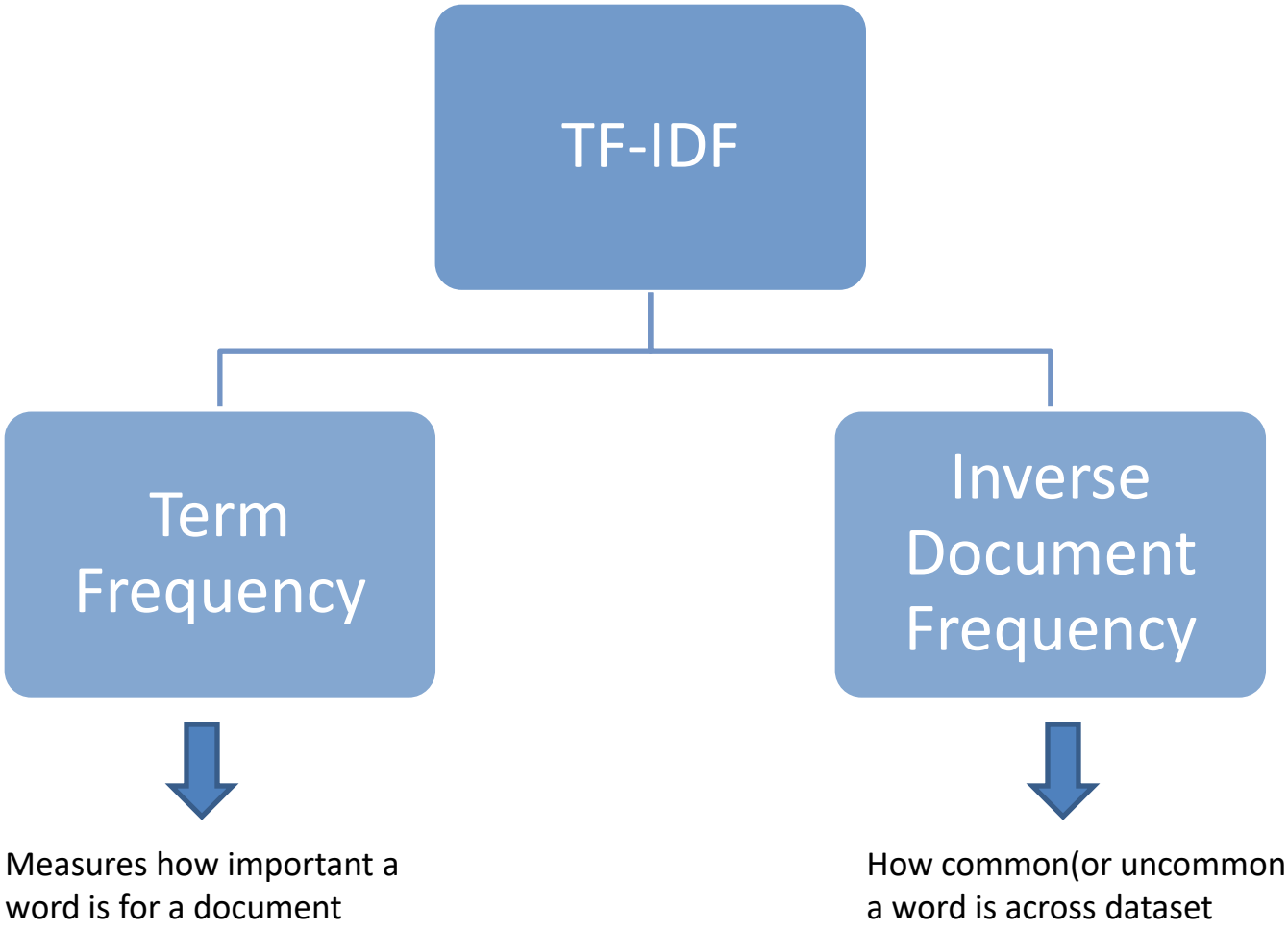
Document #1 = [1, 1, 1, 2, 1, 2, 0, 1, 0]

	a	also	boy	good	He	Is	person	She	Radhika
Index	0	1	2	3	4	5	6	7	8
Document #1	1	1	1	2	1	2	0	1	0
Document #2	1	0	0	1	0	1	1	0	1

- Document Term Matrix (DTM)

	a	also	boy	good	He	Is	person	She	Radhika
Index	0	1	2	3	4	5	6	7	8
Document #1	1	1	1	2	1	2	0	1	0
Document #2	1	0	0	1	0	1	1	0	1

TF-IDF VECTORIZER


TF-IDF
Term Frequency
Inverse Document Frequency
Measures how important a word is for a document
How common(or uncommon) a word is across dataset

- IDF will be low for words that are very common across dataset
- IDF will be very high for words that are rare and specific to documents
- TF will be high for repeated words in a doc

TF Calculations

- TF calculates the importance of a word inside a document without looking at other documents

Document #1

He is a good boy. She is also good.

He	1
is	2
a	1
good	2
boy	1
she	1
also	1
Total	**9**

$$TF = \frac{Frequency\ of\ the\ word\ in\ a\ Doc}{Total\ number\ of\ words\ in\ the\ Doc}$$

TF(He, doc#1) = 1/9 = 0.11

TF(good, doc#1) = 2/9 = 0.22

Document #2

Radhika is a good person.

Radhika	1
is	1
a	1
good	1
person	1
Total	**5**

$$TF = \frac{Frequency\ of\ the\ word\ in\ a\ Doc}{Total\ number\ of\ words\ in\ the\ Doc}$$

TF(He, doc#2) = 0/5 = 0

TF(good, doc#2) = 1/5 = 0.2

TF captures how important a word is to the document (without looking at other documents in the dataset)

IDF Calculations

- IDF tells us if a word(feature) can be used to distinguish documents.

- If a word appears in majority of documents then IDF will be close to zero i.e give low weightage to that feature.

Document #1

He is a good boy. She is also good.

He	1
is	2
a	1
good	2
boy	1
she	1
also	1
Total	**9**

Document #2

Radhika is a good person.

Radhika	1
is	1
a	1
good	1
person	1
Total	**5**

$$IDF = log(\frac{Num\ of\ Docs}{Word\ in\ Num\ of\ Docs})$$

$$IDF(He) = \log(2/1) = 0.301$$

$$IDF(good) = \log(2/2) = 0$$

- Multiply TF and IDF to find the final value of the feature

TF-IDF(He, doc#1) = 0.11 * 0.301 = 0.03311

TF-IDF(good, doc#1) = 0.22 * 0 = 0

TF-IDF(He, doc#2) = 0 * 0.301 = 0

TF-IDF(good, doc#2) = 0.2 * 0 = 0

IDF tells us if a word (feature) can be used to distinguish documents. If a word appears in the majority of the documents then IDF will be close to '0' i.e. give low weightage to that feature.

He is a good boy. She is also good.

Radhika is a good person.

Vocabulary

a, also, boy, good, He, is, person, She, Radhika

	a	also	boy	good	He	Is	person	She	Radhika
Index	0	1	2	3	4	5	6	7	8
Document #1				0	0.03311				
Document #2				0	0				

TF-IDF Vector

TEXT PREPROCESSING

HTML tags

I am going to Bangalore.

Unnecessary chars

Well this was fun! What do you think? 123#@!

Emojis

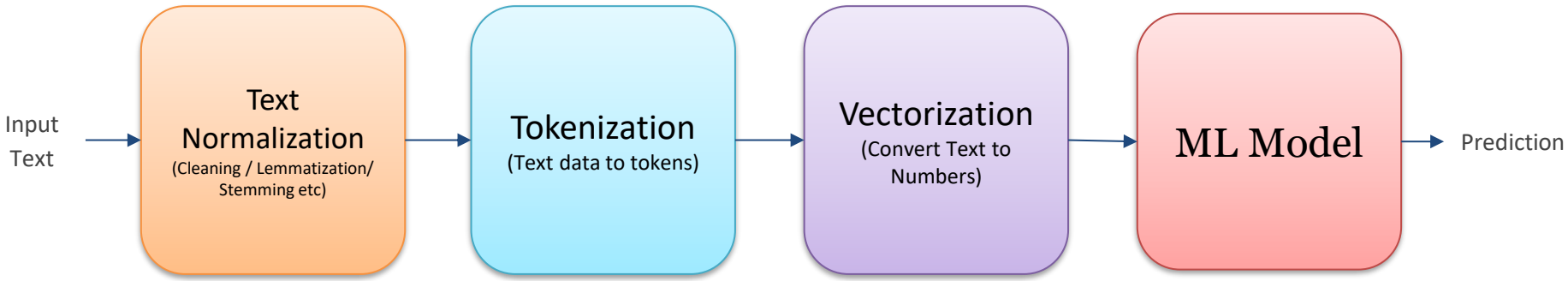
He is 😳

Accented Text

Sómě Áccěntěd tĕxt

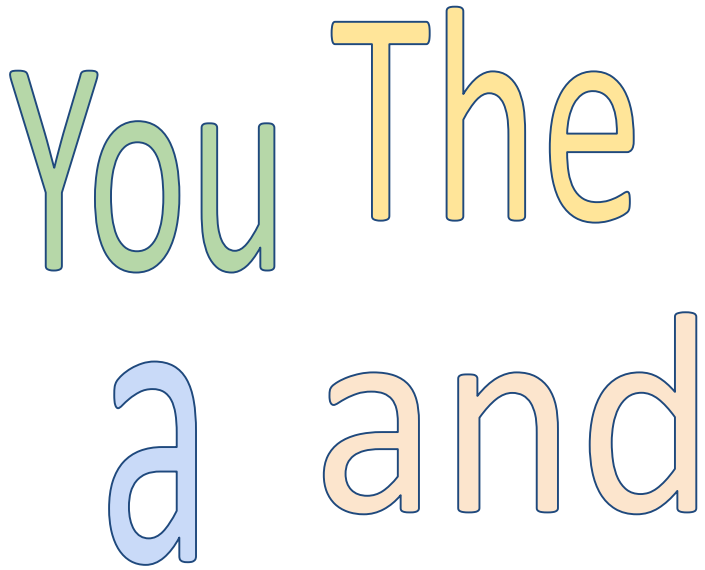
Real world data can be very messy

...



NLP Pipeline

TEXT PROCESSING TECHNIQUES



1. High frequency words *i.e* present in most documents
2. Can not be used to distinguish between documents
3. Can be **removed** as features

Stopwords

- Stopwords are commonly used words in a language that are often filtered out before processing text in natural language processing (NLP) and text mining tasks.

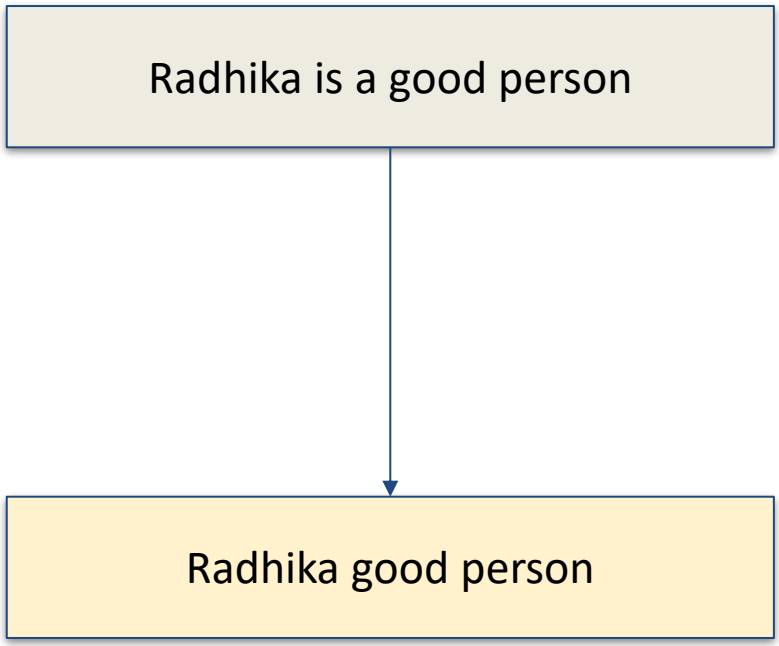
- These words are considered to have little meaningful content and are usually removed to improve the efficiency and effectiveness of text analysis.

- Examples of stopwords include words like "the", "is", "in", "and", "to", and "a".

Examples of Stopwords

- Here are some examples of stopwords in English:

- a, an, and, are, as, at, be, by, for, from, has, he, in, is, it, its, of, on, that, the, to, was, were, will, with



Stopwords Removal

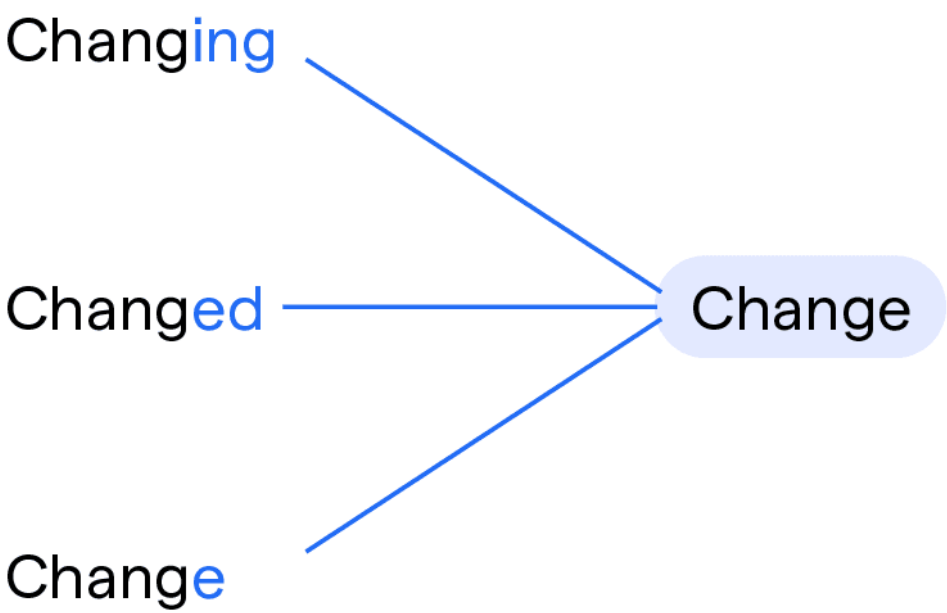
LEMMATIZATION

- Lemmatization is a text pre-processing technique used in natural language processing (NLP) models to break a word down to its root meaning to identify similarities.

- Considers the context of the word.

- Requires a dictionary (lexicon) to look up the base form.

Lemmatization



Lemmatization Examples

- The words "running", "ran", and "runs" are all forms of the word "run". Lemmatization would convert all these words to "run".

- Similarly, "better" and "best" would be converted to "good" if contextually appropriate.

Why Lemmatization is Important

- **Text Normalization:** Reduces word forms to a common base form, aiding tasks like indexing and search.
- **Improves NLP Models:** Reduces the number of unique words, enhancing model performance by lowering feature space dimensionality.
- **Context-Aware:** Lemmatization is more precise than stemming, ensuring words are transformed correctly based on context.

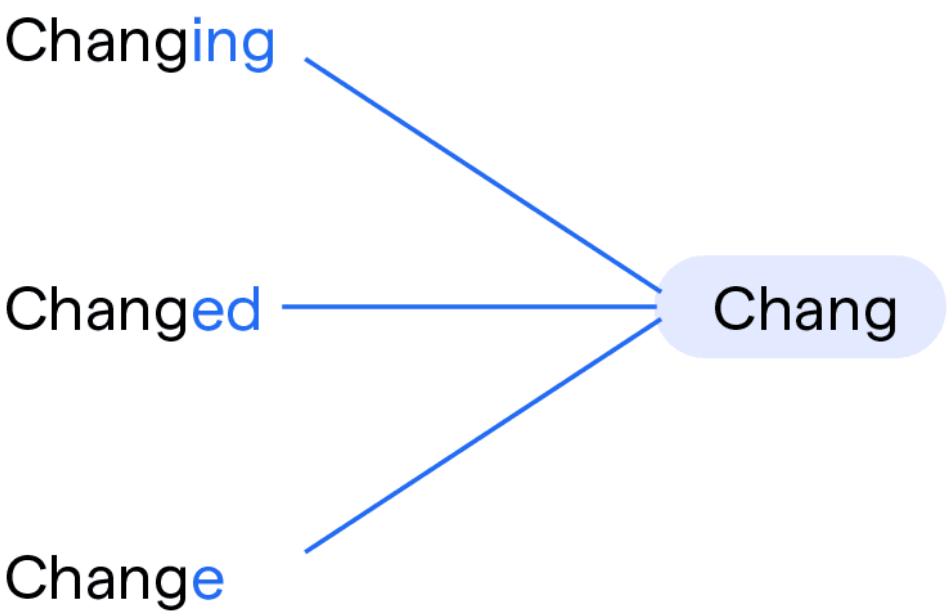
STEMMING

Stemming

- Stemming is a text processing technique used in natural language processing (NLP) to reduce words to their base or root form, called the "stem."

- The process involves removing suffixes and prefixes from words to transform them into their simplest form, often without considering the context or part of speech.

Stemming



Why Stemming is Important

- **Text Normalization**: Standardizes text by reducing word variations to a base form.
- **Search and Retrieval**: Enhances search results by matching word forms to the same base, improving recall.
- **Feature Reduction**: Reduces the number of unique terms, lowering dimensionality in machine learning models

Examples of Stemming

- For instance, the words:
- "running"
- "runner"
- "ran"
- All would be reduced to the stem "run".

Stemming vs Lemmatization

Differences Between Stemming and Lemmatization

- **Stemming**:
- Uses heuristic rules to chop off prefixes and suffixes.
- Fast and less computationally intensive.
- May produce non-dictionary words (e.g., "easili" from "easily").
- **Lemmatization**:
- Uses vocabulary and morphological analysis of words.
- Slower and more computationally intensive.
- Produces meaningful base forms (e.g., "better" to "good", "easily" to "easy").