TEXT PRE-PROCESSING

The three-step text mining process

Test Pre-processing

TEXT PRE-PROCESSING

The three-step text mining process

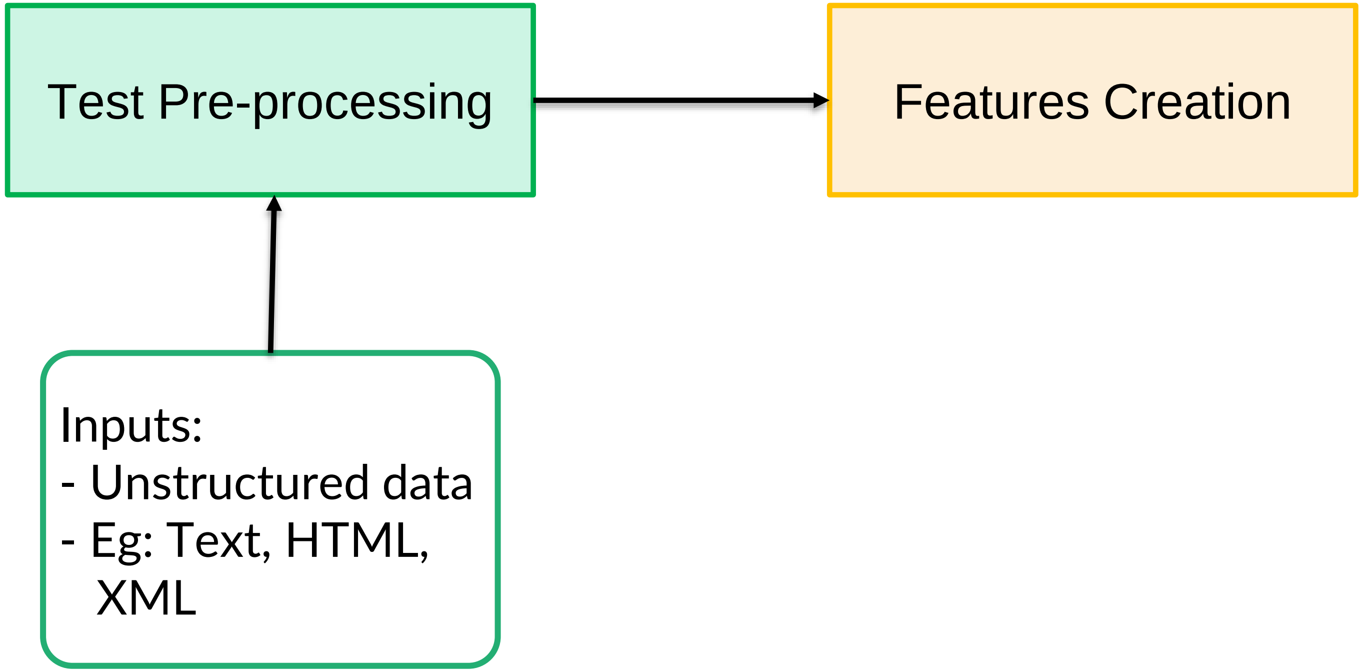
Test Pre-processing

Inputs:
- Unstructured data
- Eg: Text, HTML, XML

Remove No's
punctuations
foreign characters

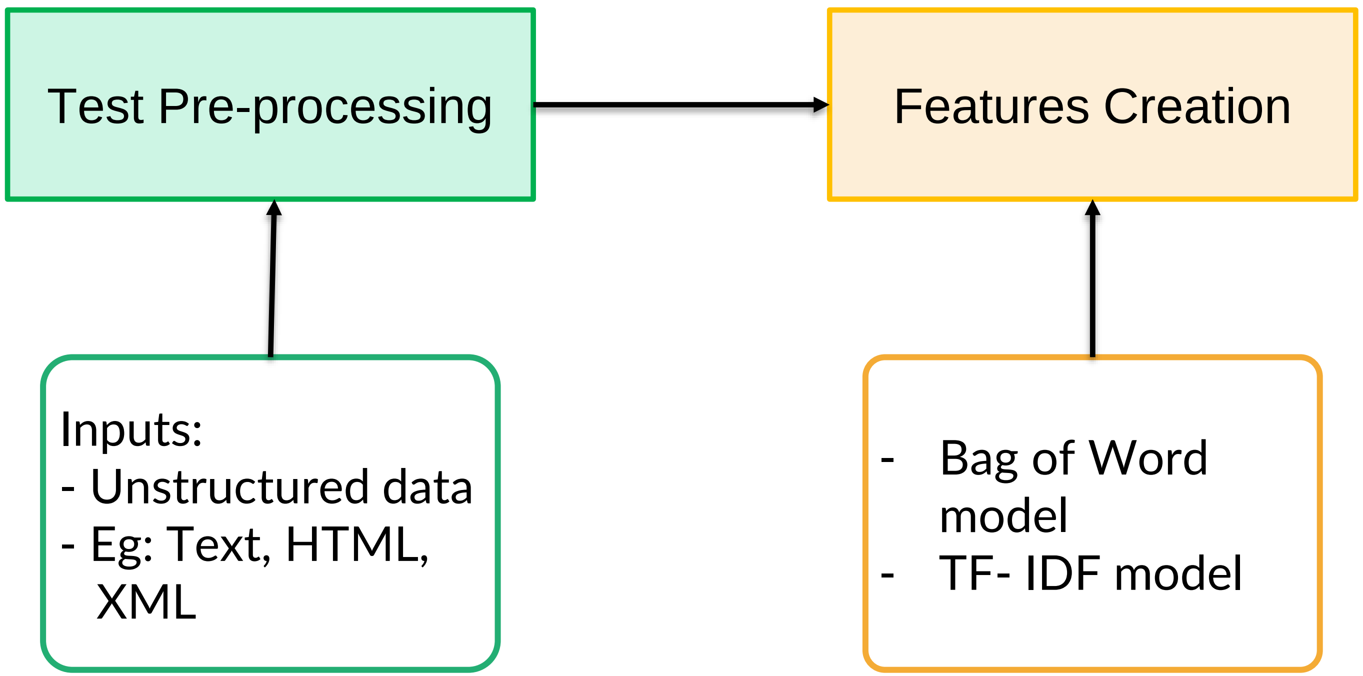
TEXT PRE-PROCESSING

The three-step text mining process



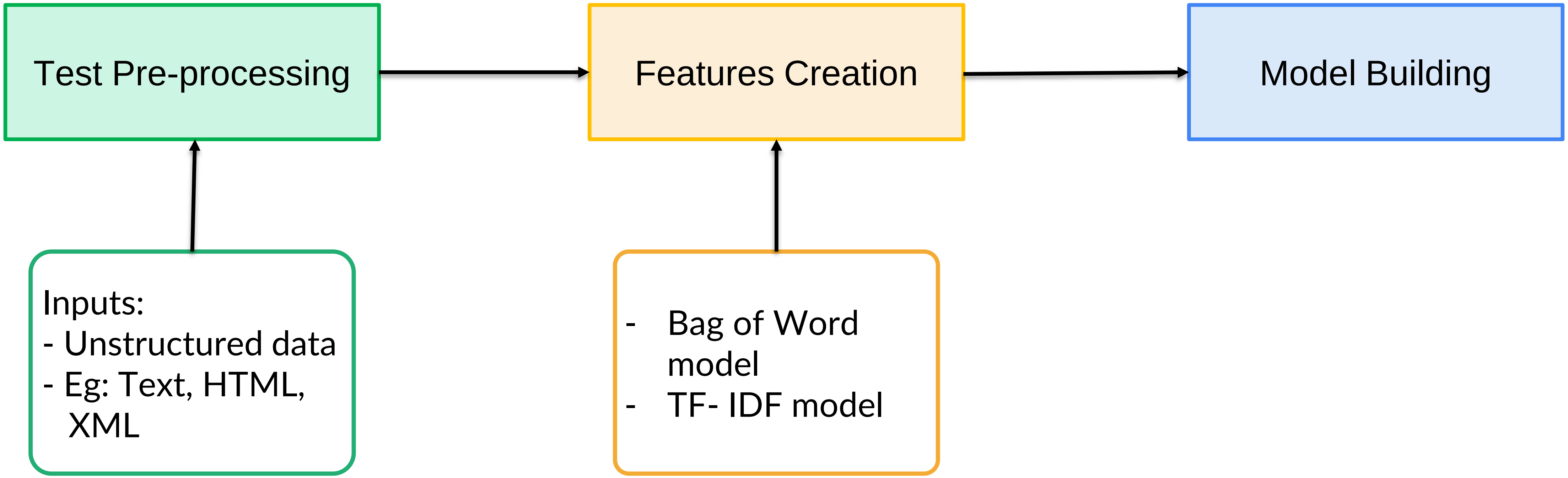
TEXT PRE-PROCESSING

The three-step text mining process



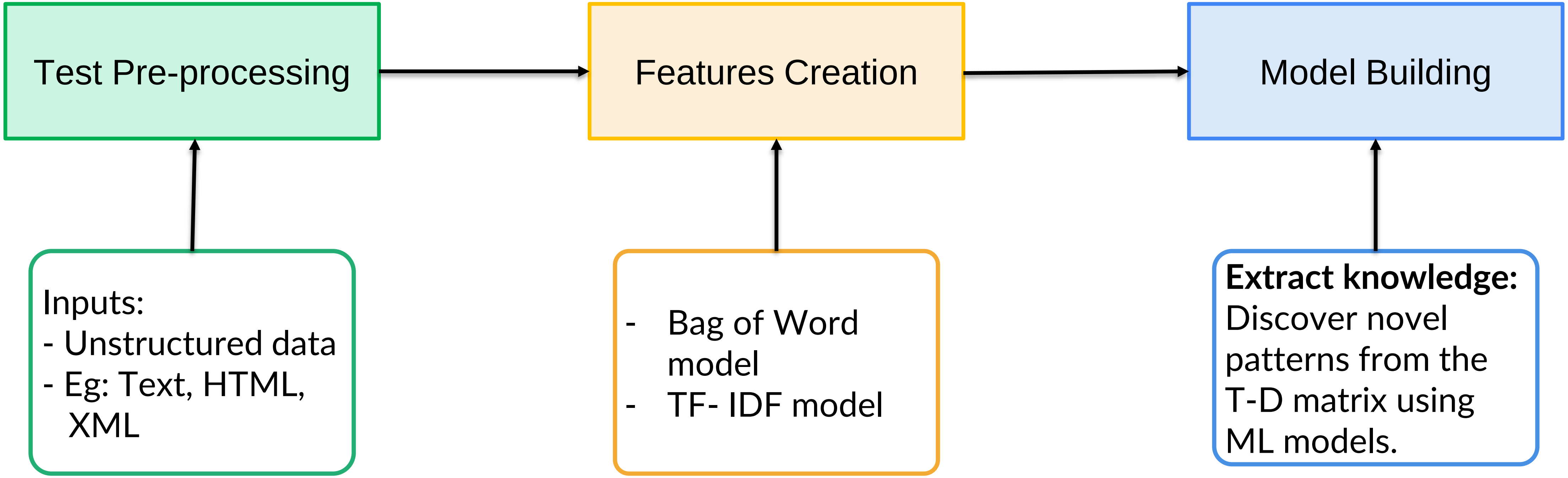
TEXT PRE-PROCESSING

The three-step text mining process



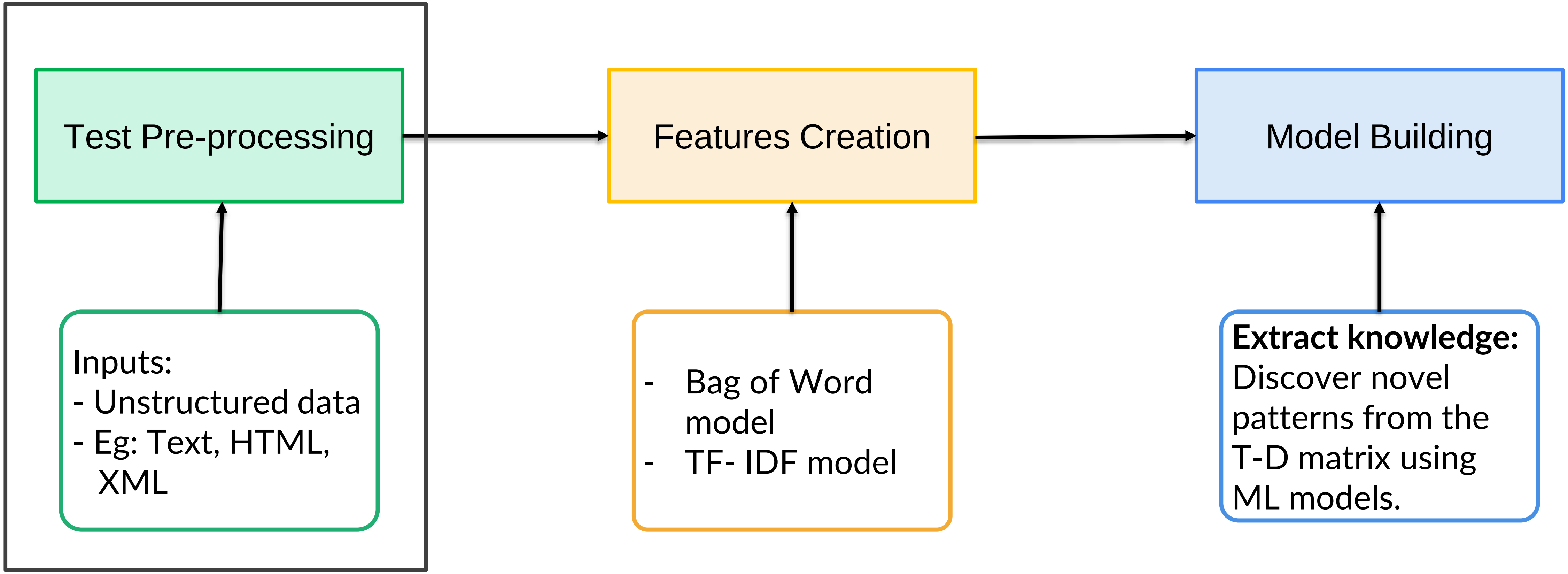
TEXT PRE-PROCESSING

The three-step text mining process



TEXT PRE-PROCESSING

The three-step text mining process



TEXT PRE-PROCESSING

- Why text preprocessing is needed?

TEXT PRE-PROCESSING

- Why text preprocessing is needed?
- Let’s look into the following sentence:

yummy
YUMMY

π Caps!!

“The apple pie was AMAZING and the Food Was Really YuMmY too! You can Checkout the entire Menu @ this Restaurant; you can each out to its website at https://www.zomato.com .”

url

?

TEXT PRE-PROCESSING

- Why text preprocessing is needed?
- Let’s look into the following sentence:

“The apple pie was AMAZING and the Food Was really YuMmY too! You can Checkout the entire Menu @ this Restaurant; you can each out to its website at https://www.zomato.com .”

TEXT PRE-PROCESSING

- Text encoding (ASCII, Unicode, etc.)
- Converting to lower case
- Removing symbols and punctuations
- Handling numbers
- Stop-word removal
- Tokenisation
- Stemming and Lemmatisation

TEXT PRE-PROCESSING

- **Text encoding (ASCII, Unicode, etc.)**
- Converting to lower case
- Removing symbols and punctuations
- Handling numbers
- Stop-word removal
- Tokenisation
- Stemming and Lemmatisation

TEXT PRE-PROCESSING

- We human can read the sentences and characters like:

  "IIT Madras Online Degree is teaching NLP course."

TEXT PRE-PROCESSING

- We human can read the sentences and characters like:

  "IIT Madras Online Degree is teaching NLP course."

- What about **machines**?

- Do they understand the characters of English (a-b, A-B) or any other language?

TEXT PRE-PROCESSING

- We human can read the sentences and characters like:

  "IIT Madras Online Degree is teaching NLP course."

- What about **machines**?
- Do they understand the characters of English (a-b, A-B) or any other language?
- The answer is NO, as they only understand numeric and digits.

TEXT PRE-PROCESSING

- We human can read the sentences and characters like:

  "IIT Madras Online Degree is teaching NLP course."

- What about **machines**?
- Do they understand the characters of English (a-b, A-B) or any other language?
- The answer is NO, as they only understand numeric and digits.
- More specifically, Bits (0 or 1).

TEXT PRE-PROCESSING

- We human can read the sentences and characters like:

  "IIT Madras Online Degree is teaching NLP course."

- What about **machines**?

- Do they understand the characters of English (a-b, A-B) or any other language?
- The answer is NO, as they only understand numeric and digits.
- More specifically, Bits (0 or 1).
- So, how machines will be able to understand the text as they works upon bits?

TEXT ENCODING

- What if we are able to convert characters into group of bits (or numeric form).

TEXT ENCODING

- What if we are able to convert characters into group of bits (or numeric form).

- A: '01000001'
- d: '01100100'
- &: '00100110'

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- It assigns 0 to127 decimal values to the symbols.

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- It assigns 0 to127 decimal values to the symbols.

Character	Decimal Value
A	65
K	75
d	100
z	122
8	56
&	38

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- It assigns 0 to127 decimal values to the symbols.
- These symbols can be:

- Letter (a-z, A-Z)

Character	Decimal Value
A	65
K	75
d	100
z	122
8	56
&	38

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- It assigns 0 to127 decimal values to the symbols.
- These symbols can be:

- Letter (a-z, A-Z)
- Numbers (0-9)

Character	Decimal Value
A	65
K	75
d	100
z	122
8	56
&	38

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- It assigns 0 to127 decimal values to the symbols.
- These symbols can be:

- Letter (a-z, A-Z)
- Numbers (0-9)
- Punctuation marks
- Special and control characters

Character	Decimal Value
A	65
K	75
d	100
z	122
8	56
&	38

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- Each symbols can be represented by **decimal values** or its equivalent **binary**, **octal** or **hexadecimal** values.

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- Each symbols can be represented by **decimal values** or its equivalent **binary**, **octal** or **hexadecimal** values.

Character	Decimal	Binary	Hexadecimal
A	65	01000001	41

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- Each symbols can be represented by **decimal values** or its equivalent **binary**, **octal** or **hexadecimal** values.

Character	Decimal	Binary	Hexadecimal
A	65	01000001	41
K	75	01001011	4B

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- Each symbols can be represented by **decimal values** or its equivalent **binary**, **octal** or **hexadecimal** values.

Character	Decimal	Binary	Hexadecimal
A	65	01000001	41
K	75	01001011	4B
d	100	01100100	64

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- Each symbols can be represented by **decimal values** or its equivalent **binary**, **octal** or **hexadecimal** values.

Character	Decimal	Binary	Hexadecimal
A	65	01000001	41
K	75	01001011	4B
d	100	01100100	64
z	122	01111010	7A
8	56	00111000	38
&	38	0010 0110	26

TEXT ENCODING

US-ASCII: *American Standard Code for Information Interchange*

- Each symbols can be represented by **decimal values** or its equivalent **binary**, **octal** or **hexadecimal** values.

The group of 8 bits can form numbers from **0 (00000000)-255(11111111)** but ASCII does not use the decimal values from 128 to 255.

TEXT ENCODING

Unicode: UTF-8

- Let’s consider the following word in German language:

‘Schrödinger’

TEXT ENCODING

Unicode: UTF-8

- Let’s consider the following word in German language:

  ‘Schrödinger’

- It covers almost all of the characters and symbols.

TEXT ENCODING

Unicode: UTF-8

- Let’s consider the following word in German language:

  ‘Schrödinger’

- It covers almost all of the characters and symbols.
- Which is not the case with ASCII.

TEXT ENCODING

Unicode: UTF-8

- Let's consider the following word in German language:

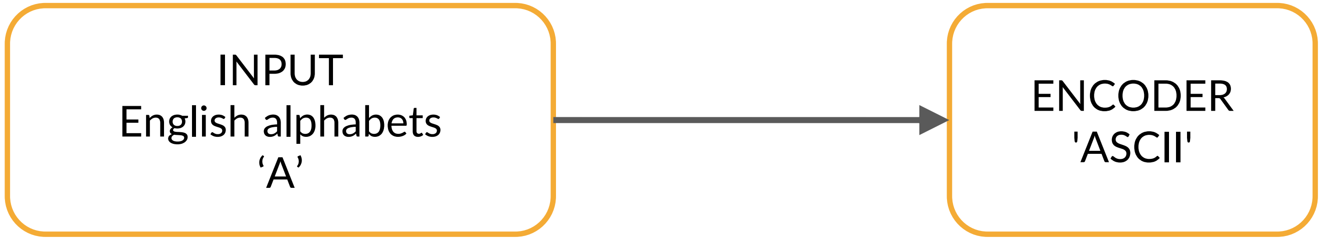
  'Schrödinger'

- It covers almost all of the characters and symbols.
- Which is not the case with ASCII.
- ASCII is the subset of UTF-8 for decimal value 0 to 127.

COMMON TEXT ENCODERS

INPUT
English alphabets
‘A’

COMMON TEXT ENCODERS



COMMON TEXT ENCODERS

INPUT
English alphabets
‘A’

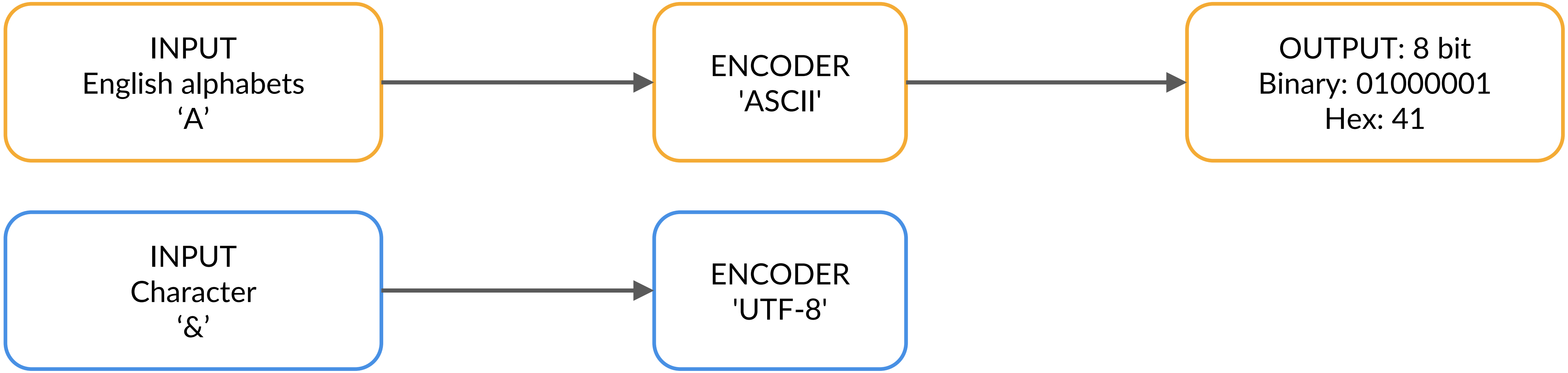
→

ENCODER
'ASCII'

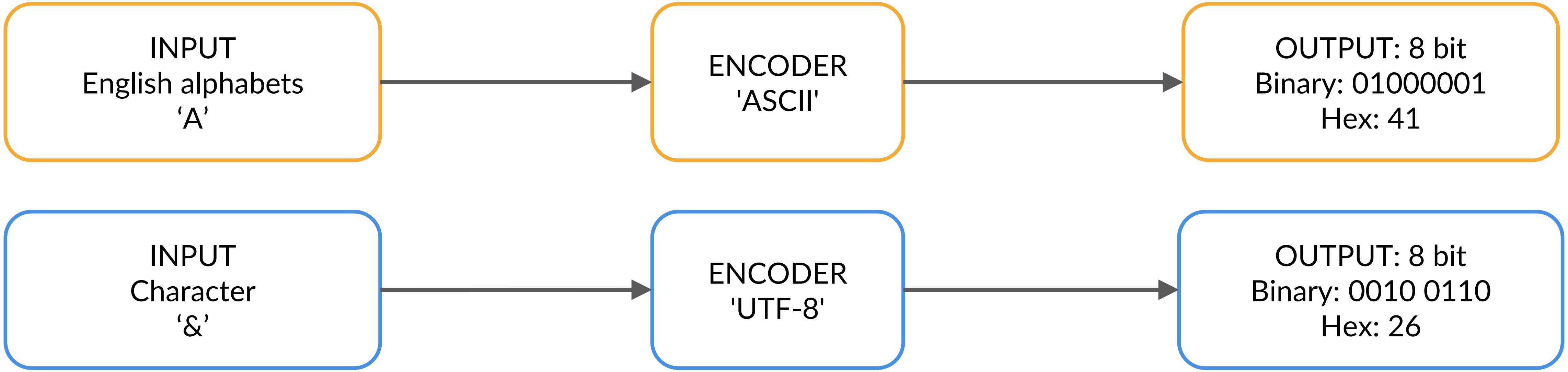
→

OUTPUT: 8 bit
Binary: 01000001
Hex: 41

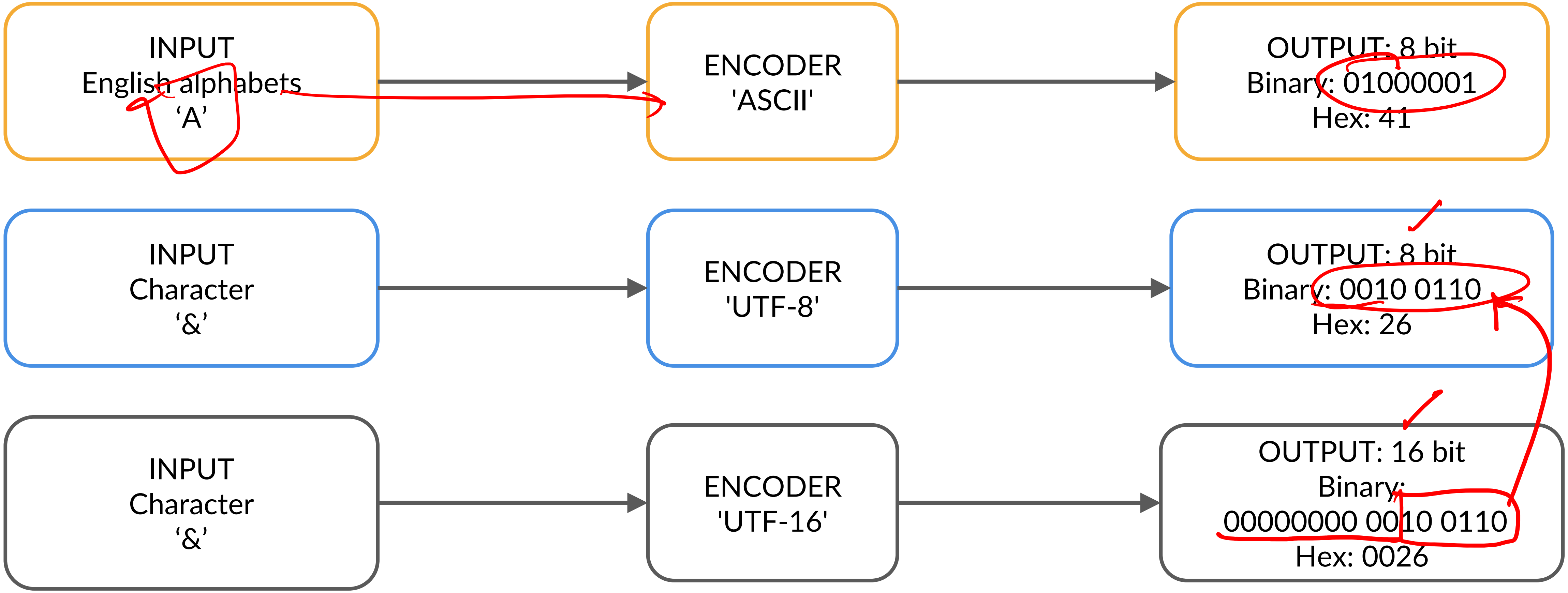
COMMON TEXT ENCODERS



COMMON TEXT ENCODERS



COMMON TEXT ENCODERS



TEXT PRE-PROCESSING

- Text encoding (ASCII, Unicode, etc.)
- **Converting to lower case**
- **Removing symbols and punctuations**
- **Handling numbers**
- Stop-word removal
- Tokenisation
- Stemming and Lemmatisation

TEXT PRE-PROCESSING

“The apple pie was AMAZING and the Food Was Really YuMmY too! You can Checkout the entire Menu at this Restaurant; .”

LOWERCASING

- **Converting to lower case:**

  “The apple pie was AMAZING and the Food Was Really YuMmY too! You can Checkout the entire Menu at this Restaurant; .”

LOWERCASING

- **Converting to lower case:**

“The apple pie was AMAZING and the Food Was Really YuMmY too! You can Checkout the entire Menu at this Restaurant; .”

“the apple pie was amazing and the food was really yummy too! you can checkout the entire menu at this restaurant; .”

? ?

SYMBOLS AND PUNCTUATION REMOVAL

- **Removing symbols and punctuations:**

  "the apple pie was amazing and the food was really yummy too! you can checkout the entire menu at this restaurant; ."

SYMBOLS AND PUNCTUATION REMOVAL(FACESHOT)

- **Removing symbols and punctuations:**

"the apple pie was amazing and the food was really yummy too! you can checkout the entire menu at this restaurant; ."

↓

the apple pie was amazing and the food was really yummy too you can checkout the entire menu at this restaurant.

HANDLING NUMBERS (FACESHOT)

- **Handling numbers:**

- Convert the number into English letter

- Use Regex

- Numeric/Alphanumeric filters based on context

TEXT PRE-PROCESSING (FACESHOT)

- Text encoding (ASCII, Unicode, etc.)
- Converting to lower case
- Removing symbols and punctuations
- Handling numbers
- **Stop-word removal** → Analogous to Outliers
- Tokenisation
- Stemming and Lemmatisation

STOP WORDS

Leaders keep on coming and going. Every once in a while, we come across someone as pre-eminent as APJ Abdul Kalam. His name will certainly go down in history as one of the greatest presidents that India has ever seen. Moreover, people will also remember him as a brilliant scientist. The man was a precious gem for each and every Indian.

Source: https://www.toppr.com/guides/speech-for-students/apj-abdul-kalam-speech/

STOP WORDS

Leaders keep **on** coming **and** going. Every once **in a** while, **we** come across someone **as** pre-eminent **as** APJ Abdul Kalam. **His** name **will** certainly go down **in** history **as** one **of the** greatest presidents **that** India **has** ever seen. Moreover, people **will also** remember **him as a** brilliant scientist. **The** man **was a** precious gem **for** each **and** every Indian.

Source: https://www.toppr.com/guides/speech-for-students/apj-abdul-kalam-speech/

STOP WORDS

- **Highly frequent words:** Such as **'His', 'an', 'was'** or **'the'**

STOP WORDS

Reduce noise
Reduce size

- **Highly frequent words:** Such as 'His', 'an', 'was' or 'the'

STOP WORDS

Reduce noise
Reduce size

- **Highly frequent words:** Such as 'His', 'an', 'was' or 'the'
- **Significant words:** Typically more important to understand the text

STOP WORDS

Reduce noise
Reduce size

- **Highly frequent words:** Such as ‘His’, ‘an’, ‘was’ or ‘the’

- **Significant words:** Typically more important to understand the text

Example: ‘leaders’, ‘coming’, ‘going’, ‘greatest’ or ‘Indian’ etc.

STOP WORDS

Reduce noise
Reduce size

- **Highly frequent words:** Such as **'His'**, **'an'**, **'was'** or **'the'**

- **Significant words:** Typically more important to understand the text

Example: **'leaders'**, **'coming'**, **'going'**, **'greatest'** or **'Indian'** etc.

- **Rarely occurring words:** Less important than significant words

STOP WORDS

Reduce noise
Reduce size

- **Highly frequent words:** Such as **'His'**, **'an'**, **'was'** or **'the'**

- **Significant words:** Typically more important to understand the text

Example: **'leaders'**, **'coming'**, **'going'**, **'greatest'** or **'Indian'** etc.

- **Rarely occurring words:** Less important than significant words

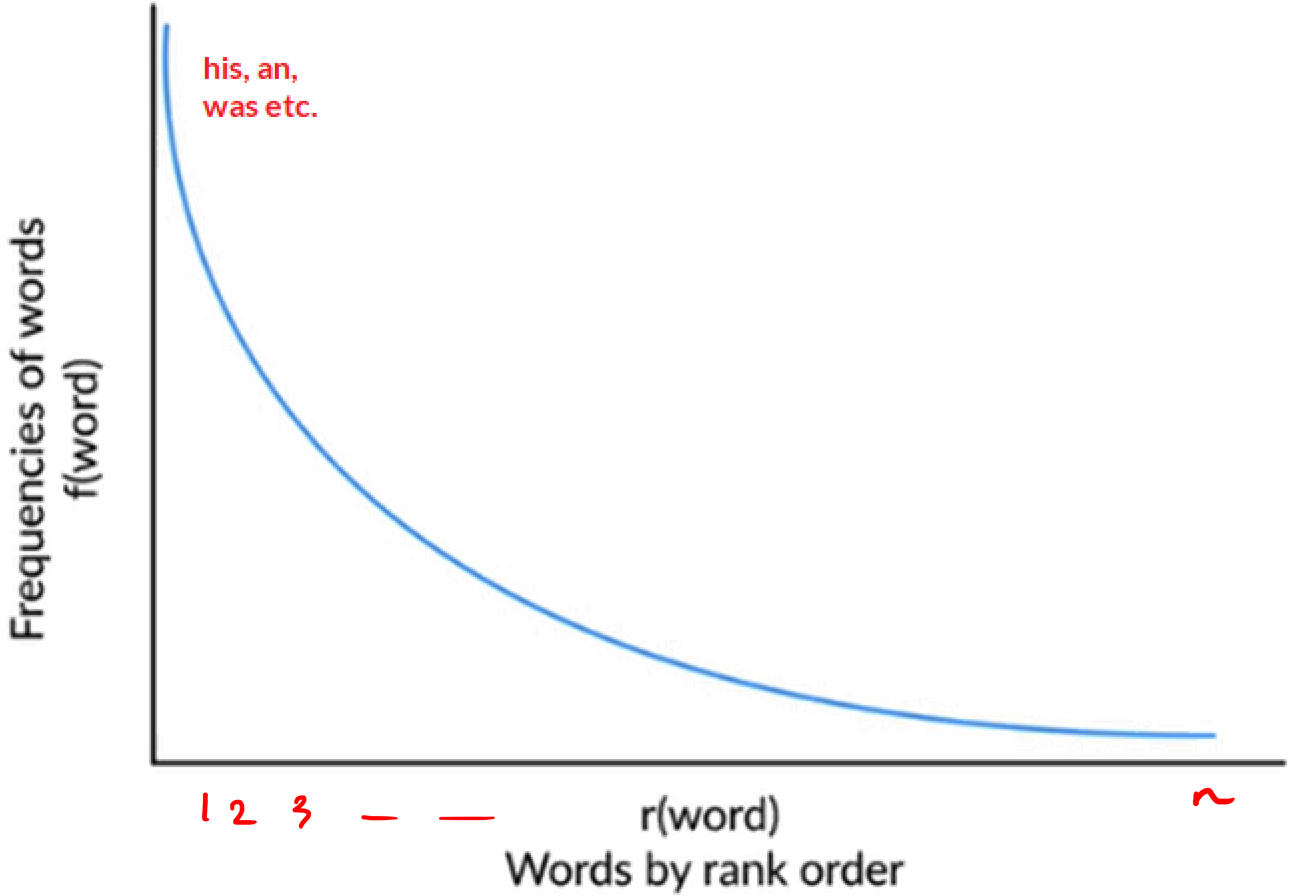
Example: **'pre-eminent'**, **'brilliant'**, **'Abdul'** or **'gem'** etc.

STOP WORDS

Leaders keep **on** coming **and** going. Every once **in a** while, **we** come across someone **as** pre-eminent **as** APJ Abdul Kalam. **His** name **will** certainly go down **in** history **as** one **of the** greatest presidents **that** India **has** ever seen. Moreover, people **will also** remember **him as a** brilliant scientist. **The** man **was a** precious gem **for** each **and** every Indian.

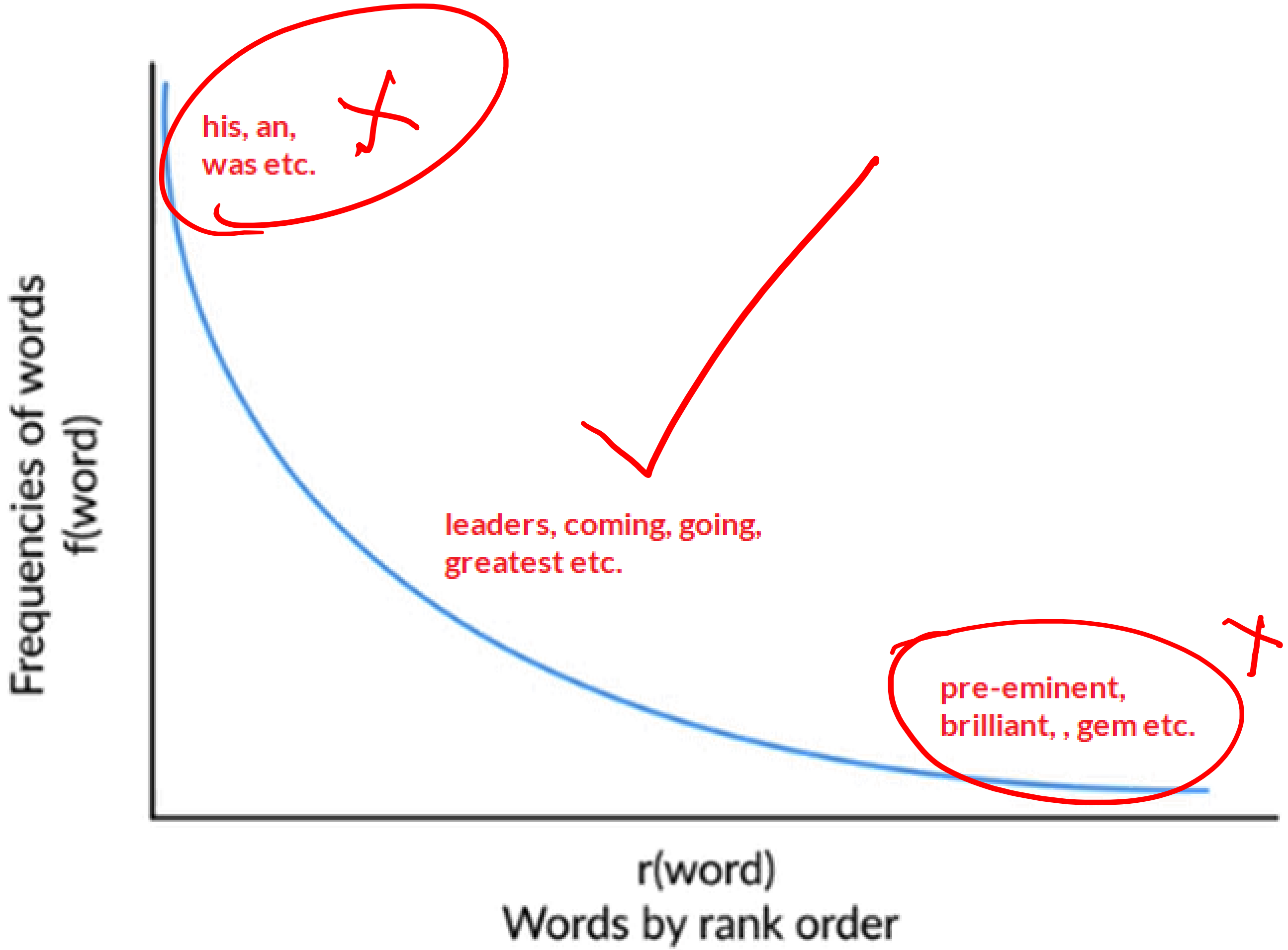
Source: https://www.toppr.com/guides/speech-for-students/apj-abdul-kalam-speech/

STOP WORDS



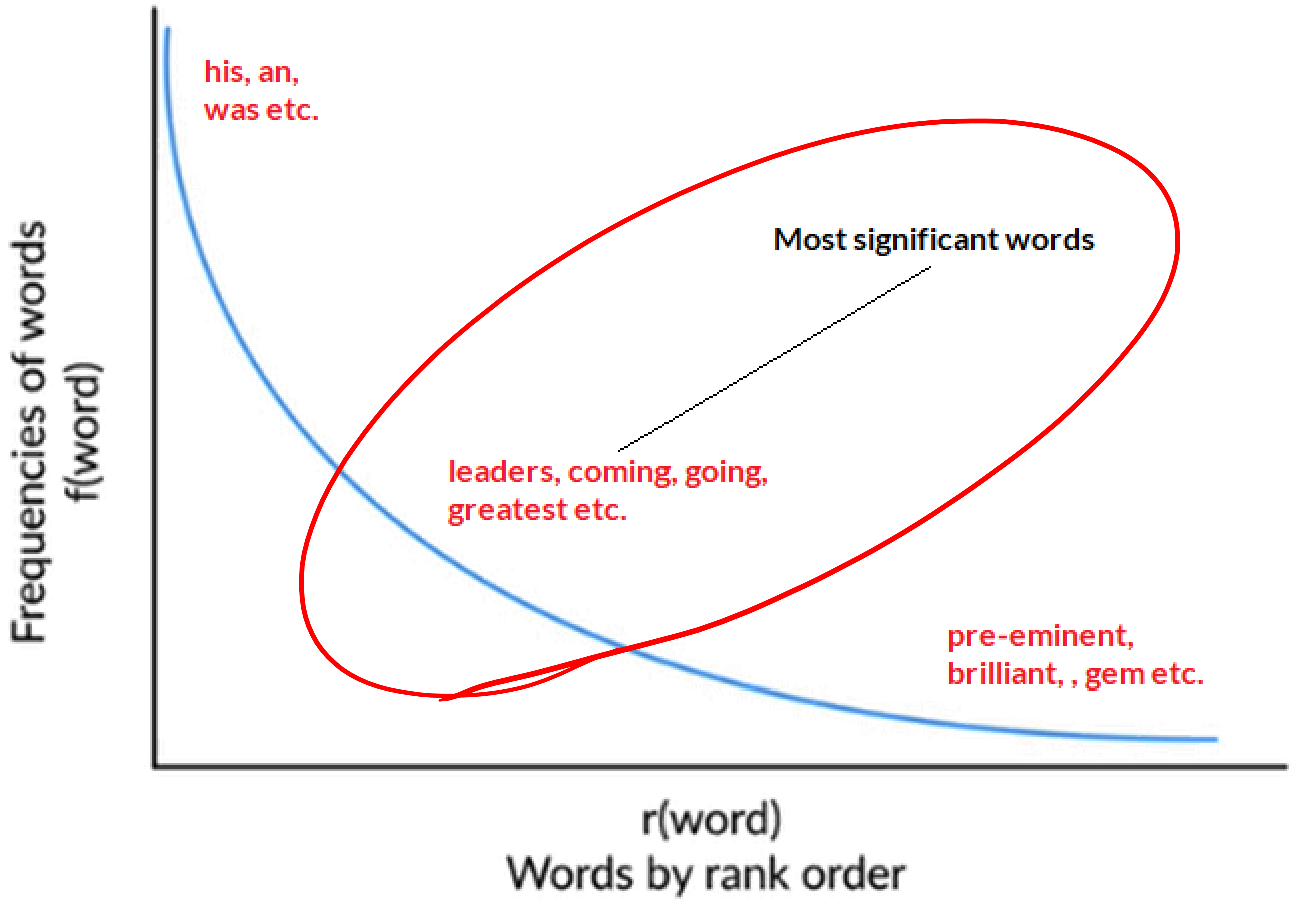
his, an,
was etc.
Frequencies of words
f(word)
1 2 3 — —
r(word)
Words by rank order
n

STOP WORDS



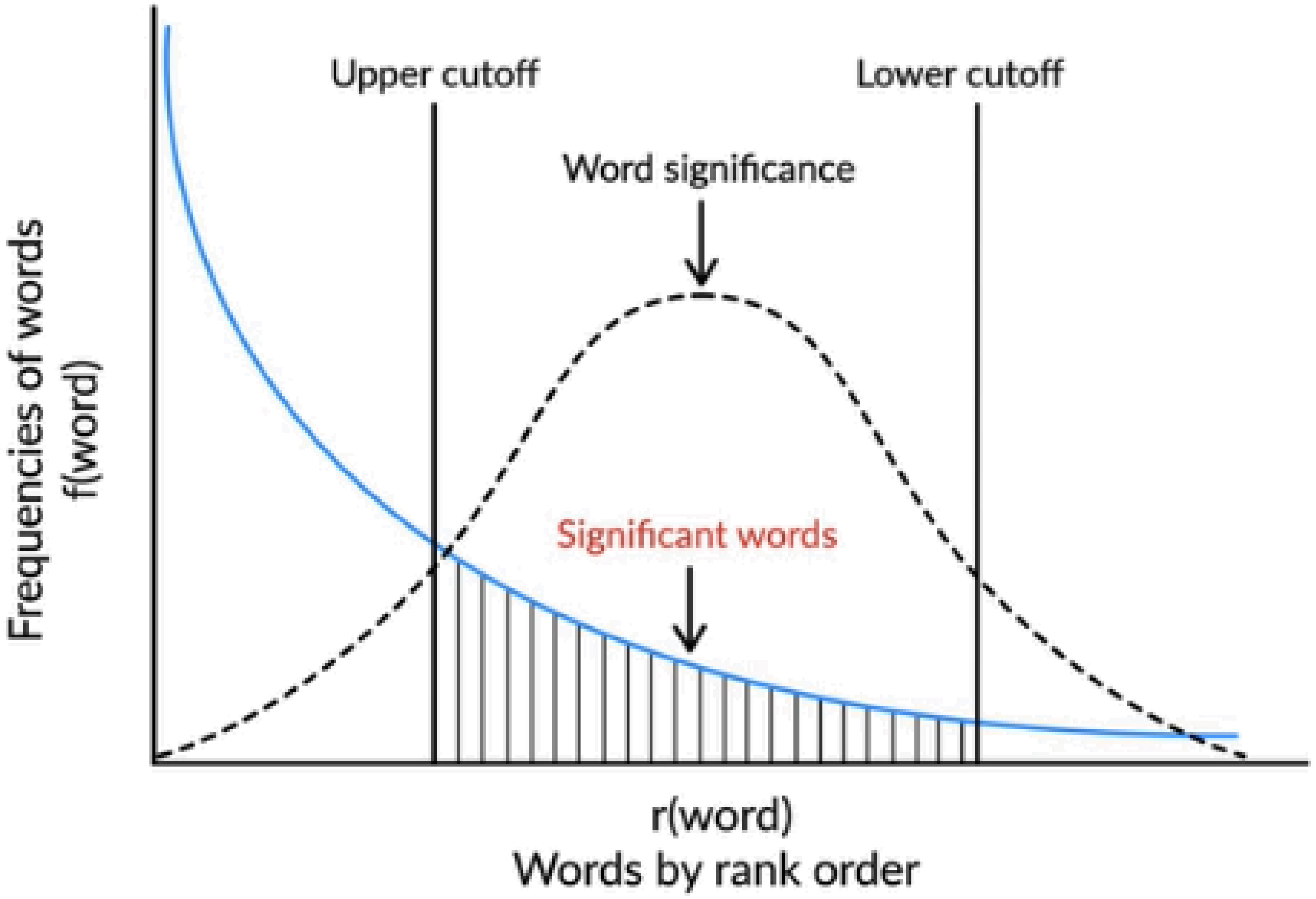
his, an,
was etc.
leaders, coming, going,
greatest etc.
pre-eminent,
brilliant, , gem etc.
Frequencies of words
f(word)
r(word)
Words by rank order

STOP WORDS

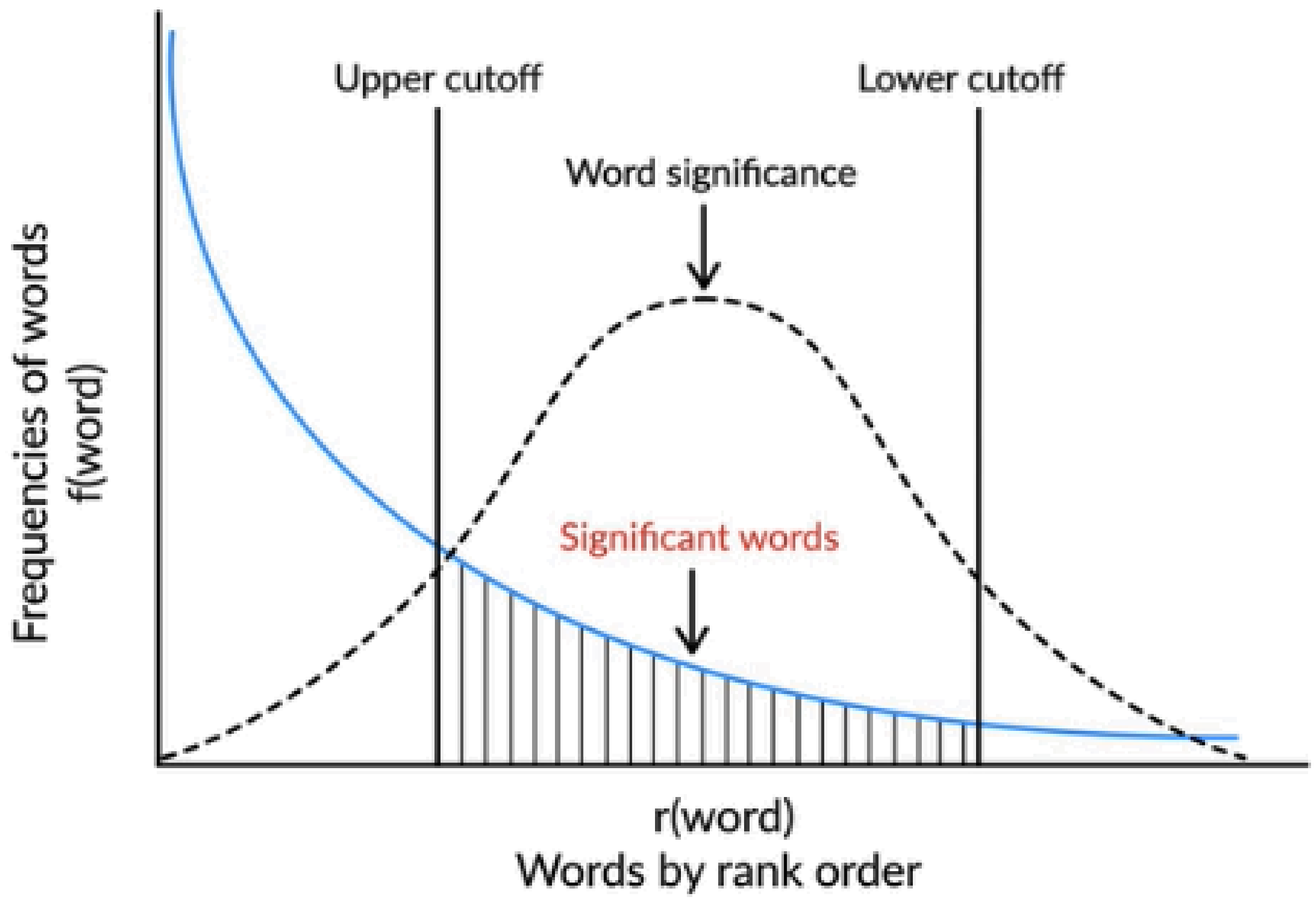


his, an,
was etc.
Most significant words
leaders, coming, going,
greatest etc.
pre-eminent,
brilliant, , gem etc.
Frequencies of words
f(word)
r(word)
Words by rank order

STOP WORDS


Upper cutoff
Lower cutoff
Word significance
Significant words
Frequencies of words
f(word)
r(word)
Words by rank order

STOP WORDS



Zipf's Law: f(word) * r(word) ~ constant

TEXT PRE-PROCESSING (FACESHOT)

- Text encoding (ASCII, Unicode, etc.)
- Converting to lower case
- Removing symbols and punctuations
- Handling numbers
- Stop-word removal
- **Tokenisation**
- Stemming and Lemmatisation

TOKENISATION

Tokenisation: Splitting text into smaller elements (characters, words, sentences, paragraphs)

TOKENISATION

Tokenisation: Splitting text into smaller elements (characters, words, sentences, paragraphs)

> the apple pie was amazing and the food was really yummy too you can checkout the entire menu at this restaurant.

TOKENISATION

Tokenisation: Splitting text into smaller elements (characters, words, sentences, paragraphs)

the apple pie was amazing and the food was really yummy too you can checkout the entire menu at this restaurant.

list of words → unit of modeling

[the, apple, pie, was, amazing, and, the, food, was, really, yummy, too, you, can, checkout, the, entire, menu, at, this, restaurant]

TOKENISATION

Tokenisation: Splitting text into smaller elements (characters, words, sentences, paragraphs)

- **Word tokeniser** splits text into different words.

TOKENISATION

Tokenisation: Splitting text into smaller elements (characters, words, sentences, paragraphs)

- **Word tokeniser** splits text into different words.
- **Sentence tokeniser** splits text into different sentences.

TOKENISATION

Why tokenisation?

- Features have to be extracted based on the unit of modeling.
- Unit of modeling can be word or sentence or paragraphs based on learning objective.
- Hence, after deciding learning objective, tokenization has to be done at the respective level.

TEXT PRE-PROCESSING

- Text encoding (ASCII, Unicode, etc.)
- Converting to lower case
- Removing symbols and punctuations
- Handling numbers
- Stop-word removal
- Tokenisation
- **Stemming and Lemmatisation**

STEMMING

Stemming: It removes or stems the last few characters of a word.

STEMMING

Stemming: It removes or stems the last few characters of a word.

	Stemmed
playing	play
plays	play
played	play
am	am
are	are
is	is
goes	goe
going	go
went	went
gone	gone

Analogous to Scaling

STEMMING

Stemming: It removes or stems the last few characters of a word.

- Often leading to incorrect meanings and spelling.

rule base

	Stemmed
playing	play
plays	play
played	play
am	am
are	are
is	is
goes	goe
going	go
went	went
gone	gone

written a rule

ed

s

ing

STEMMING AND LEMMATISATION

Lemmatisation: It considers the context and converts the word into its meaningful base form.

STEMMING AND LEMMATISATION

Lemmatisation: It considers the context and converts the word into its meaningful base form.

	Stemmed	Lemmatised
playing	play	
plays	play	
played	play	
am	am	
are	are	
is	is	
goes	goe	
going	go	
went	went	
gone	gone	

STEMMING AND LEMMATISATION

Lemmatisation: It considers the context and converts the word into its meaningful base form.

word → dictionary → root

	Stemmed	Lemmatised
playing	play	play
plays	play	play
played	play	play
am	am	be
are	are	be
is	is	be
goes	goe	go
going	go	go
went	went	go
gone	gone	go

STEMMING AND LEMMATISATION

Lemmatisation: It considers the context and converts the word into its meaningful base form.

- The base form is called **'Lemma'.**

	Stemmed	Lemmatised
playing	play	play
plays	play	play
played	play	play
am	am	be
are	are	be
is	is	be
goes	goe	go
going	go	go
went	went	go
gone	gone	go

Stemming (rule based) (fast)

Lemma → dictionary (taken time)

STEMMING AND LEMMATISATION

Lemmatisation: It considers the context and converts the word into its meaningful base form.

- The base form is called **'Lemma'.**
- A lemma is the canonical form, dictionary form, or citation form of a set of words.

	Stemmed	Lemmatised
playing	play	play
plays	play	play
played	play	play
am	am	be
are	are	be
is	is	be
goes	goe	go
going	go	go
went	went	go
gone	gone	go

STEMMING AND LEMMATISATION

Lemmatisation: It considers the context and converts the word into its meaningful base form.

- The base form is called **'Lemma'.**
- A lemma is the canonical form, dictionary form, or citation form of a set of words.
- Sometimes, the same word can have multiple and different lemmas.

	Stemmed	Lemmatised
playing	play	play
plays	play	play
played	play	play
am	am	be
are	are	be
is	is	be
goes	goe	go
going	go	go
went	went	go
gone	gone	go

STEMMING Vs LEMMATISATION

"Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, [1][2] similar to data mining."

STEMMING Vs LEMMATISATION

“Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, [1][2] similar to data mining.”

Stemmed sentence:

“Data scienc is an interdisciplinari field that use scientif method , process , algorithm and system to extract knowledg and insight from data in various form , both structur and unstructur , [1][2] similar to data mine.”

STEMMING Vs LEMMATISATION

"Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, [1][2] similar to data mining."

Lemmatised sentence:

"Data science is an interdisciplinary field that use scientific method , process , algorithm and system to extract knowledge and insight from data in various form , both structure and unstructure , [1][2] similar to data mine."

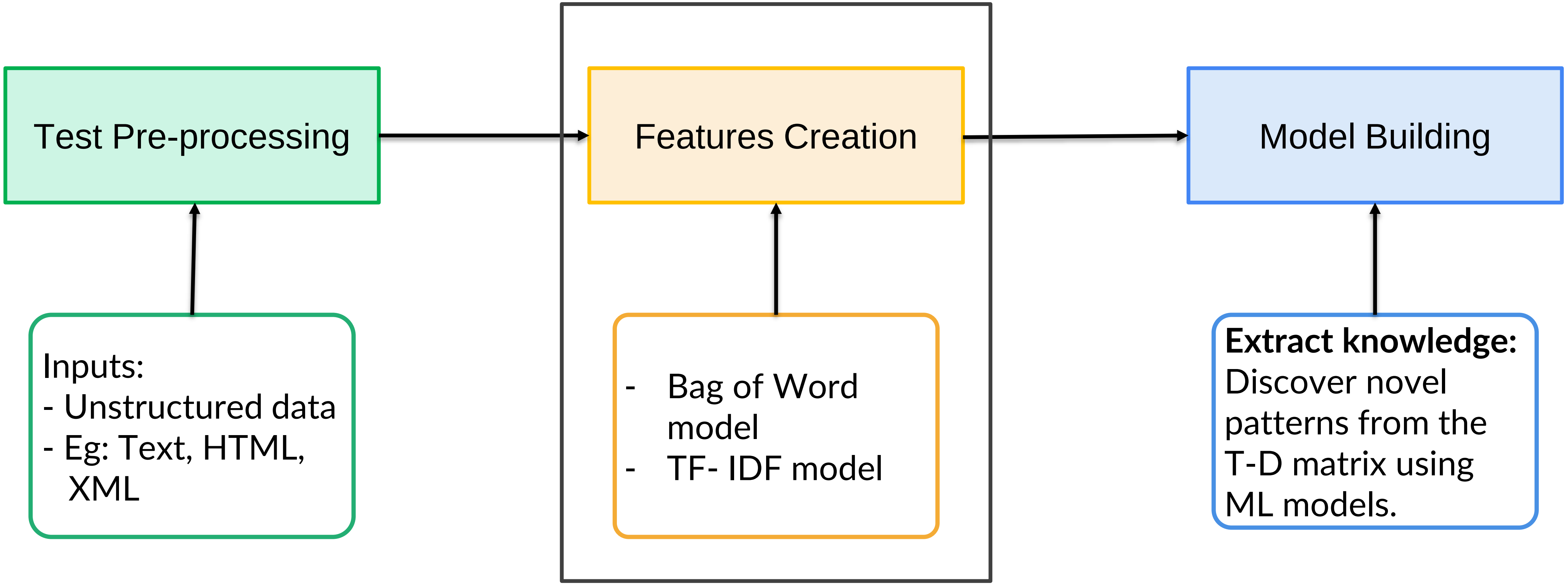
NLP TOOLKITS

NLTK - https://www.nltk.org/

Spacy - https://spacy.io/

TEXT PRE-PROCESSING

The three-step text mining process



TEXT PRE-PROCESSING

- **Features Creation**

- Bag of Word Model

- TF-IDF Model

FEATURES CREATION

- You have pre-processed text!!

FEATURES CREATION

- You have pre-processed text!!
- How you can use this textual data for model building (like Logistic Regression) which can predict something like sentiments.

FEATURES CREATION

- You have pre-processed text!!
- How you can use this textual data for model building (like Logistic Regression) which can predict something like sentiments.
- For model, you need features.

FEATURES CREATION

- You have pre-processed text!!
- How you can use this textual data for model building (like Logistic Regression) which can predict something like sentiments.
- For model, you need features.
- How do you create features using textual data?

FEATURES CREATION

Phone’s review	Sentiment
camera, not, good	
screen, awesome, microprocessor, slow, happy	
hotspot, not, connect	
…	
…	

FEATURES CREATION

Phone's review	Sentiment
camera, not, good	Negative
screen, awesome, microprocessor, slow, happy	Positive
hotspot, not, connect	Negative
…	…
…	…

FEATURES CREATION

	Phone's review	**Sentiment**
Doc_1	camera, not, good	Negative
Doc_2	screen, awesome, microprocessor, slow, happy	Positive
Doc_3	hotspot, not, connect	Negative

FEATURES CREATION

	camera	not	good	screen	awesome	microprocessor	slow	happy	hotspot	connect	…	Sentiment
Doc_1												Negative
Doc_2												Positive
Doc_3												Negative
Doc_4												…
Doc_5												…

	Phone's review	**Sentiment**
Doc_1	camera, not, good	Negative
Doc_2	screen, awesome, microprocessor, slow, happy	Positive
Doc_3	hotspot, not, connect	Negative

FEATURES CREATION

	camera	not	good	screen	awesome	microprocessor	slow	happy	hotspot	connect	…	Sentiment
Doc_1	value	value	value									Negative
Doc_2				value	value	value	value	Value				Positive
Doc_3		value				value				value		Negative
Doc_4	…	…	…	…	…	…	…	…	…	…	…	…
Doc_5	…	…	…	…	…	…	…	…	…	…	…	…

	Phone's review	**Sentiment**
Doc_1	camera, not, good	Negative
Doc_2	screen, awesome, microprocessor, slow, happy	Positive
Doc_3	hotspot, not, connect	Negative

FEATURES CREATION

	camera	not	good	screen	awesome	microprocessor	slow	happy	hotspot	connect	…	Sentiment
Doc_1	value	value	value	0	0	0	0	0	0	0	0	Negative
Doc_2	0	0	0	value	value	value	value	Value	0	0	0	Positive
Doc_3	0	value	0	0	0	value	0	0	0	value	0	Negative
Doc_4	…	…	…	…	…	…	…	…	…	…	…	…
Doc_5	…	…	…	…	…	…	…	…	…	…	…	…

	Phone's review	**Sentiment**
Doc_1	camera, not, good	Negative
Doc_2	screen, awesome, microprocessor, slow, happy	Positive
Doc_3	hotspot, not, connect	Negative

FEATURES CREATION

	camera	not	good	screen	awesome	microprocessor	slow	happy	hotspot	connect	…	Sentiment
Doc_1	value	value	value	0	0	0	0	0	0	0	0	Negative
Doc_2	0	0	0	value	value	value	value	Value	0	0	0	Positive
Doc_3	0	value	0	0	0	value	0	0	0	value	0	Negative
Doc_4	…	…	…	…	…	…	…	…	…	…	…	…
Doc_5	…	…	…	…	…	…	…	…	…	…	…	…
…	…	…	…	…	…	…	…	…	…	…	…	…

To fill these values, we have two models:

1. Bag of Words Model (BOW)
2. TF-IDF Model

TEXT PRE-PROCESSING

- Features Creation

- Bag of Word Model

- TF-IDF Model

BAG OF WORDS MODEL

- The bag of words model is used to create features from text.

BAG OF WORDS MODEL

- The bag of words model is used to create features from text.
- It is a cross-tab of frequency of each unique word with respect to all documents.

BAG OF WORDS MODEL

Document 1

The quick brown fox jumped over the lazy dog's back.

BAG OF WORDS MODEL

Document 1

The quick brown fox jumped over the lazy dog's back.

Document 2

Now is the time for all good men to come to the aid of their party.

BAG OF WORDS MODEL

Stop Words

Document 1

The quick brown fox jumped over the lazy dog's back.

Document 2

Now is the time for all good men to come to the aid of their party.

BAG OF WORDS MODEL

Tokenisation

Document 1

[quick. brown fox, jump, over, lazy, dog, back]

Document 2

[now, time, all good, men, come, aid, their, party]

BAG OF WORDS MODEL

Document 1

[quick. brown fox, jump, over, lazy, dog, back]

	aid	all	brown	come	dog	fox	good	jump	lazy	men	now	over	party	quick	their	time
Document1	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**0**
Document2																

BAG OF WORDS MODEL

Document 2

[now, time, all good, men, come, aid, their, party]

	aid	all	brown	come	dog	fox	good	jump	lazy	men	now	over	party	quick	their	time
Document1	0	0	1	0	1	1	0	1	1	0	0	1	0	1	0	0
Document2	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**

BAG OF WORDS MODEL

	aid	all	brown	come	dog	fox	good	jump	lazy	men	now	over	party	quick	their	time
Document1	0	0	1	0	1	1	0	1	1	0	0	1	0	1	0	0
Document2	1	1	0	1	0	0	1	0	0	1	1	0	1	0	1	1

Count Vector

TEXT PRE-PROCESSING

- Features Creation

- Bag of Word Model

- TF-IDF Model

TF-IDF MODEL

- Stop words can't give the relative importance of a word in a particular document but the TF-IDF can give.

TF-IDF MODEL

- Stop words can’t give the relative importance of a word in a particular document but the TF-IDF can give.
- There are two terms for TF-IDF:

TF-IDF MODEL

- Stop words can’t give the relative importance of a word in a particular document but the TF-IDF can give.
- There are two terms for TF-IDF:

1. Term Frequency (TF):

2. Inverse Document Frequency (IDF):

TF-IDF MODEL

- Stop words can’t give the relative importance of a word in a particular document but the TF-IDF can give.
- There are two terms for TF-IDF:

1. Term Frequency (TF):

(Number of times a word appears in a document/ Total number of words in that document)

2. Inverse Document Frequency (IDF):

TF-IDF MODEL

- Stop words can't give the relative importance of a word in a particular document but the TF-IDF can give.
- There are two terms for TF-IDF:

1. Term Frequency (TF):

(Number of times a word appears in a document/ Total number of words in that document)

2. Inverse Document Frequency (IDF):

IDF= Log(Total number of documents(N)/ Number of documents containing the word)

TF-IDF MODEL

- Stop words can't give the relative importance of a word in a particular document but the TF-IDF can give.
- There are two terms for TF-IDF:

1. Term Frequency (TF):

(Number of times a word appears in a document/ Total number of words in that document)

2. Inverse Document Frequency (IDF):

IDF= Log(Total number of documents(N)/ Number of documents containing the word)

TF-IDF Score = TF X IDF

TF-IDF MODEL

- Let’s look into the following three documents:

Document1: “ Harry Potter is a great movie. It is based on Harry’s life.“

Document2: "The success of a song depends on the music.“

Document3: “There is a new movie releasing this week. The movie is fun to watch."

TF-IDF MODEL

Document1: “Harry Potter is a great movie. It is based on Harry’s life.“
Document2: "The success of a song depends on the music.“
Document3: “There is a new movie releasing this week. The movie is fun to watch.“

	Based	depend	fun	potter	great	movie	music	new	release	song	success	life	harry	watch	week
1															
2															
3															

TF-IDF MODEL

Document1: “Harry Potter is a great movie. It is based on Harry’s life.“
Document2: "The success of a song depends on the music.“
Document3: “There is a new movie releasing this week. The movie is fun to watch.“

	Based	depend	fun	potter	great	movie	music	new	release	song	success	life	harry	watch	week
1															
2															
3															

- TF Score (movie, document1):

TF-IDF MODEL

Document1: "Harry Potter is a great movie. It is based on Harry's life."
Document2: "The success of a song depends on the music."
Document3: "There is a new movie releasing this week. The movie is fun to watch."

	Based	depend	fun	potter	great	**movie**	music	new	release	song	success	life	harry	watch	week
1															
2															
3															

- TF Score (movie, document1): 1

TF-IDF MODEL

Document1: "Harry Potter is a great movie. It is based on Harry's life."
Document2: "The success of a song depends on the music."
Document3: "There is a new movie releasing this week. The movie is fun to watch."

	Based	depend	fun	potter	great	movie	music	new	release	song	success	life	harry	watch	week
1															
2															
3															

- TF Score (movie, document1): 1/

TF-IDF MODEL

Document1: “Harry Potter is a great movie. It is based on Harry’s life.“
Document2: "The success of a song depends on the music.“
Document3: “There is a new movie releasing this week. The movie is fun to watch.“

	Based	depend	fun	potter	great	**movie**	music	new	release	song	success	life	harry	watch	week
1															
2															
3															

- TF Score (movie, document1): 1/7

TF-IDF MODEL

Document1: “Harry Potter is a great movie. It is based on Harry’s life.“
Document2: "The success of a song depends on the music.“
Document3: “There is a new movie releasing this week. The movie is fun to watch.“

	Based	depend	fun	potter	great	**movie**	music	new	release	song	success	life	harry	watch	week
1															
2															
3															

- TF Score (movie, document1): 1/7
- IDF Score (movie): log(3/2)

TF-IDF MODEL

Document1: "Harry Potter is a great movie. It is based on Harry's life."
Document2: "The success of a song depends on the music."
Document3: "There is a new movie releasing this week. The movie is fun to watch."

	Based	depend	fun	potter	great	movie	music	new	release	song	success	life	harry	watch	week
1						**0.025**									
2															
3															

- TF Score (movie, document1): 1/7
- IDF Score (movie): log(3/2)
- TF- IDF Score (movie, document1): 1/7 X log(3/2) = 0.025

TF-IDF MODEL

Document1: "Harry Potter is a great movie. It is based on Harry's life."
Document2: "The success of a song depends on the music."
Document3: "There is a new movie releasing this week. The movie is fun to watch."

	Based	depend	fun	potter	great	**movie**	music	New	release	song	success	life	harry	watch	week
1	0.08	0	0	0.032	0.67	**0.025**	0	0	0	0	0	0.078	0.123	0	0
2	0	0.8	0	0	0	0	0.55	0	0	0.02	0.054	0	0	0	0
3	0	0	0.10	0	0	0.078	0	0.87	0.032	0	0	0	0	0.41	0.078

- TF Score (movie, document1): 1/7
- IDF Score (movie): log(3/2)
- TF- IDF Score (movie, document1): 1/7 X log(3/2) = 0.025

Note: The TF-IDF values may be incorrect in the matrix.

TEXT PRE-PROCESSING

The three-step text mining process