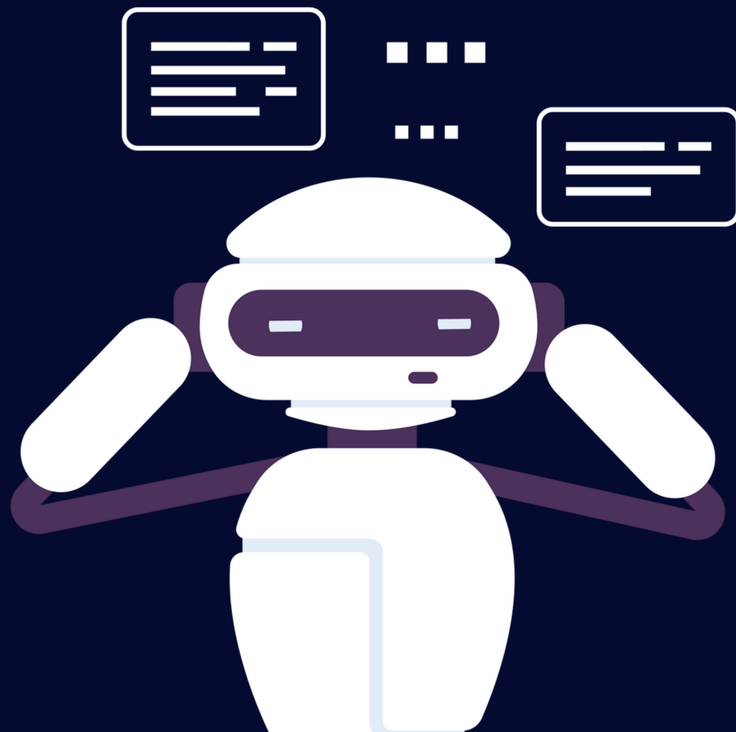
TEXT REPRESENTATION

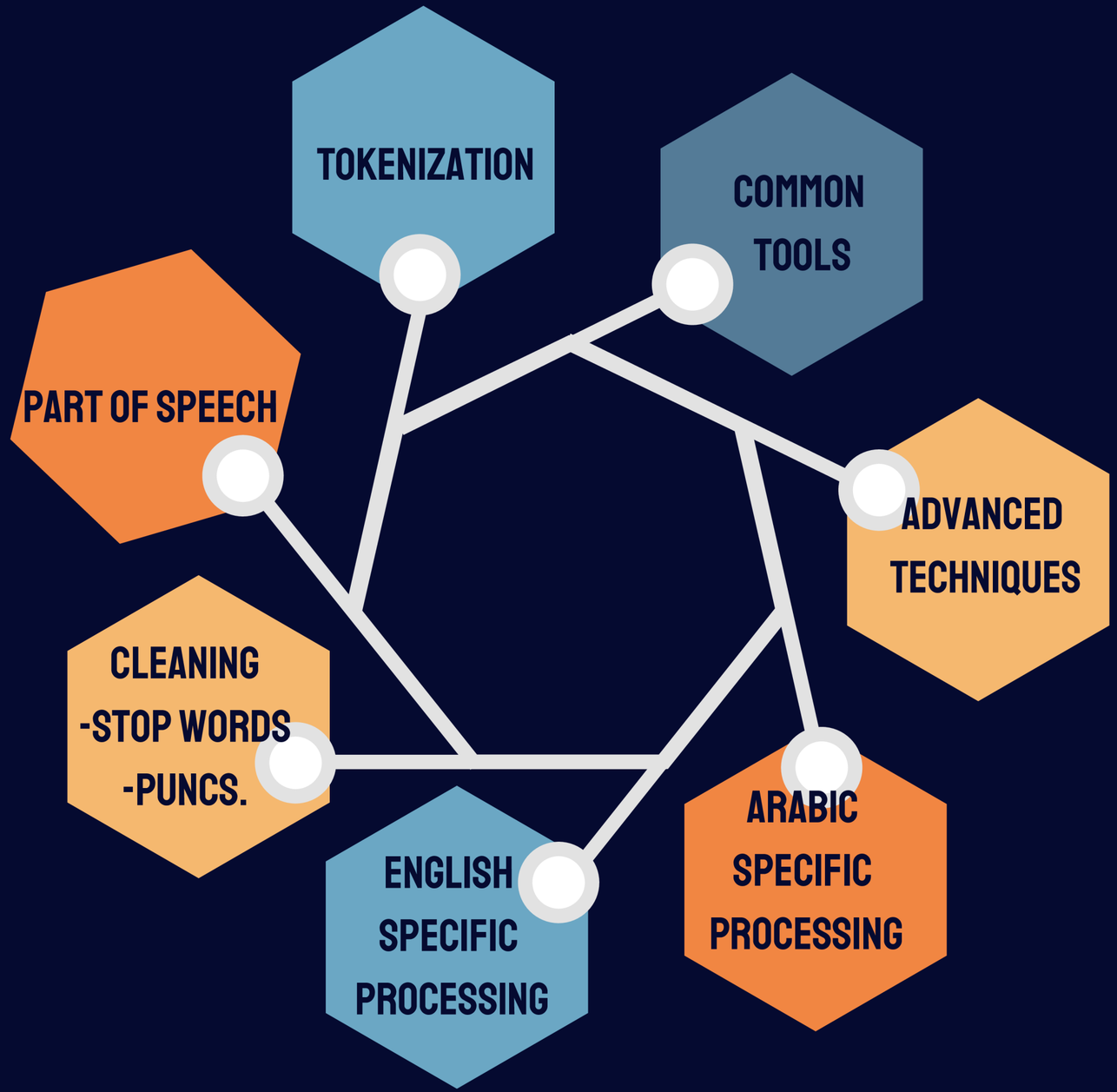
Natural Language Processing

Presented by Mohamed Atwan

Revision **



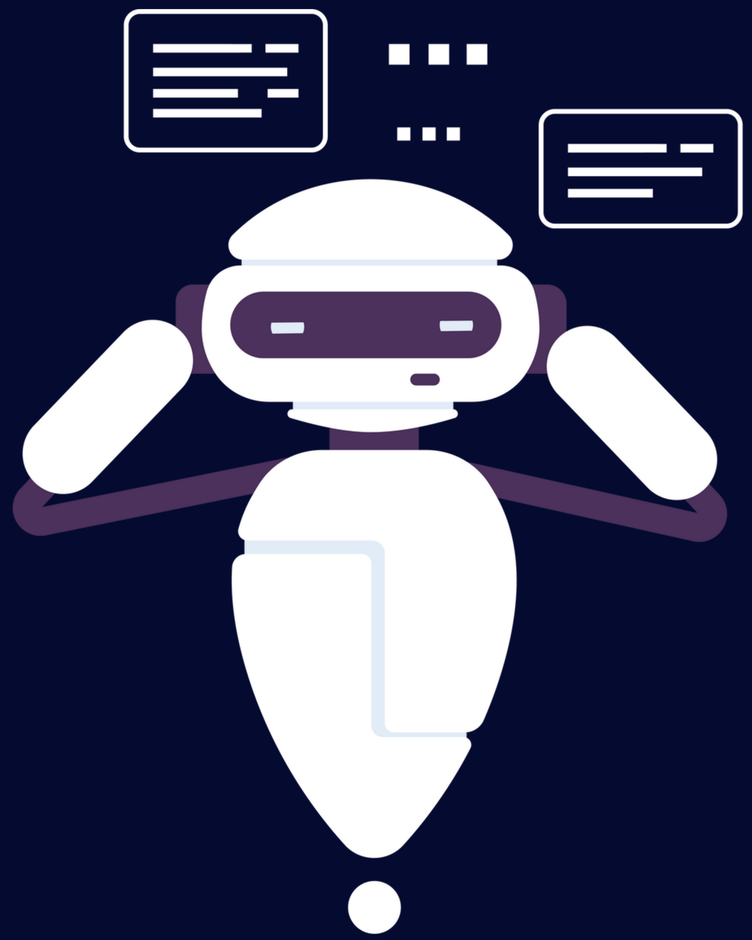
GENERAL STEPS IN TEXT PROCESSING



Quizz Time

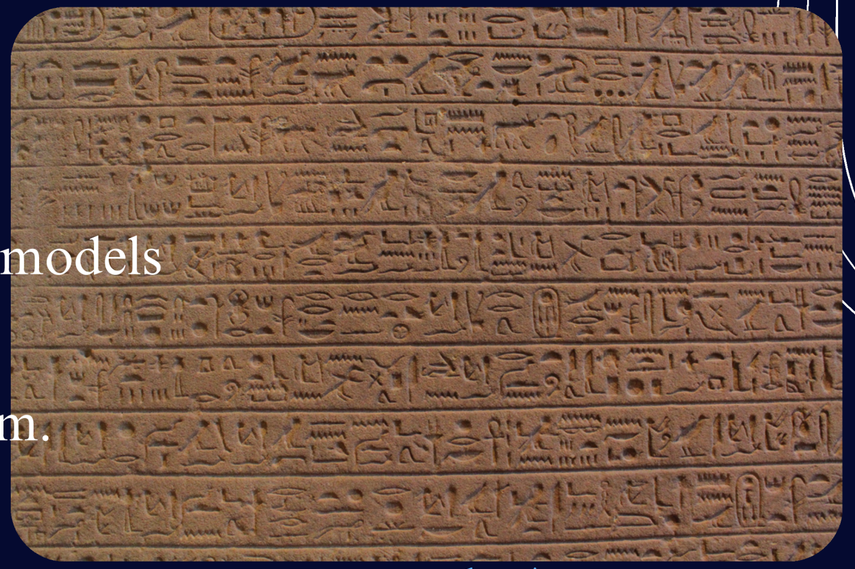
A JOURNEY TO UNDERSTANDING AND PROCESSING TEXTS

1. Introduction to Text Representation
2. Bag of Words (**BOW**)
3. Term Frequency-Inverse Document Frequency (**TF-IDF**)
4. N-GRAM
5. **Practical Demonstration**



INTRODUCTION TO TEXT REPRESENTATION

- Why Text Representation is Important:
  - Text is ؟؟؟ (**St. Un St**) data; machine learning models cannot process it directly.
  - Text needs to be converted into numerical form.



- Challenges:
  - **Variability** in words, word **order**, and **context**.
  - **Ambiguities**, especially in languages like Arabic.

I Read A **book**
I **book** A ticket

الكتاب **على** المكتب
ستجد **على** فى المكتب

- Classical methods include **Bag of Words (BOW)**, **TF-IDF .**

BAG OF WORDS (BOW)

SIMPLE WAY TO REPRESENT TEXT BY COUNTING THE FREQUENCY OF EACH WORD.

- **Features:** Each word in the document is treated as a **feature**, with its **frequency count** as **its value**.
- **Limitations:**
  - **No word order**
  - **No Context (Rules)**
  - **Stop Words Issue**
  - **Large vocabulary leads to high-dimensional vectors.**

Document D1	*The child makes the dog happy* the: 2, dog: 1, makes: 1, child: 1, happy: 1
Document D2	*The dog makes the child happy* the: 2, child: 1, makes: 1, dog: 1, happy: 1

↓

	child	dog	happy	makes	the	BoW Vector representations
D1	1	1	1	1	2	[1,1,1,1,2]
D2	1	1	1	1	2	[1,1,1,1,2]

TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

A REFINEMENT OF BOW THAT ADJUSTS WORD FREQUENCIES BASED ON THEIR IMPORTANCE ACROSS ALL DOCUMENTS.

$$\boldsymbol{TF(t,d)} = \frac{(Number\ of\ occurrences\ of\ term\ t\ in\ document\ d)}{(Total\ number\ of\ terms\ in\ the\ document\ d)}$$

$$\boldsymbol{IDF(t,D)} = \log_e \frac{(Total\ number\ of\ documents\ in\ the\ corpus)}{(Number\ of\ documents\ with\ term\ t\ in\ them)}$$

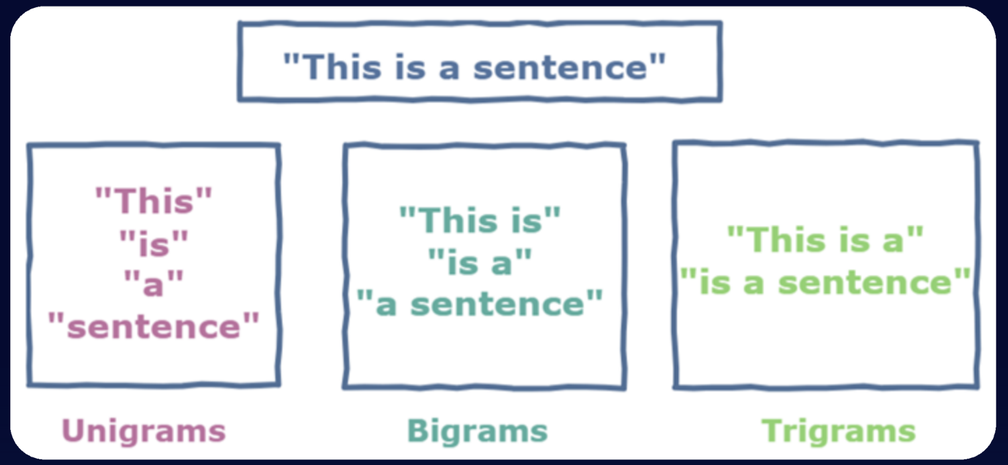
$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

COMPARISON

Specification	Bag of Words (BoW)	TF-IDF
Focus	Word frequency in a single document	Frequency of words adjusted by document frequency
Weighting	Raw word counts	Weighted by term frequency and inverse document frequency
Complexity	Lower computational complexity	Higher computational complexity (due to IDF calculation)
Feature Space	Larger, includes all words in the dataset	Smaller, excludes common words across documents
Sensitivity	Less sensitive to word importance across documents	Sensitive to word importance based on document frequency
Handling of Common Words	No adjustment for common words	Penalizes common words, emphasizes rare ones
Use Case	Basic text classification	Advanced text classification, information retrieval

N-GRAM

- Types of n-grams
  - Unigrams are individual words, bigrams are pairs of consecutive words, and trigrams are groups of three consecutive words.
- Example: "In the sentence 'The weather is beautiful'
  - Unigrams: ['The', 'weather', 'is', 'beautiful']
  - Bigrams: ['The weather', 'weather is', 'is beautiful']
  - Trigrams: ['The weather is', 'weather is beautiful']"

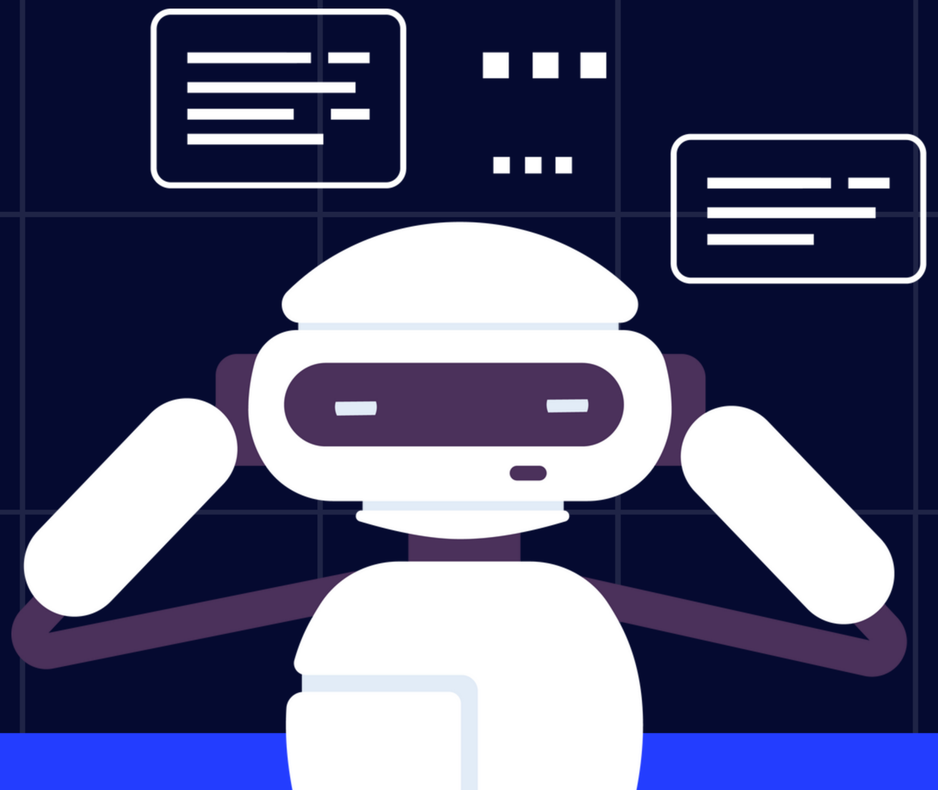


- Why use n-grams?
  - N-grams help us understand context.
    - By considering multiple words together, we get a clearer sense of meaning in text.
  - Example: "For sentiment analysis, a bigram like 'not good' is more informative than the separate words 'not' and 'good'."

USE CASES

- SEARCH ENGINES & INFORMATION RETRIEVAL
- TEXT ANALYTICS & CLASSIFICATION TASKS

Q&A and Discussion



Quizz Time


CODE TIME !!

Thank you

**

@Mo7amed3twan