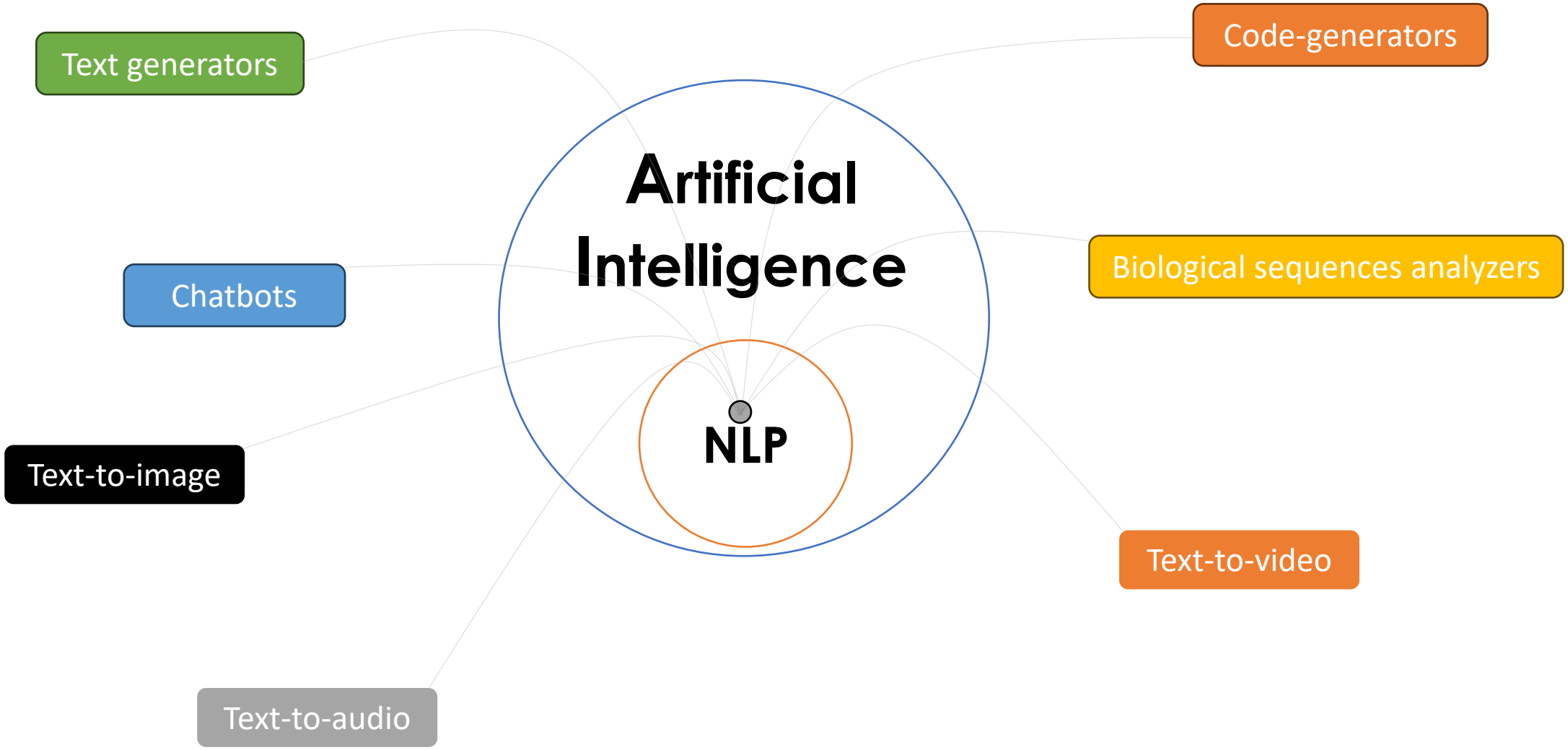
CHAPTER VI

Introduction to Natural Language Processing

PLAN

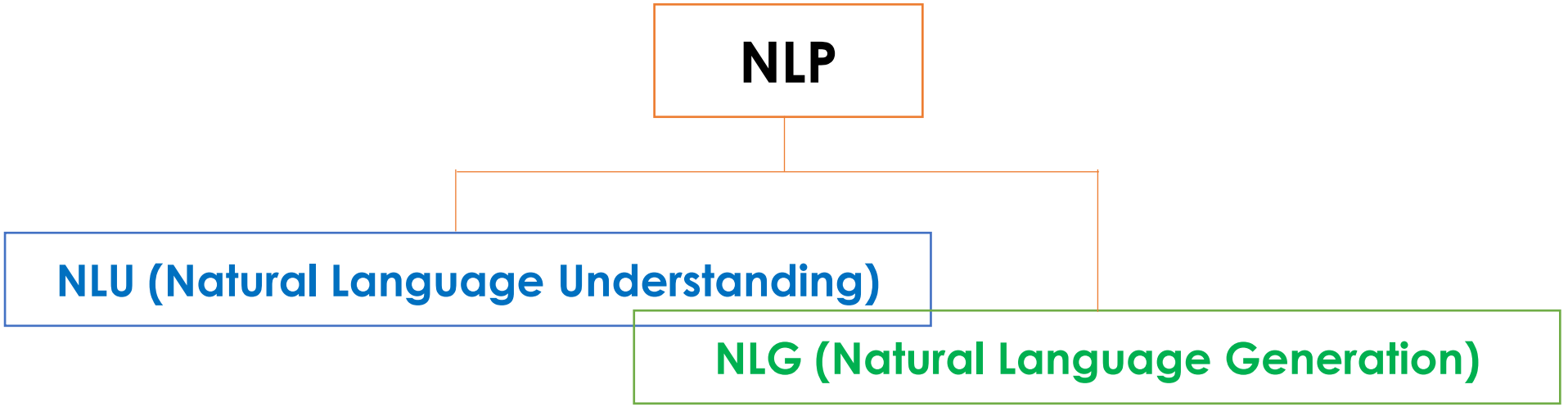
- What is NLP**?**
- NLP Applications
- How does NLP work**?**
- NLP techniques
- Libraries and Frameworks for NLP
- Python examples

NLP: A fast-growing research field in AI

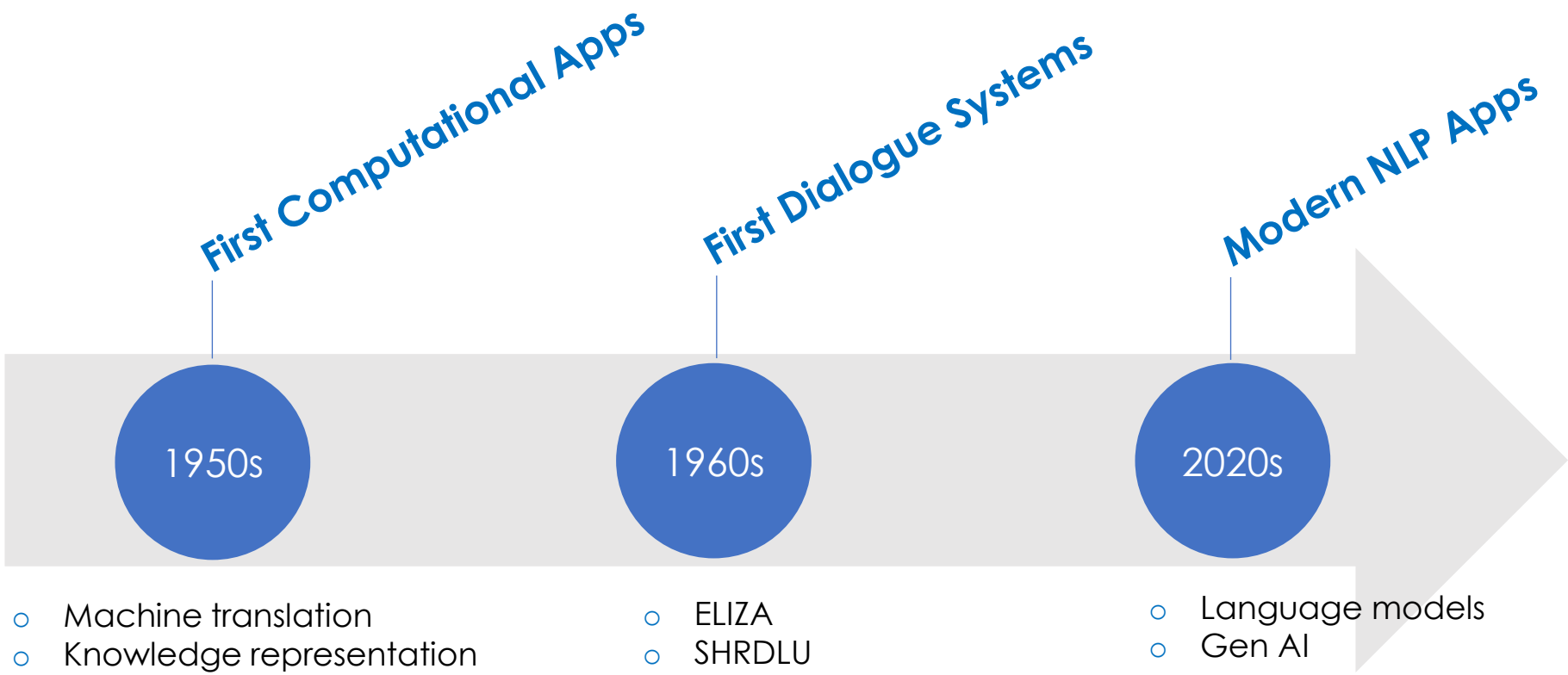


What is NLP ?

How to program computers to **analyze** the meanings of input text and **generate** meaningful, expressive output.



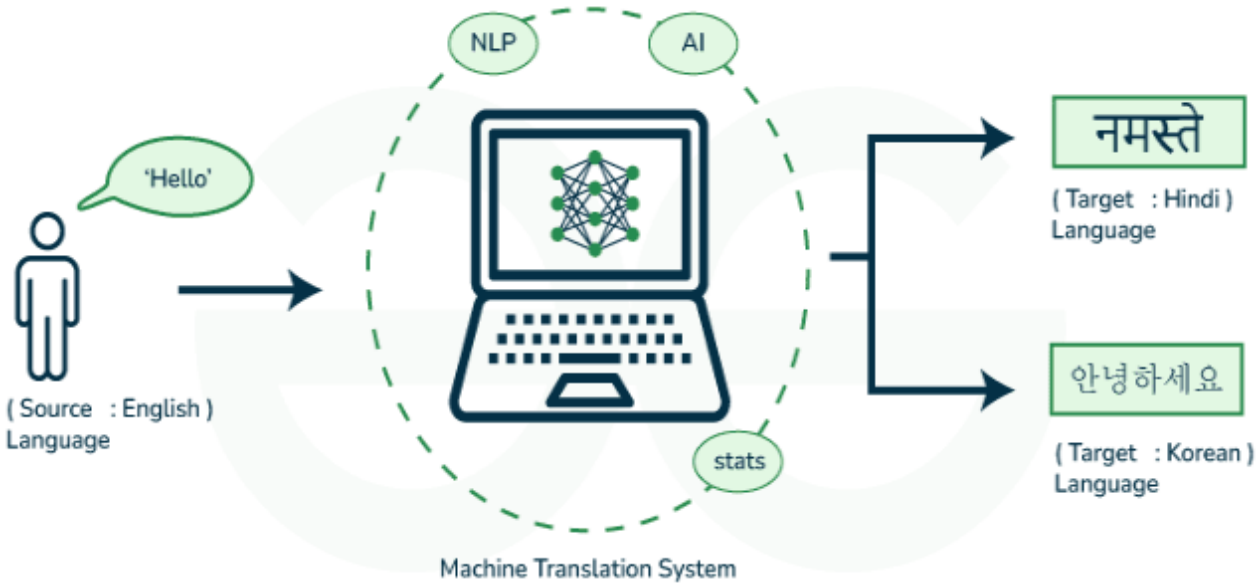
The early days of NLP



Main NLP Applications

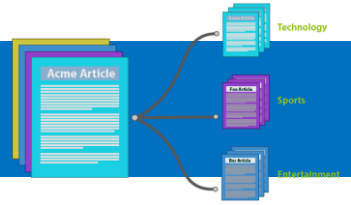
1. Machine Translation

- **Rule-based**
  (Grammar rules and dictionaries)
- **Statistical**
  (Examine extensive human translations)
- **Neural**
  (Training on Source-Target language dataset)
- **Hybrid**
  (Use of multiple machine translation models)



Main NLP Applications

2. Text Classification



- **Document classification**

  (Document categorization: Techno, Sport, Art,..)

- **Sentiment analysis**

  (Classifying emotional quality)

- **Toxicity classification**

  (Detecting threats, insults, hatred towards entities)

- **Spam detection**

  (Classify emails as either spam or not)

- **Hadith authentication**

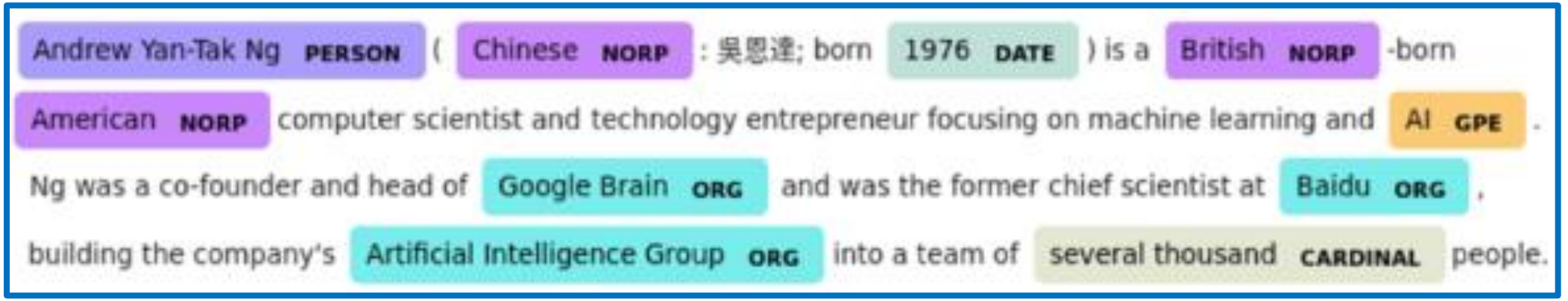
  (Verify originality of Prophetic Hadiths)

- **Misinformation and Fake news detection,…**

Main NLP Applications

3. Named Entity Recognition

Extract entities in a piece of text into predefined categories such as personal **names**, **organizations**, **locations**, and **quantities**.



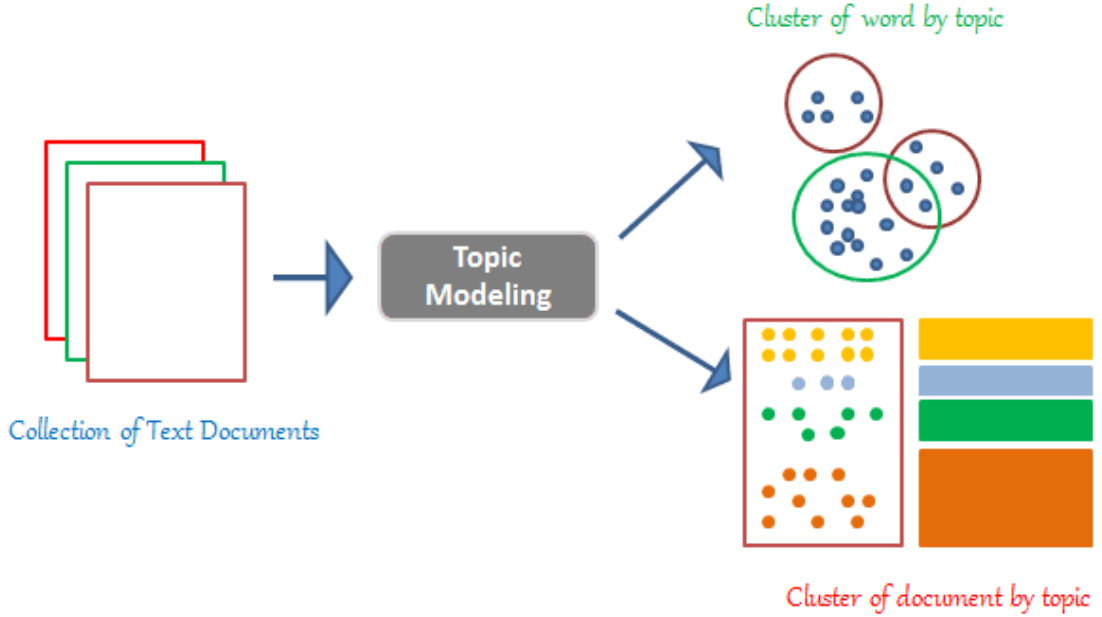
Main NLP Applications

4. Topic Modeling

Unsupervised text mining task that takes a corpus of documents and discovers abstract topics within that corpus.

Techniques:

- Latent Semantic Analysis (LSA)
- Latent Dirichlet Allocation (LDA)
- LDA2Vec
- BERTopic



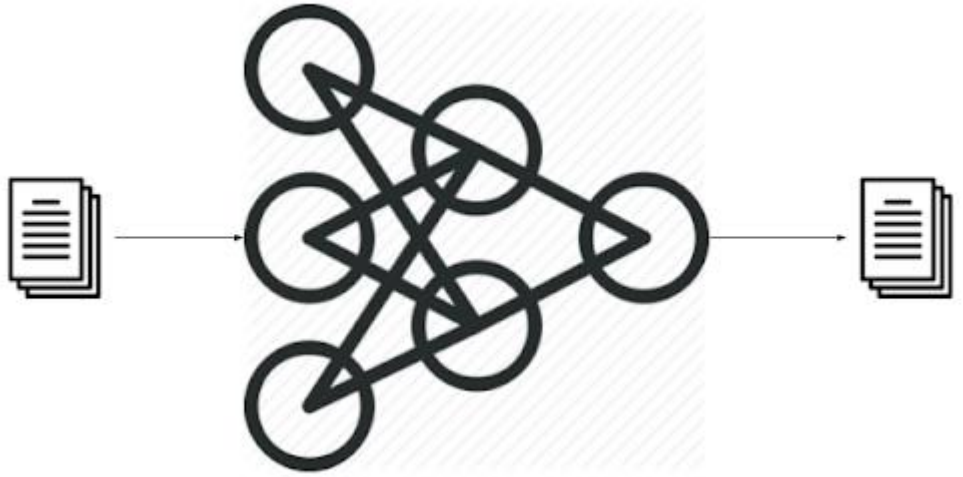
Main NLP Applications

5. Text Generation

Automatically produces text that is similar to human-written text (such as: Tweets, Blogs, Essays, Computer code,..): LSTM-RNN, BERT, BARD, ChatGPT,...

Variations:

- **Autocomplete:** predicts what word comes next
- **Chatbots:** automate one side of a conversation
  - Questions & Answers database
  - Conversation generation



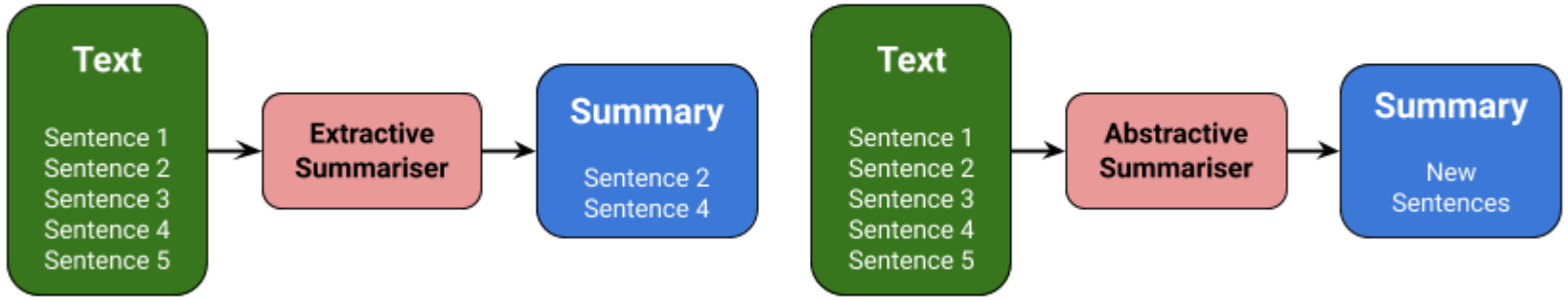
Main NLP Applications

6. Text Summarization

Shortening text to highlight the most relevant information

Variations:

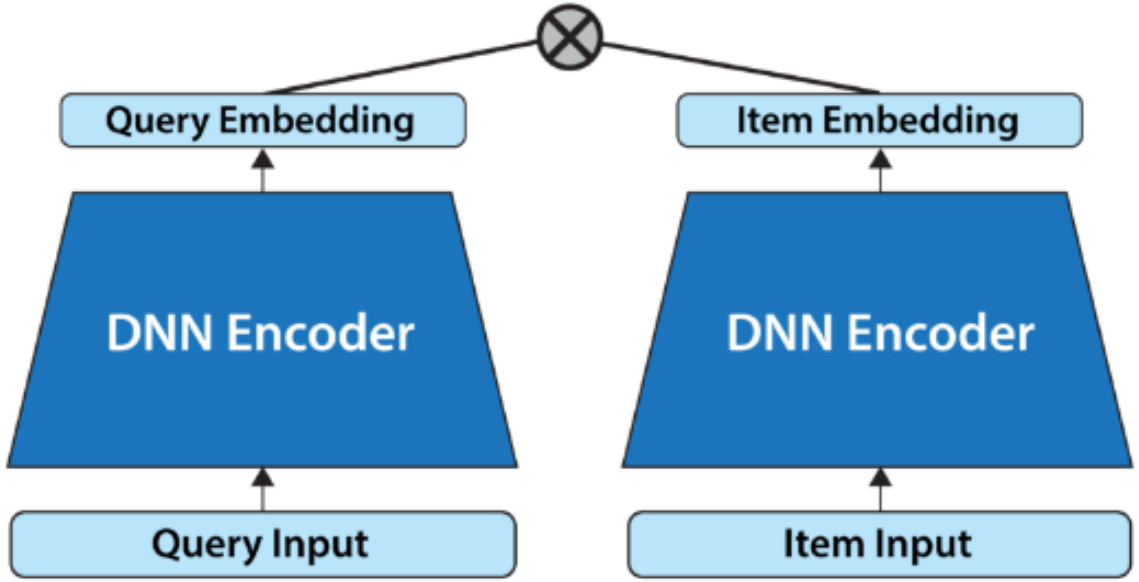
- **Extraction:** extracting the most important sentences from a long text and combining these to form a summary
- **Abstraction:** writing the abstract that includes words and sentences that are not present in the original text



Main NLP Applications

7. Information Retrieval

Finds (indexing and matching) the documents that are most relevant to a query.



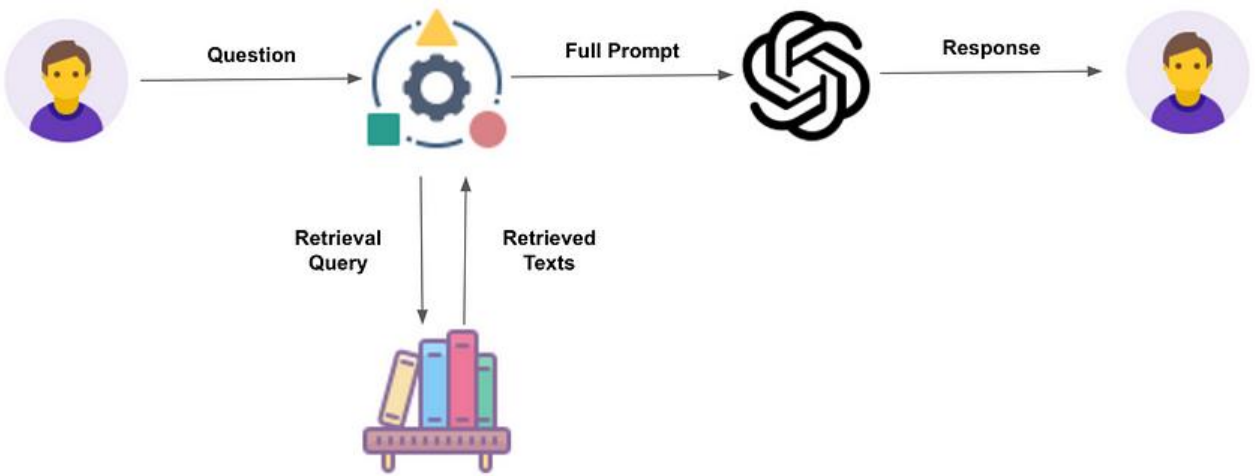
- **Indexing:** using a vector space
- **Matching:** using similarity score

Main NLP Applications

8. Question/Answering

Answering questions asked by humans in a natural language

- **Multiple choice:** question problem is composed of a question and a set of possible answers
- **Open-domain:** the model provides answers to questions in natural language without any options provided



Main NLP Applications

9. Other NLP apps

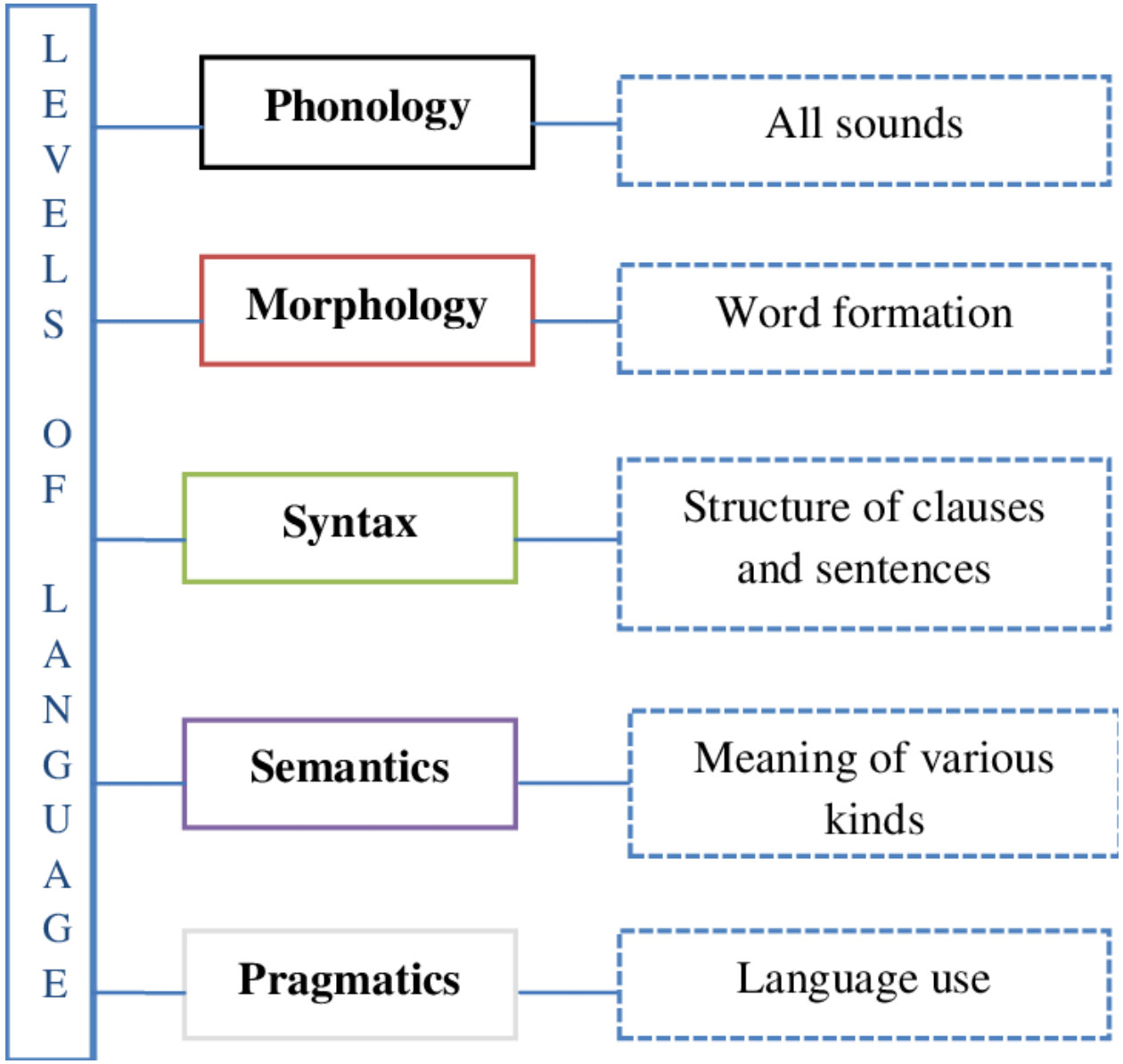
- **Grammatical error correction:** encode grammatical rules to correct the grammar within text.
- **Part-of-Speech Tagging:** classifying words in a text according to their grammatical categories (such as noun, verb, and adjective).
- **Language modeling:** building models that predict the probability of a sequence of words.
- **Speech recognition:** transform spoken language into a machine-readable format.

Main NLP Applications

9. Other NLP apps

- **Grammatical error correction:** encode grammatical rules to correct the grammar within text.
- **Part-of-Speech Tagging:** classifying words in a text according to their grammatical categories (such as noun, verb, and adjective).
- **Language modeling:** building models that predict the probability of a sequence of words.
- **Speech recognition:** transform spoken language into a machine-readable format.

NLP Processing levels



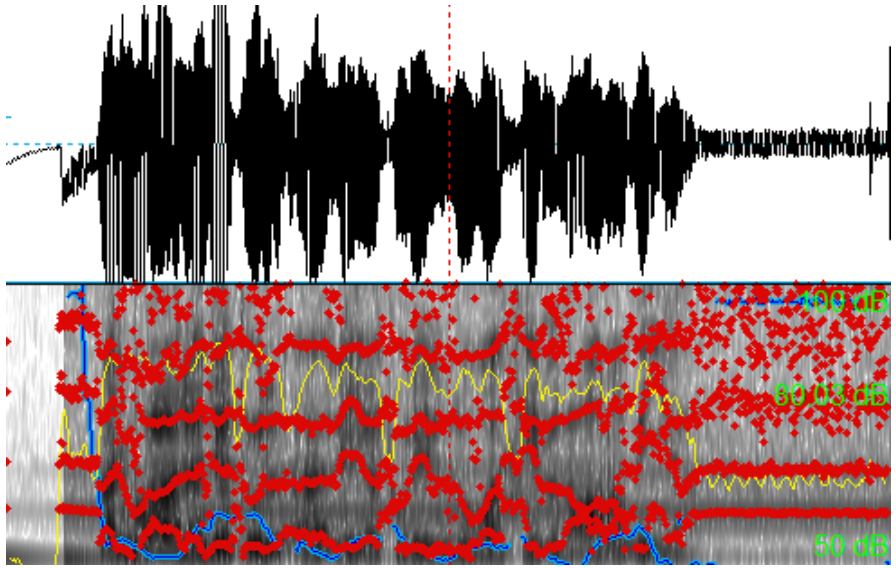
NLP Processing levels

LEVELS OF LANGUAGE

- **Phonology** — All sounds
- **Morphology** — Word formation
- **Syntax** — Structure of clauses and sentences
- **Semantics** — Meaning of various kinds
- **Pragmatics** — Language use

- Phoneme detection
- Prosody identification
- Transitions

العربية | اللغة | الآلية | المعالجة



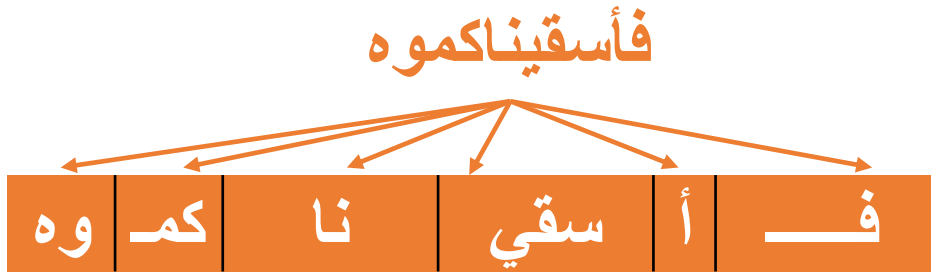
al-muʿālağatu al-ʾālīatu liluġati al-ʿārabīa

NLP Processing levels

LEVELS OF LANGUAGE

Phonology	All sounds
Morphology	Word formation
Syntax	Structure of clauses and sentences
Semantics	Meaning of various kinds
Pragmatics	Language use

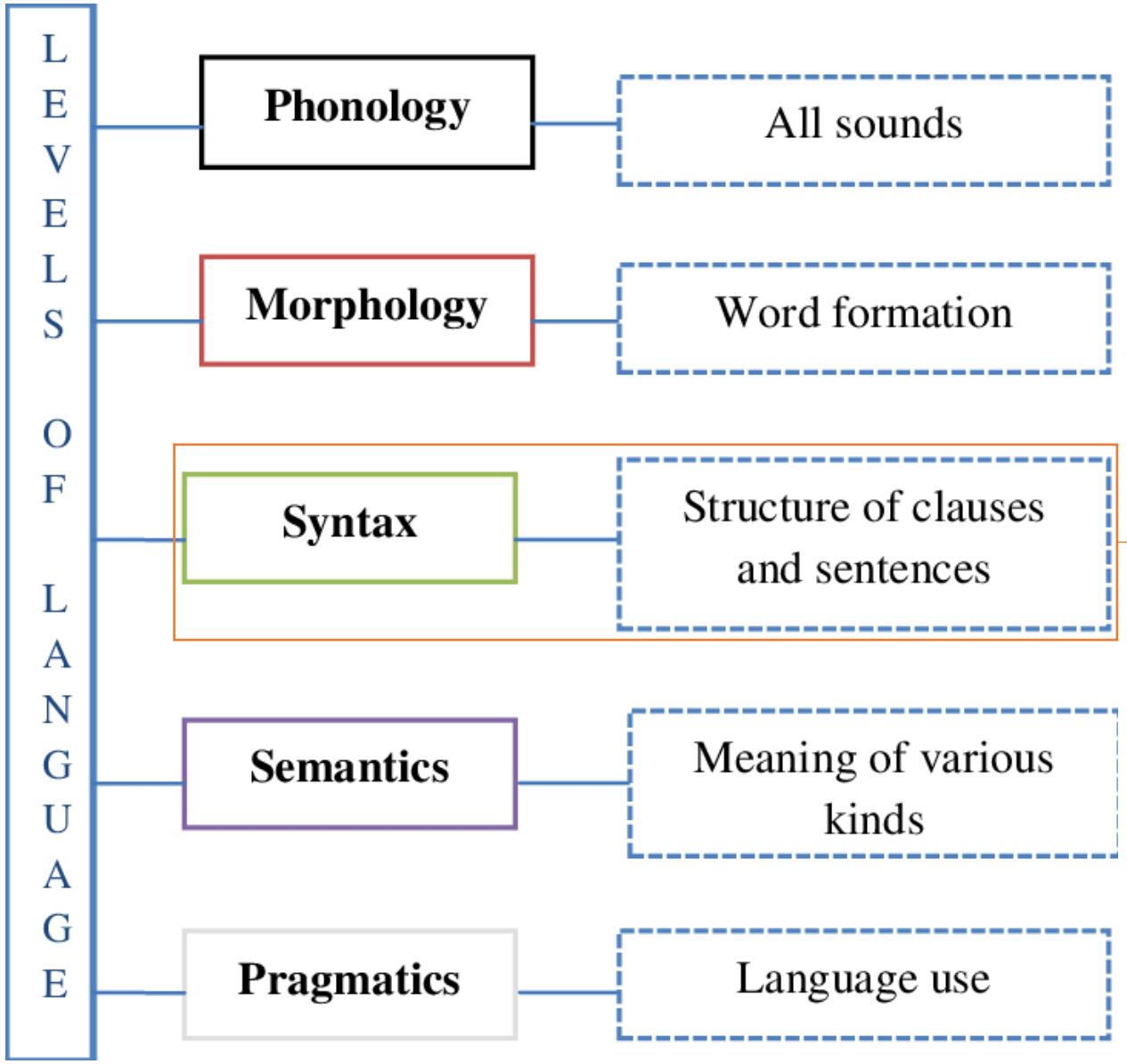
- Morphemes
- Tokens

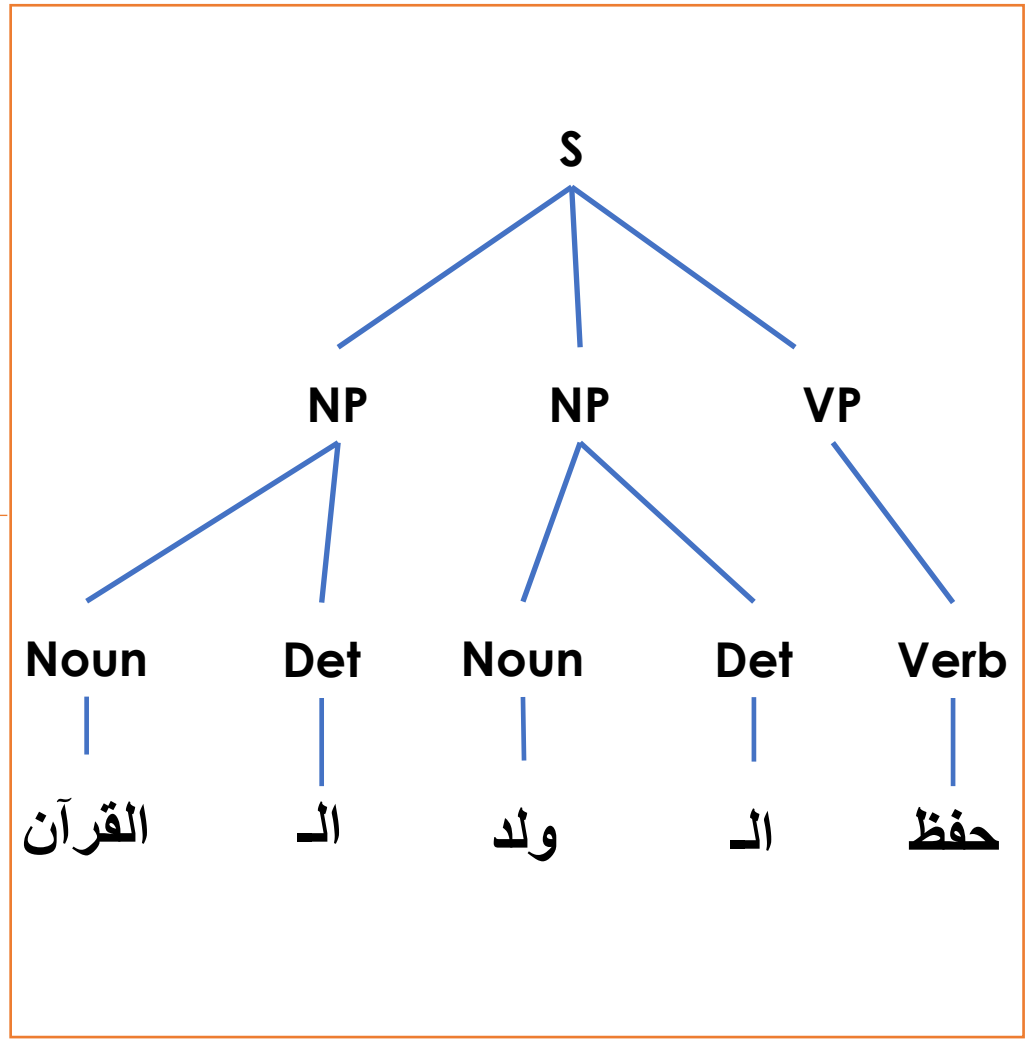


يذهب محمد إلى المسجد كل يوم

يذهب، محمد، إلى، المسجد، كل، يوم

NLP Processing levels

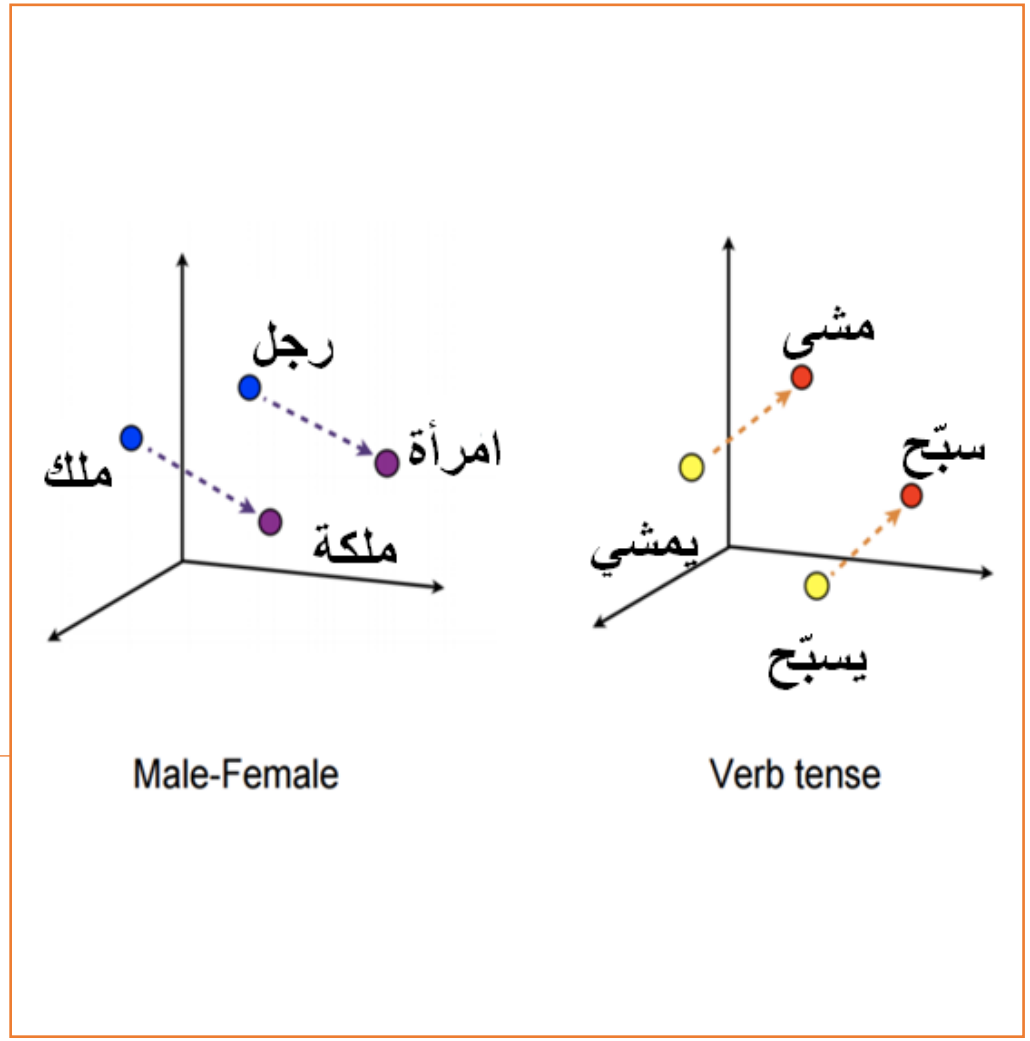




NLP Processing levels

LEVELS OF LANGUAGE

Phonology	All sounds
Morphology	Word formation
Syntax	Structure of clauses and sentences
Semantics	Meaning of various kinds
Pragmatics	Language use

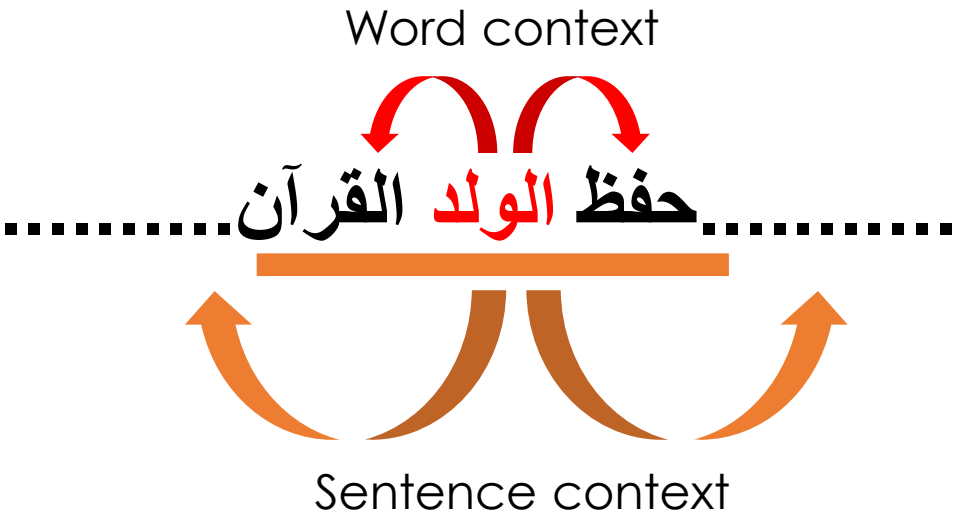


NLP Processing levels

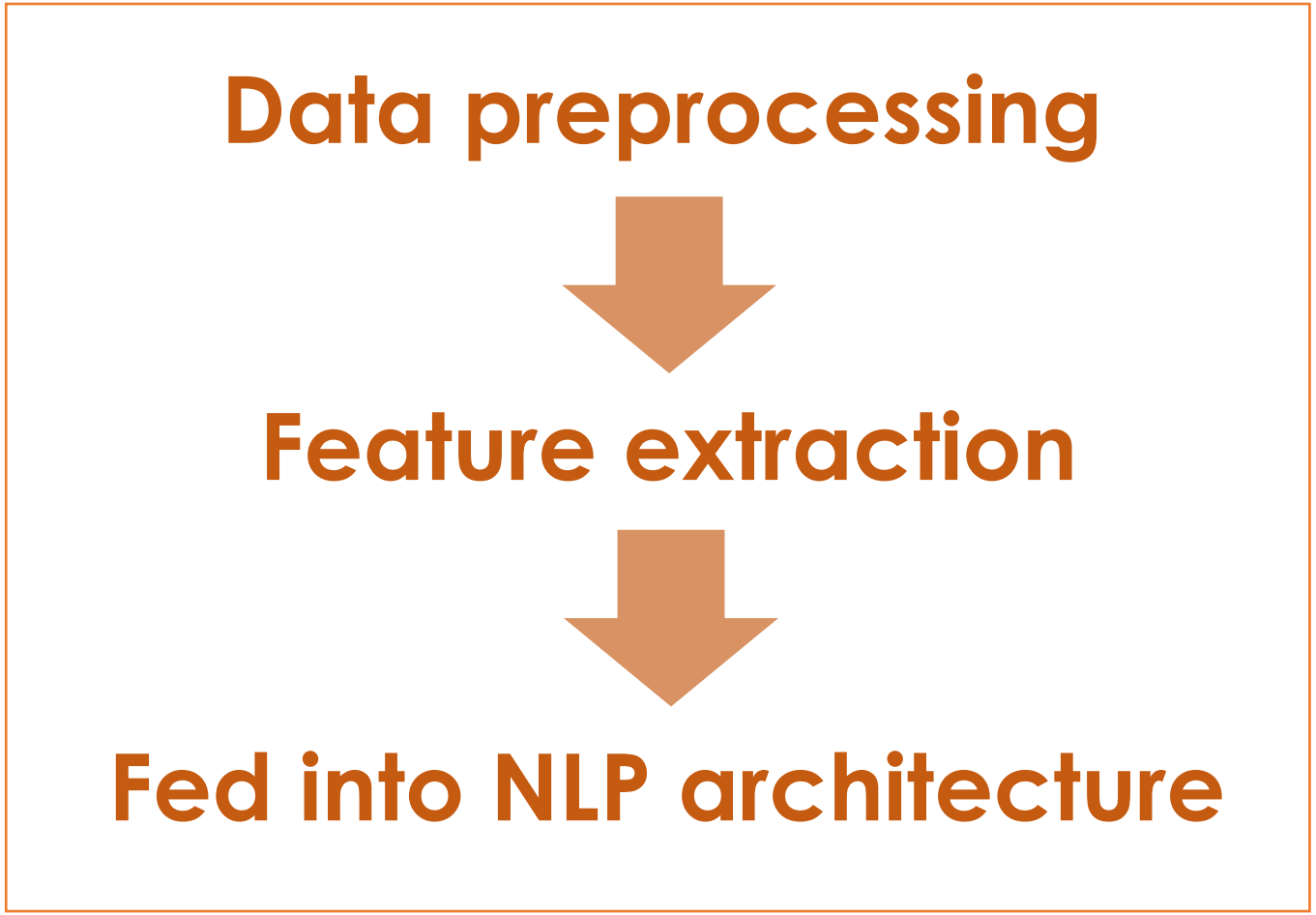
LEVELS OF LANGUAGE

Phonology	All sounds
Morphology	Word formation
Syntax	Structure of clauses and sentences
Semantics	Meaning of various kinds
Pragmatics	Language use

- Meaning in the context



How does NLP work?



How does NLP work?

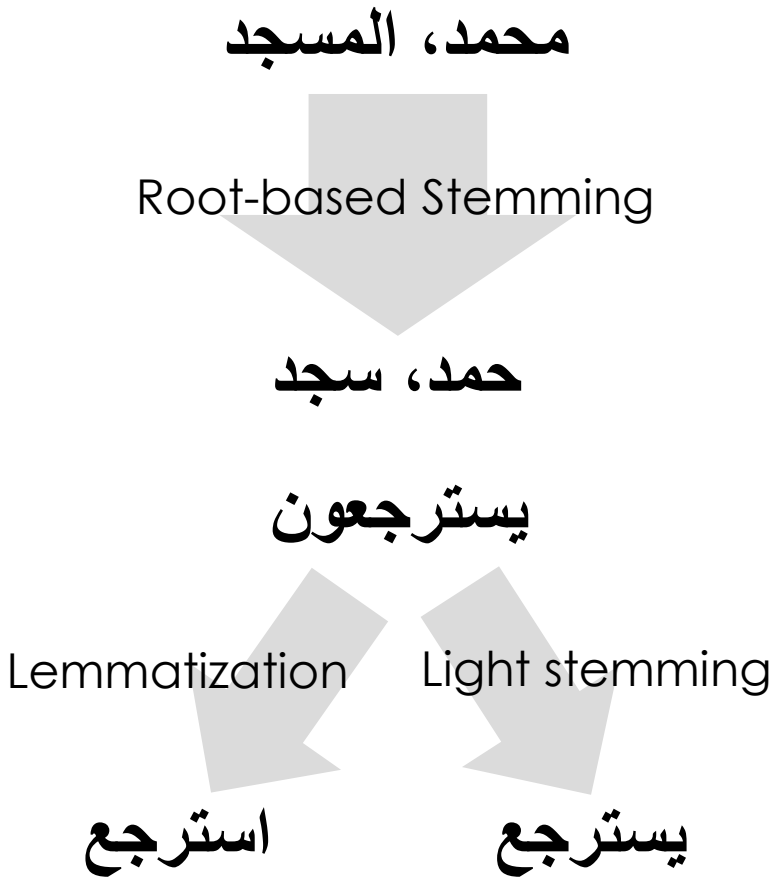
1. Data preprocessing

- **Stemming and Lemmatization:** converting words to their base forms.
- **Sentence segmentation:** breaks a large piece of text into meaningful sentence units.
- **Stop word removal:** remove words that don't add much information to the text.
- **Tokenization:** splits text into individual words.

How does NLP work?

1. Data preprocessing

- **Stemming and Lemmatization:**



How does NLP work?

1. Data preprocessing

- **Stop word removal:**

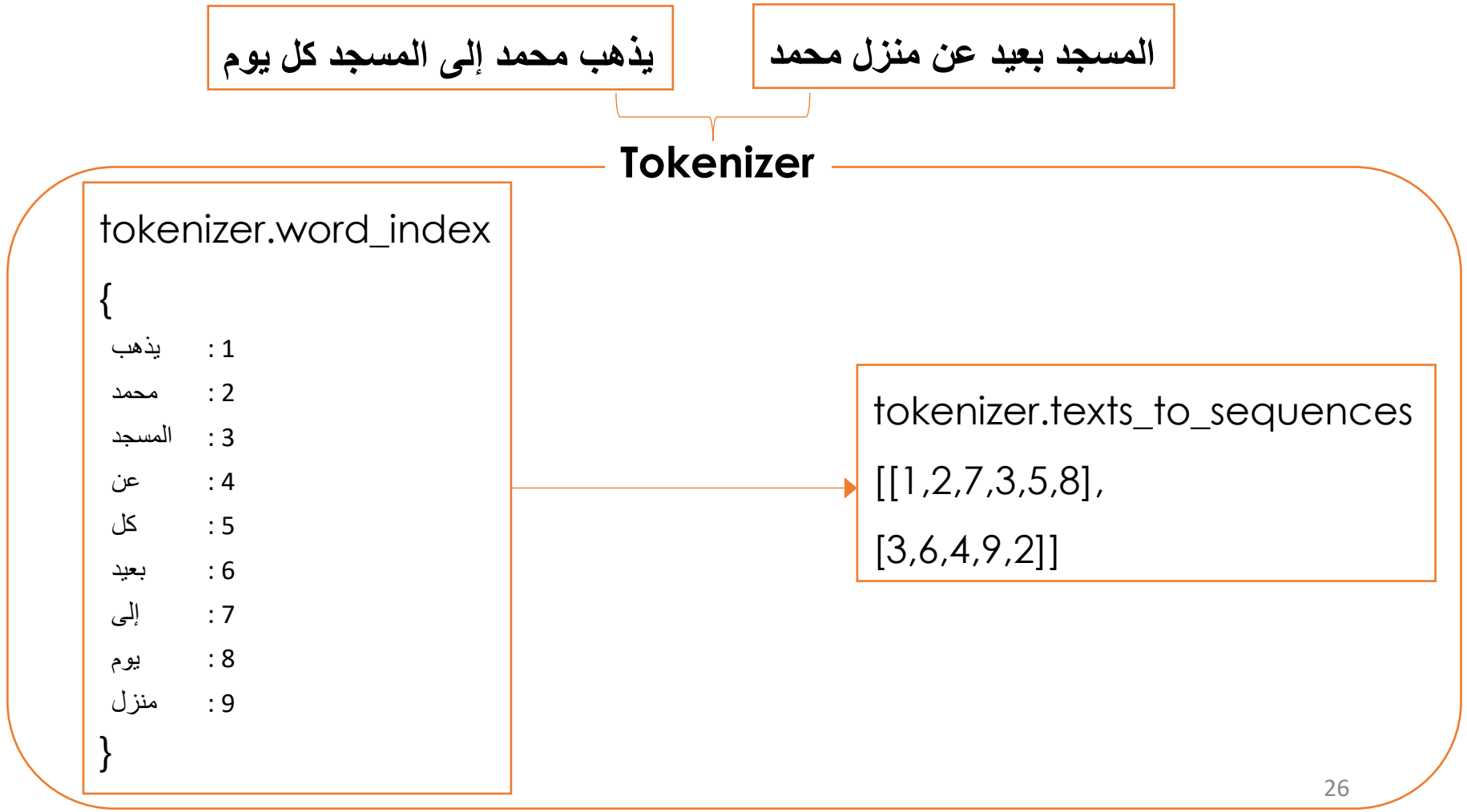
يذهب محمد إلى المسجد كل يوم

يذهب، محمد، المسجد، يوم

How does NLP work?

1. Data preprocessing

- **Tokenization:**



How does NLP work?

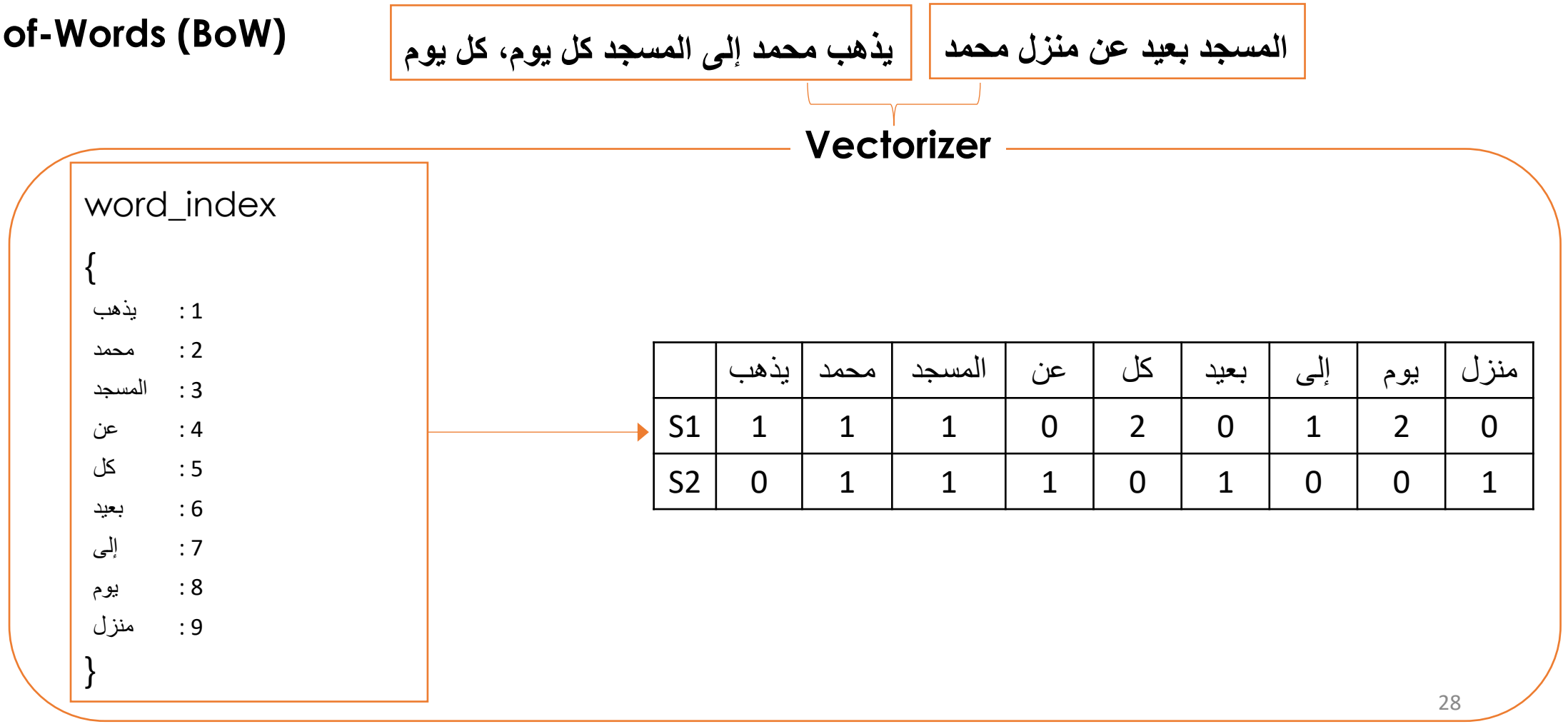
2. Feature extraction

- **Bag-of-Words**
- **One-Hot-Encoding**
- **N-Grams**
- **TF-IDF**
- **Word Embeddings**
  - **Word2Vec (CBoW, Skip-Gram)**
  - **GLoVE**

How does NLP work?

2. Feature extraction

- **Bag-of-Words (BoW)**



	يذهب	محمد	المسجد	عن	كل	بعيد	إلى	يوم	منزل
S1	1	1	1	0	2	0	1	2	0
S2	0	1	1	1	0	1	0	0	1

How does NLP work?

2. Feature extraction

- **One-Hot-Encoding**

يذهب محمد إلى المسجد كل يوم

المسجد بعيد عن منزل محمد

Vectorizer

word_index

{

يذهب : 1

محمد : 2

المسجد : 3

عن : 4

كل : 5

بعيد : 6

إلى : 7

يوم : 8

منزل : 9

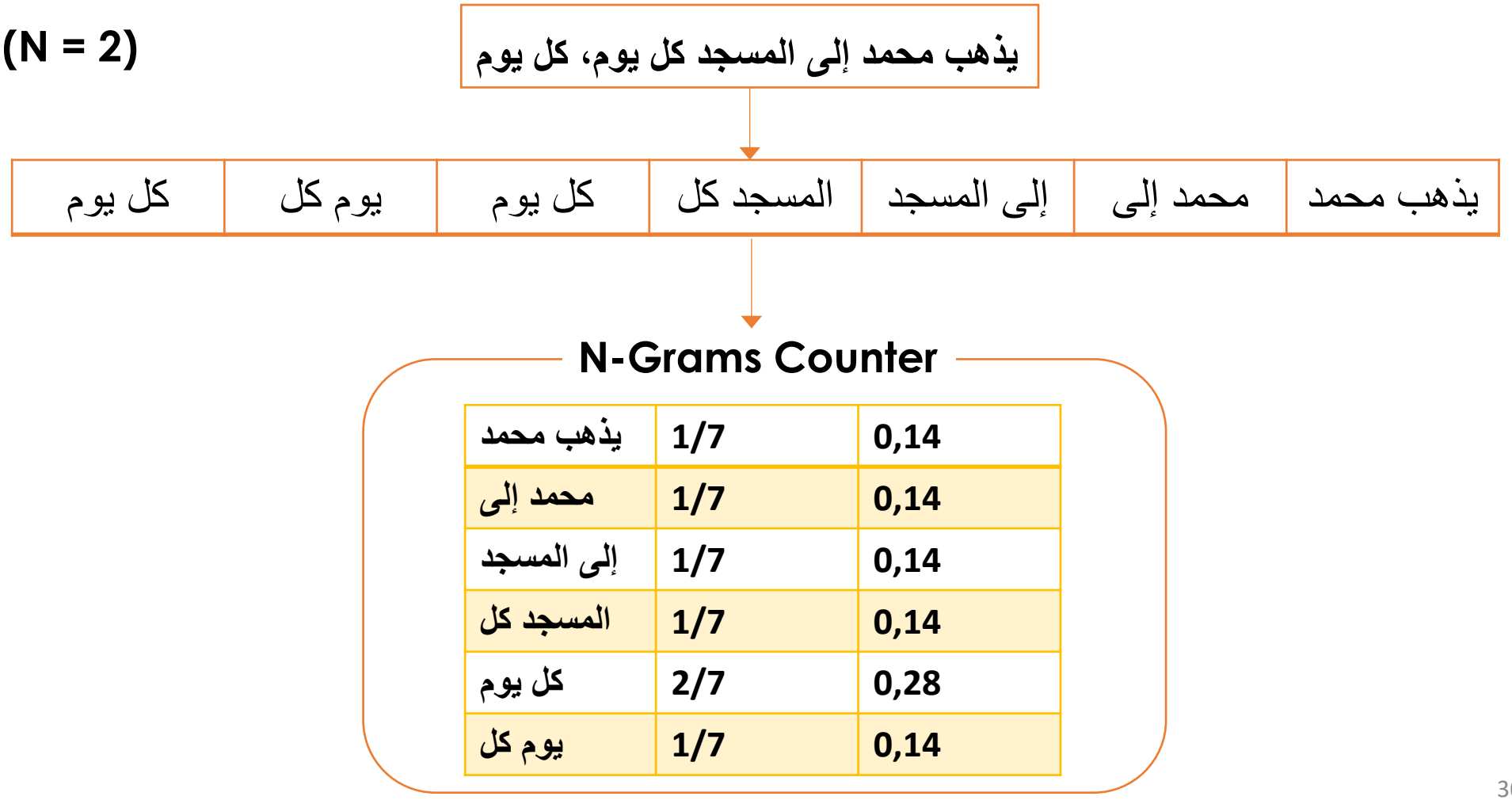
}

	يذهب	محمد	المسجد	عن	كل	بعيد	إلى	يوم	منزل
يذهب	1	0	0	0	0	0	0	0	0
محمد	0	1	0	0	0	0	0	0	0
المسجد	0	0	1	0	0	0	0	0	0
عن	0	0	0	1	0	0	0	0	0
كل	0	0	0	0	1	0	0	0	0
بعيد	0	0	0	0	0	1	0	0	0
إلى	0	0	0	0	0	0	1	0	0
يوم	0	0	0	0	0	0	0	1	0
منزل	0	0	0	0	0	0	0	0	1

How does NLP work?

2. Feature extraction

- **N-Grams (N = 2)**



يذهب محمد	1/7	0,14
محمد إلى	1/7	0,14
إلى المسجد	1/7	0,14
المسجد كل	1/7	0,14
كل يوم	2/7	0,28
يوم كل	1/7	0,14

How does NLP work?

2. Feature extraction

- **TF-IDF:**
  - Weights each word by its importance
  - TF (Term Frequency) = *Number of occurrences of the word in document / Number of words in document*
  - IDF (Inverse Docment Frequency) = *log(number of documents in the corpus / number of documents that include the word)*

D1	هذا أمر جيد وممتاز
D2	هذا أمر سيء للغاية

TF-IDF Vectorizer

TF

	هذا	أمر	جيد	سيء	و	للغاية	ممتاز
D1	1/5	1/5	1/5	0	1/5	0	1/5
D2	1/4	1/4	0	1/4	0	1/4	0

IDF

هذا	أمر	جيد	سيء	و	للغاية	ممتاز
log(2/2)	log(2/2)	log(2/1)	log(2/1)	log(2/1)	log(2/1)	log(2/1)

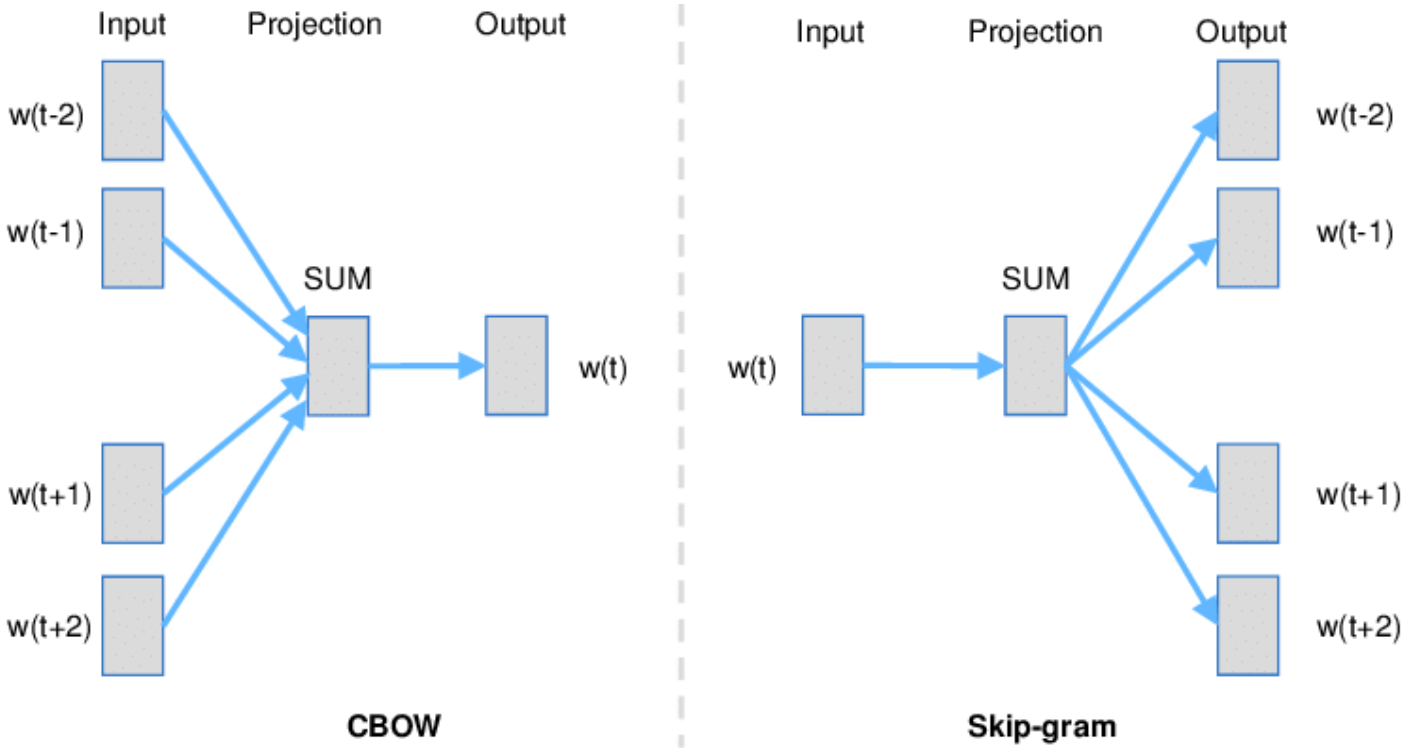
TF-IDF Features

	هذا	أمر	جيد	سيء	و	للغاية	ممتاز
D1	0	0	0,060	0	0,060	0	0,060
D2	0	0	0	0,075	0	0,075	0

How does NLP work?

2. Feature extraction

- **Word embeddings (Word2Vec):**



يذهب محمد إلى المسجد كل يوم

Word2Vec

يذهب	[0.2, 0.3, -0.1, 0.5, ...]
محمد	[0.1, -0.4, 0.6, -0.2, ...]
إلى	[0.3, -0.2, 0.4, 0.1, ...]
المسجد	[-0.5, 0.2, 0.3, -0.1, ...]
كل	[0.4, 0.1, -0.3, 0.2, ...]
يوم	[0.6, -0.3, 0.2, 0.4, ...]

NLP techniques

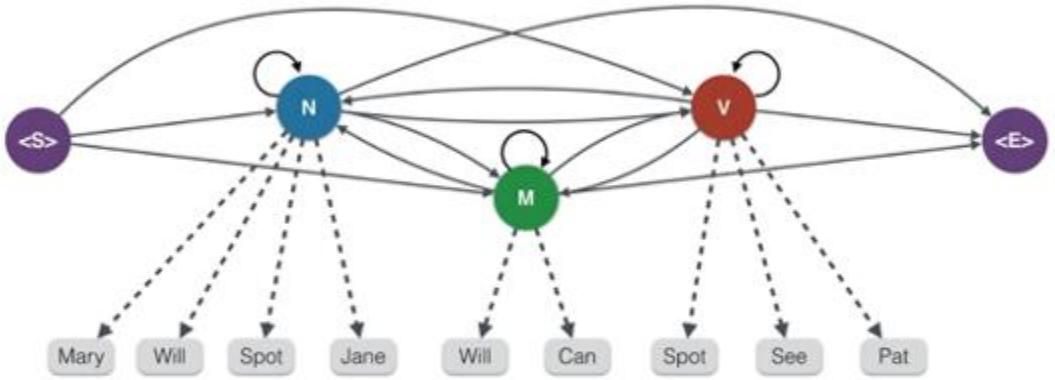
- **Traditional machine learning techniques**
  - Logistic regression
  - Naive Bayes
  - Decision trees
  - LDA/LSA
  - Hidden Markov Models (HMM)
- **Deep learning techniques**
  - Convolutional Neural Networks (CNN)
  - Recurrent Neural Networks (RNN)
  - Autoencoders
  - Seq2Seq models
  - Transformers

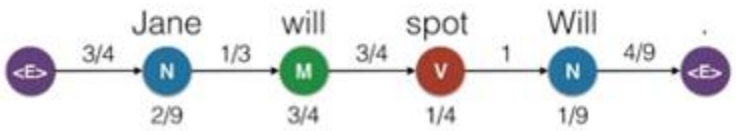
NLP techniques

POS-Tagging with HMM

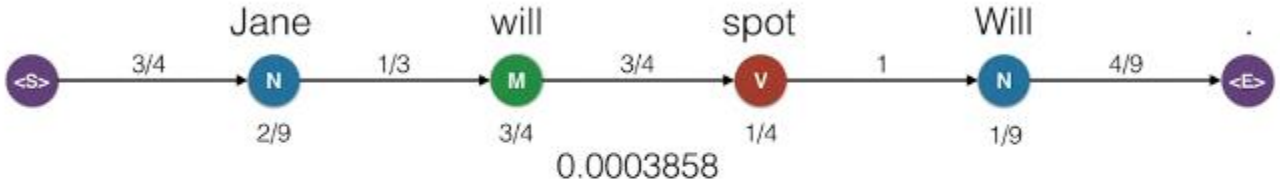
S = Jane will spot Will

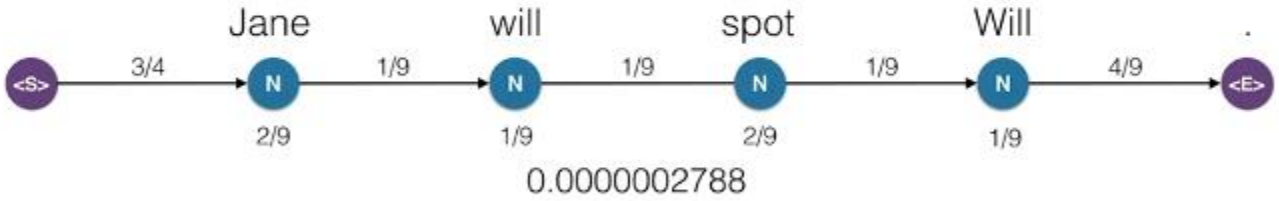
What will be the most likely assignment for each word**?**





High probability sequence

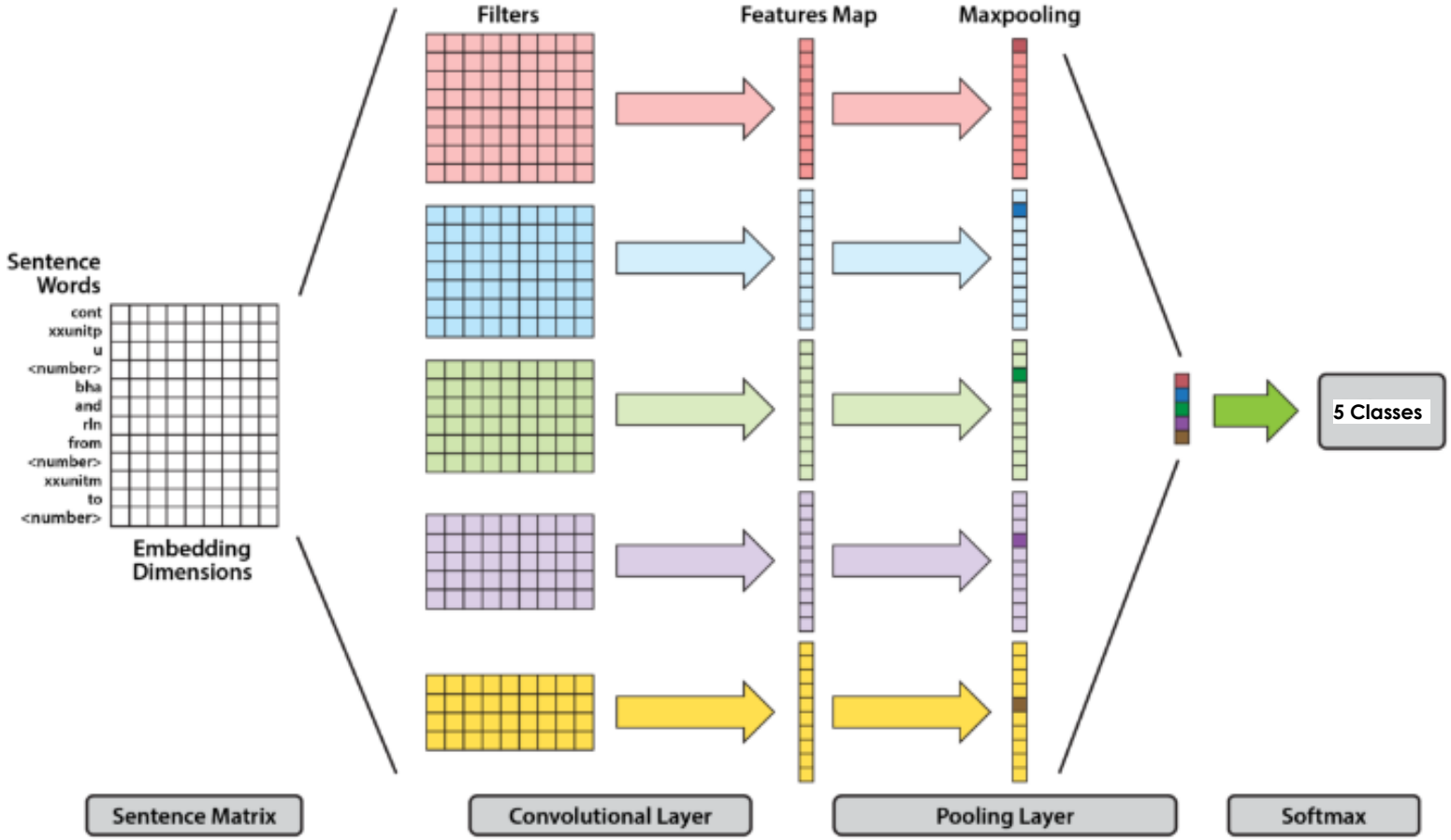




NLP techniques

CNN-Based text classfication

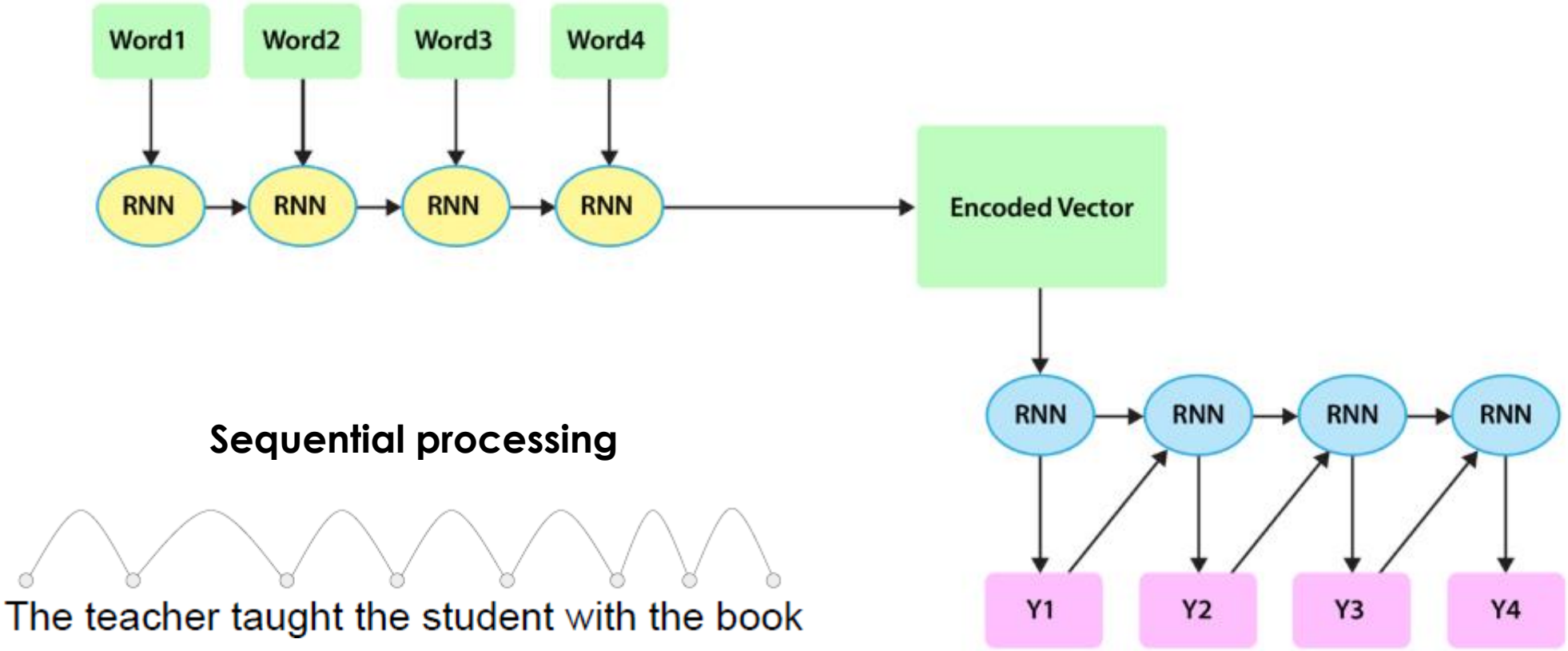
Given a sentence, a CNN uses convolutional layers to refine represenations of input words, before combining them to render a classification



NLP techniques

RNN-Based Seq2Seq model for Machine translation

Given a sentence, a RNN encodes the sequence and then iteratively generates a translation



NLP techniques

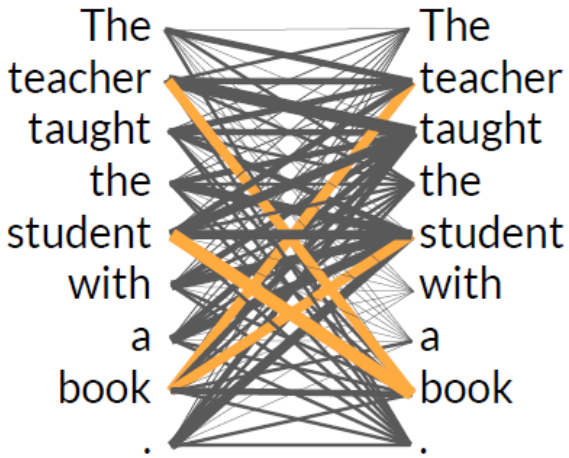
Transformer architecture

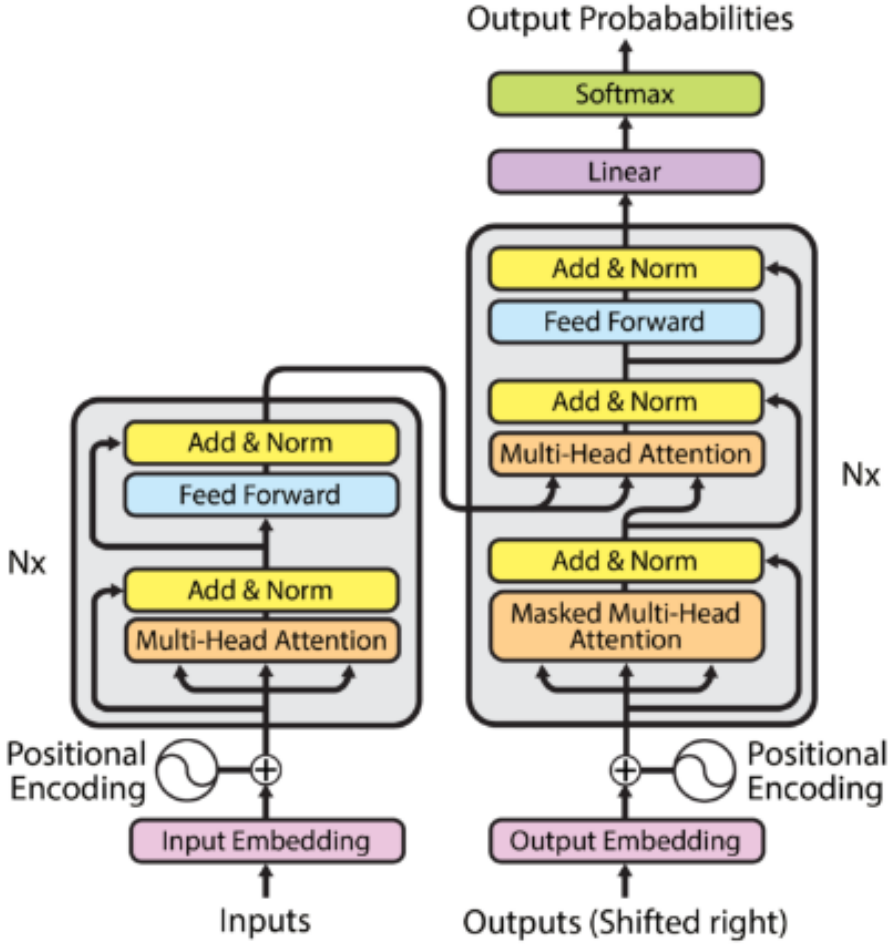
Relies entirely on a self-attention mechanism to draw global dependencies between input and output,

It is at the core of new language models:

- **Encoder only: BERT, ROBERTA**
- **Autoregressive (Decoder only): GPT, BLOOM**
- **Seq2Seq (Encoder-Decoder): T5, BART**

Parallel processing





NLP system evaluation

- Evaluating an NLP system is a critical process to ensure it meets its intended purpose and performs effectively.

- **Evaluation methods** depend on the specific tasks the system is designed for, such as text classification, machine translation,..

Evaluating a classification model

Confusion matrix

- Total number of examples: 1000
- Class A: 262 Class B: 237 Class C: 283 Class D: 218

Actual Label \ Predicted Label	A	B	C	D
A	205	10	1	46
B	6	199	0	32
C	9	17	223	34
D	21	8	3	186

$$Total\ accuracy = \frac{\sum correct}{\sum all} = \frac{813}{1000} = 0.813$$

Confusion matrix

Actual Label \ Predicted Label	Positive	Negative
Positive	38	17
Negative	3	42

True Positive (TP): Predicted positive matches actual positive

True Negative (TN): Predicted negative matches actual negative

False Positive (FP) ("Type I Error"): Predicted positive does not match actual negative

False Negative (FN) ("Type II Error"): Predicted negative does not match actual positive

Confusion matrix

- Total number of examples: 1000
- Class A: 262 Class B: 237 Class C: 283 Class D: 218

Predicted Label

Actual Label	A	B	C	D
A	205	10	1	46
B	6	199	0	32
C	9	17	223	34
D	21	8	3	186

True Positive (TP): Predicted positive matches actual positive

True Negative (TN): Predicted negative matches actual negative

False Positive (FP) ("Type I Error"): Predicted positive does not match actual negative

False Negative (FN) ("Type II Error"): Predicted negative does not match actual positive

Confusion matrix

- Total number of examples: 1000
- Class A: 262 Class B: 237 Class C: 283 Class D: 218

Predicted Label

Actual Label	A	B	C	D
A	205	10	1	46
B	6	199	0	32
C	9	17	223	34
D	21	8	3	186

Per class Accuracy:

$$\boldsymbol{Accuracy_B} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{199 + 728}{1000} = 0.927$$

Confusion matrix

- Total number of examples: 1000
- Class A: 262 Class B: 237 Class C: 283 Class D: 218

Actual Label \ Predicted Label	A	B	C	D
A	205	10	1	46
B	6	199	0	32
C	9	17	223	34
D	21	8	3	186

True Positive Rate (Sensitivity, Recall, Hit Rate):

$$\boldsymbol{TPR_B} = \frac{TP}{TP + FN} = \frac{199}{199 + 38} = 0.840$$

Confusion matrix

- Total number of examples: 1000
- Class A: 262 Class B: 237 Class C: 283 Class D: 218

Actual Label \ Predicted Label	A	B	C	D
A	205	10	1	46
B	6	199	0	32
C	9	17	223	34
D	21	8	3	186

True Negative Rate (Specificity, Selectivity):

$$\boldsymbol{TNR_B} = \frac{TN}{TN + FP} = \frac{728}{728 + 35} = 0.954$$

Confusion matrix

- Total number of examples: 1000
- Class A: 262 Class B: 237 Class C: 283 Class D: 218

Predicted Label

Actual Label	A	B	C	D
A	205	10	1	46
B	6	199	0	32
C	9	17	223	34
D	21	8	3	186

Positive Predictive Value (Precision):

$$\boldsymbol{PPV_B} = \frac{TP}{TP + FP} = \frac{199}{199 + 35} = 0.850$$

Confusion matrix

- Total number of examples: 1000
- Class A: 262 Class B: 237 Class C: 283 Class D: 218

Predicted Label

Actual Label	A	B	C	D
A	205	10	1	46
B	6	199	0	32
C	9	17	223	34
D	21	8	3	186

F1 score:

$$\boldsymbol{F1_B} = 2 \times \frac{PPV \times TPR}{PPV + TPR} = 2 \times \frac{0.850 \times 0.840}{0.850 + 0.840} = 0.845 = \frac{2 \times TP}{2 \times TP + FP + FN} = \frac{2 \times 199}{2 \times 199 + 35 + 38} = 0.845$$

Confusion matrix

- Total number of examples: 1000
- Class A: 262 Class B: 237 Class C: 283 Class D: 218

Predicted Label

Actual Label	A	B	C	D
A	205	10	1	46
B	6	199	0	32
C	9	17	223	34
D	21	8	3	186

$$\boldsymbol{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\boldsymbol{F1\ Score} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

- Per-Class Accuracy:
- Per-Class F1 Score:
- Total Accuracy:
- F1 Score Average:

Confusion matrix

- Total number of examples: 1000
- Class A: 262 Class B: 237 Class C: 283 Class D: 218

		Predicted Label			
		A	**B**	**C**	**D**
Actual Label	**A**	205	10	1	46
	B	6	199	0	32
	C	9	17	223	34
	D	21	8	3	186

- Per-Class Accuracy: 0.907 0.927 0.936 0.856
- Per-Class F1 Score: 0.815 0.845 0.875 0.721
- Total Accuracy: 0.813
- F1 Score Average: 0.818

Evaluating a text generation model

ROUGE

Recall-Oriented Understudy for Gisting Evaluation

ROUGE calculates the intersection of the common n-grams between the auto-generated text (candidate) and the human generated text (reference):

- ROUGE-N
- ROUGE-L

Example:

Reference: التمر هو ثمرة أشجار النخيل

Candidate: أشجار النخيل تثمر التمر

ROUGE

Recall-Oriented Understudy for Gisting Evaluation

ROUGE-N: measures the number of matching n-grams between the candidate and the reference

$$Recall = \frac{Number\ of\ common\ n-grams\ between\ candidate\ and\ reference}{Total\ number\ of\ n-grams\ in\ reference}$$

$$Precision = \frac{Number\ of\ common\ n-grams\ between\ candidate\ and\ reference}{Total\ number\ of\ n-grams\ in\ candidate}$$

$$F1-Score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

ROUGE-N	Reference	Candidate	Recall	Precision
ROUGE-1	التمر – هو – ثمرة – أشجار - النخيل	أشجار – النخيل – تثمر - التمر	3/5	3/4
ROUGE-2	التمر هو – هو ثمرة – ثمرة أشجار – أشجار النخيل	أشجار النخيل – النخيل تثمر – تثمر التمر	1/4	1/3

ROUGE

Recall-Oriented Understudy for Gisting Evaluation

ROUGE-L: measures the longest common subsequence (LCS) of words (not necessarily consecutive) between the candidate and the reference

$$Recall = \frac{Length\ of\ LCS}{Total\ number\ of\ 1-grams\ in\ reference}$$

$$Precision = \frac{Length\ of\ LCS}{Total\ number\ of\ 1-grams\ in\ candidate}$$

ROUGE-L	Reference	Candidate	Recall	Precision
	التمر هو ثمرة **أشجار النخيل**	**أشجار النخيل** تثمرالتمر	2/5	2/4