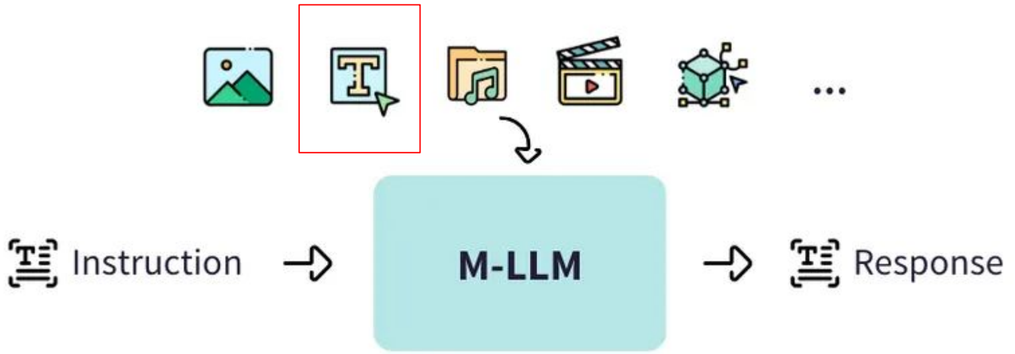
6CS030 Big Data 2024

Lecture 3 : Natural Language Processing (NLP)



deepseek

Gemini

ChatGPT

Claude

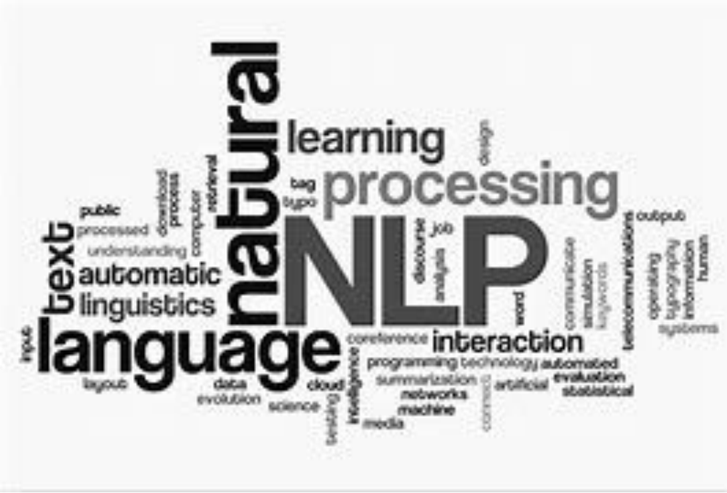
Qwen2.5

Agenda

- Natural Language Understanding
- Types of Linguistic Analysis
- Solving an NLP based task

What Is Language?

- **Language** : A structured system of communication using complex combinations of characters, words, and sentences.
- **Linguistics** : The systematic study of language.
- **NLP & Linguistics** : Understanding linguistic concepts is essential for effectively studying Natural Language Processing (NLP).



Phonemes

- Phonemes are the smallest units of sound in a language.
- They have no meaning individually but form meaningful words when combined.
- Standard English has **44** phonemes, made up of single or combined letters.
    - Phonemes are crucial in speech applications like:
    - Speech recognition
    - Speech-to-text transcription
    - Text-to-speech conversion

Consonant phonemes, with sample words		Vowel phonemes, with sample words	
1. /b/ - bat	13. /s/ - sun	1. /a/ - ant	13. /oi/ - coin
2. /k/ - cat	14. /t/ - tap	2. /e/ - egg	14. /ar/ - farm
3. /d/ - dog	15. /v/ - van	3. /i/ - in	15. /or/ - for
4. /f/ - fan	16. /w/ - wig	4. /o/ - on	16. /ur/ - hurt
5. /g/ - go	17. /y/ - yes	5. /u/ - up	17. /air/ - fair
6. /h/ - hen	18. /z/ - zip	6. /ai/ - rain	18. /ear/ - dear
7. /j/ - jet	19. /sh/ - shop	7. /ee/ - feet	19. /ure/[4] - sure
8. /l/ - leg	20. /ch/ - chip	8. /igh/ - night	20. /ə/ - corner (the 'schwa' - an unstressed vowel sound which is close to /u/)
9. /m/ - map	21. /th/ - thin	9. /oa/ - boat	
10. /n/ - net	22. /**th**/ - then	10. /**oo**/ - boot	
11. /p/ - pen	23. /ng/ - ring	11. /oo/ - look	
12. /r/ - rat	24. /zh/[3] - vision	12. /ow/ - cow	

Phonemes and Examples

Morphemes

UNBREAKABLE

UN + BREAK + ABLE

- A morpheme is the smallest unit of language that has a meaning.
- It is formed by a combination of phonemes. Not all morphemes are words, but all prefixes and suffixes are morphemes.
- For example, in the word “multimedia,” “multi-” is not a word but a prefix that changes the meaning when put together with “media.” “Multi-” is a morpheme.

CATS

CAT + S

TUMBLING

TUMBLE + ING

UNRELIABILITY

UN + RELY + ABLE + ITY

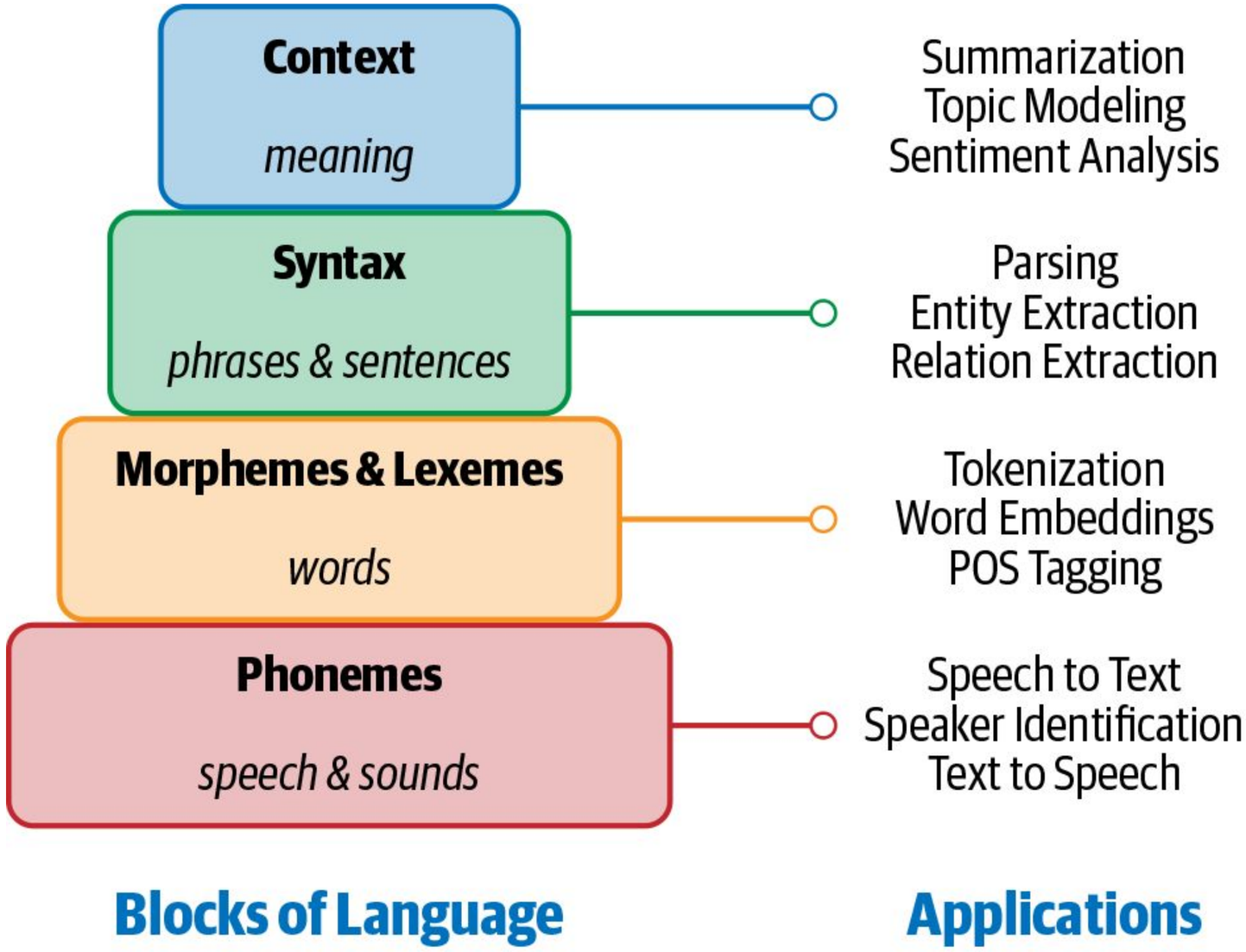
lexemes

- A lexeme represents the most basic building block of a language.
- If you open a typical dictionary, the entries there are lexemes.
- Most lexemes have variations which build upon its most basic form.
- Thus, the lexeme walk could vary in form, such as in walks, walked, and walking. The lexeme slow also varies in form: slower, slowest, slowly.

DOG

RUN

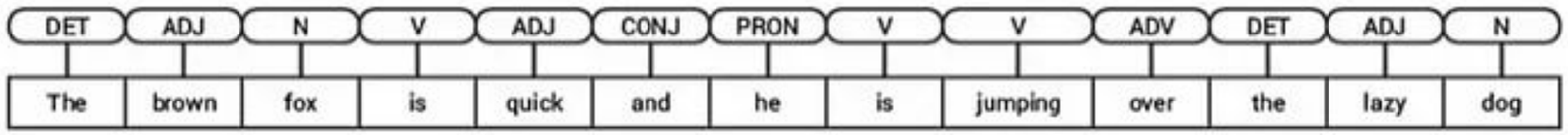
Morphological analysis studies word structure through morphemes and lexemes, forming the foundation for NLP tasks like tokenization, stemming, word embeddings, and part-of-speech tagging.

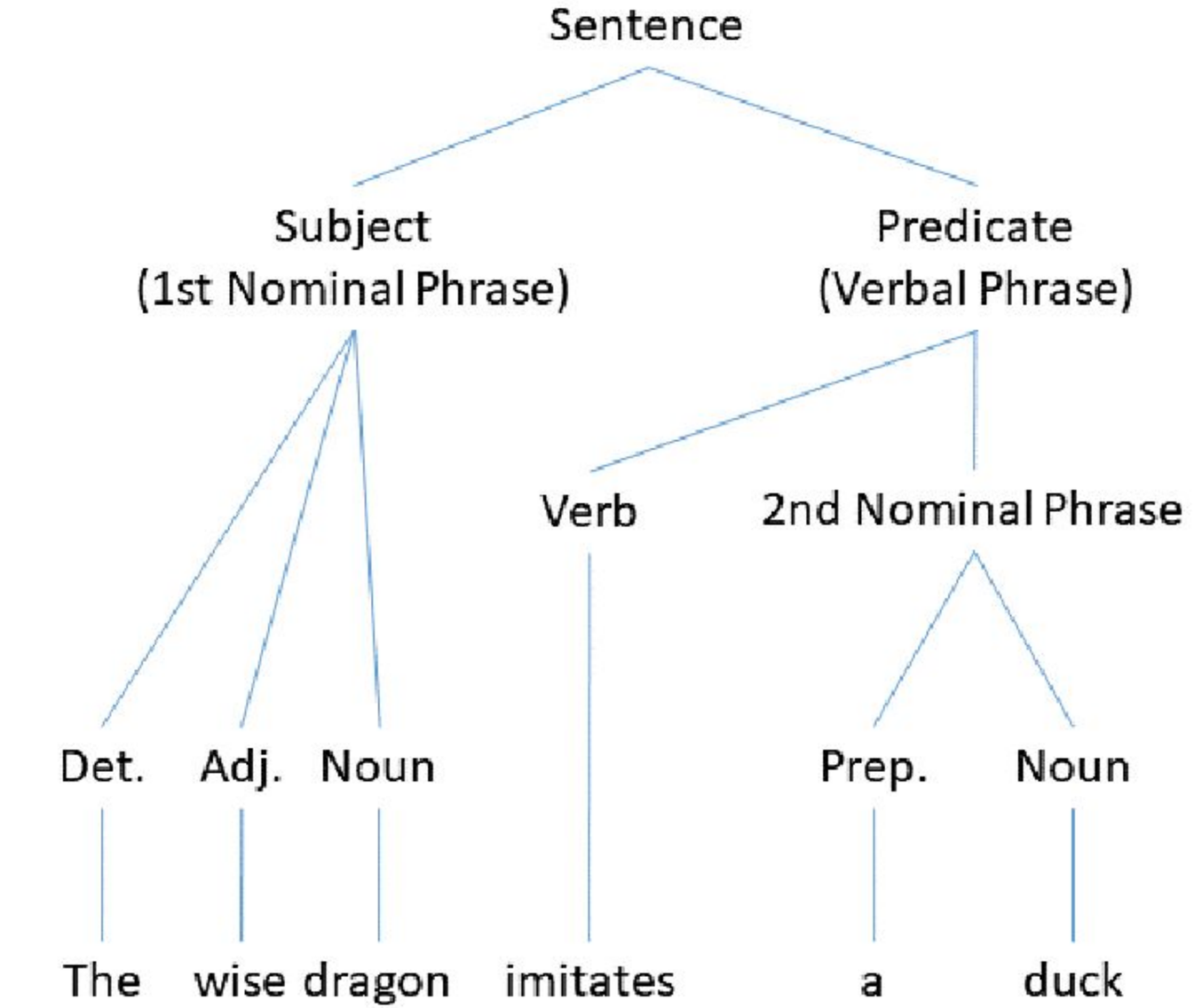


Building Blocks of Language and their applications.

Syntax

- **Syntax** : Defines rules for forming grammatically correct sentences from words and phrases.
- **NLP Applications** : Tasks like entity extraction and relation extraction rely on syntactic parsing.
- **Language Variation** : Different languages have unique syntactic structures, requiring tailored processing approaches.





Syntactic structure of a sentences

Context

- **Context :** Combines language parts to convey meaning, including literal meaning, world knowledge, and common sense.
- **Contextual Influence :** The meaning of a sentence can change based on context, as words and phrases may have multiple meanings. *"I saw a bat outside."*

- Generally, context is composed from semantics and pragmatics.
  - Semantics is the direct meaning of the words and sentences without external context.
  - Pragmatics adds world knowledge and external context of the conversation to enable us to infer implied meaning.

Natural Language Processing

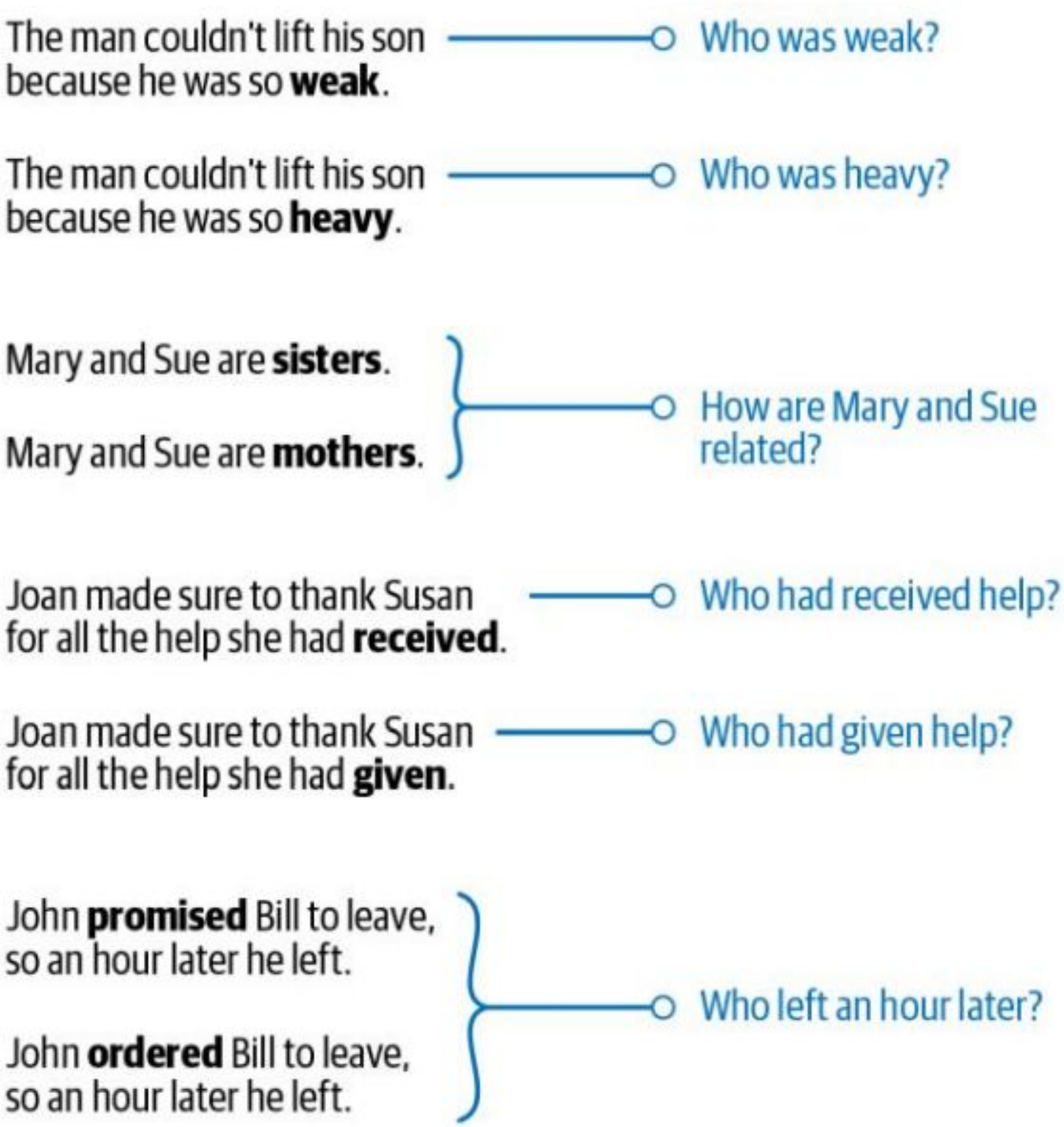
- NLP stands for Natural Language Processing, a part of Computer Science, Human Language, and Artificial Intelligence.
- This technology is used by computers to understand, analyze, manipulate, and interpret human languages.
- NLP algorithms are widely used everywhere in areas like Gmail spam, any search, games, and many more.

Why is NLP Challenging?

- Ambiguity
- Common Knowledge
- Creativity
- Diversity Across Languages

Ambiguity

- Ambiguity means uncertainty of meaning.
- Most human languages are inherently ambiguous.
- This schema has pairs of sentences that differ by only a few words, but the **meaning of the sentences is often flipped** because of this minor change.
- These examples are easily disambiguated by a human but are not solvable using most NLP techniques.



Common Knowledge

- **Common Knowledge** : A set of facts most humans know and assume in conversations without explicitly mentioning them.

For example, consider two sentences: “man bit dog” and “dog bit man.” We all know that the first sentence is unlikely to happen, while the second one is very possible.

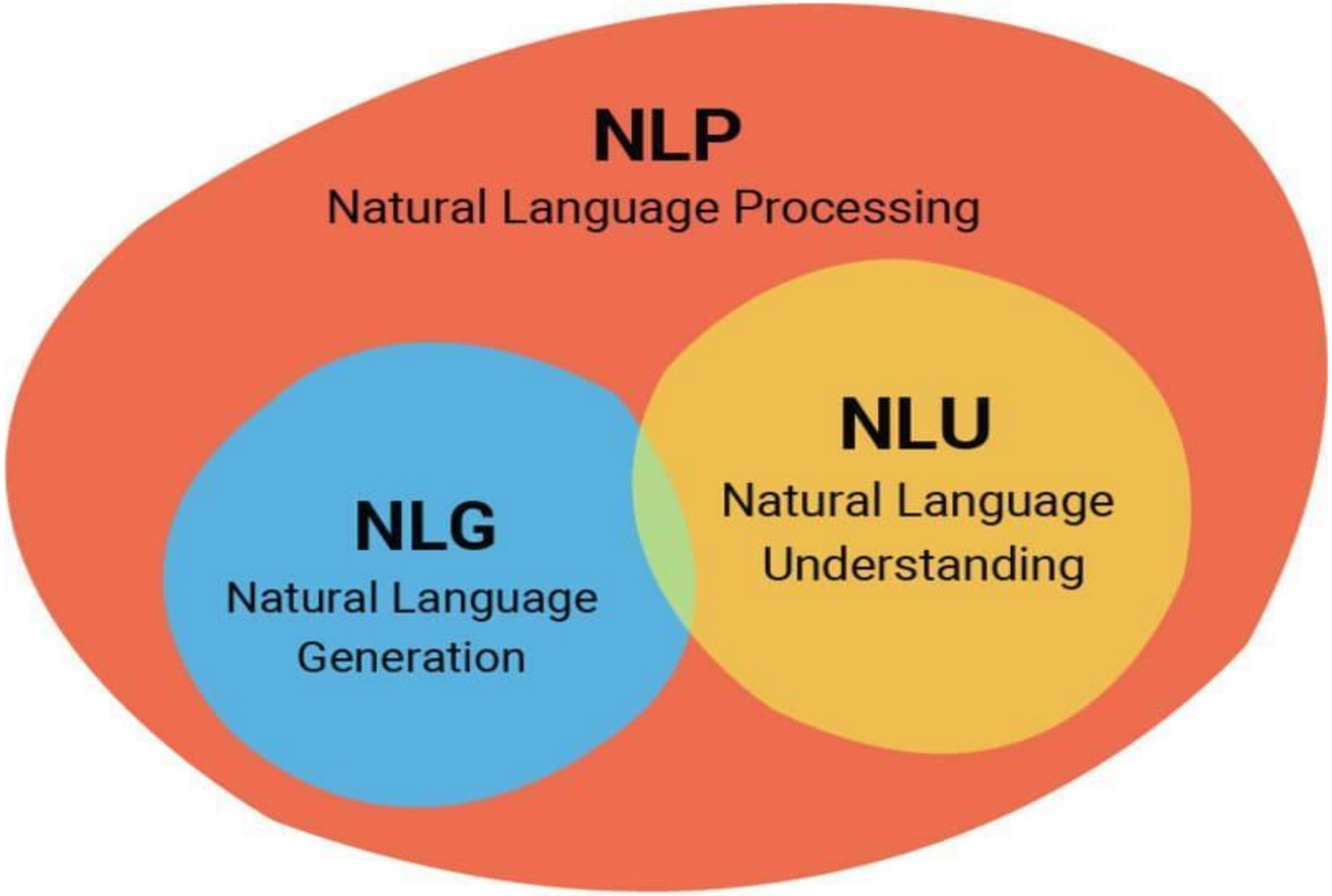
- **Contextual Understanding** : Common knowledge affects sentence meaning (e.g., "man bit dog" vs. "dog bit man").
- **NLP Challenge** : Encoding human common knowledge in computational models for accurate language understanding.

Creativity

- Language is not just rule driven; there is also a creative aspect to it. Various styles, dialects, genres, and variations are used in any language.
- Poems are a great example of creativity in language.
- Making machines understand creativity is a hard problem not just in NLP, but in AI in general.

Diversity Across Languages

- For most languages in the world, there is no direct mapping between the vocabularies of any two languages.
- This makes porting an NLP solution from one language to another hard.
- A solution that works for one language might not work at all for another language.
- This means that one either builds a solution that is language agnostic or that one needs to build separate solutions for each language.
- While the first one is conceptually very hard, the other is laborious and time intensive.



Components of NLP

Natural Language Understanding

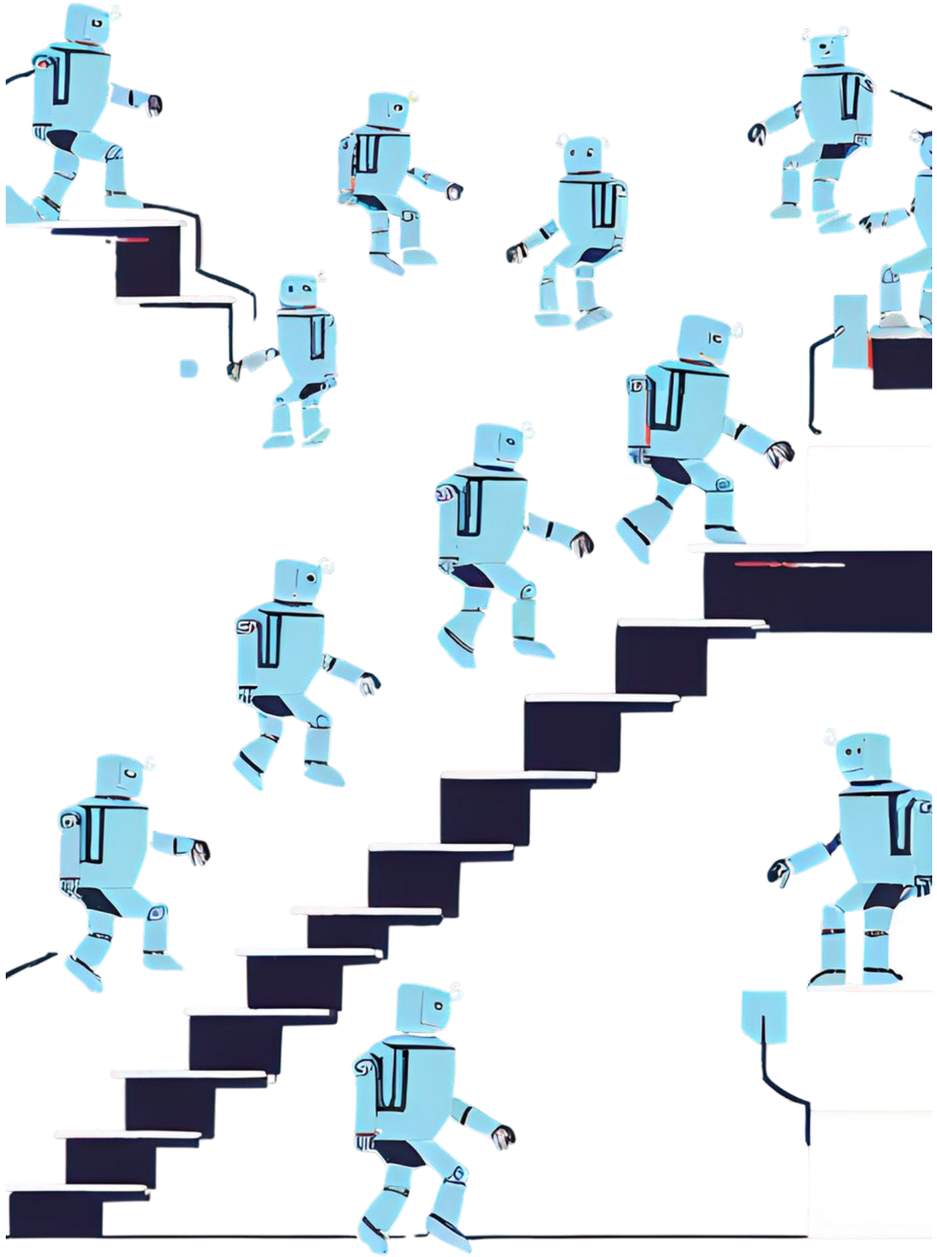
- Natural language understanding is the ability of a computer program to understand the meaning of human language in a way that is similar to how humans do.
- This can involve understanding the grammar and structure of a language, as well as the meanings of words and phrases and how they are used in different contexts.
- Natural language understanding is an important part of natural language processing, which is the field of computer science and artificial intelligence that focuses on enabling computers to process and understand human language.

Natural Language Generation

- Natural language generation is the ability of a computer program to produce written or spoken language that is similar to how humans do.
- This can involve generating text or speech that is grammatically correct and coherent, and that conveys information or ideas in a way that is understandable to humans.
- Natural language generation is an important part of natural language processing, which is the field of computer science and artificial intelligence that focuses on enabling computers to process and understand human language.

BREAK

Phases In NLP

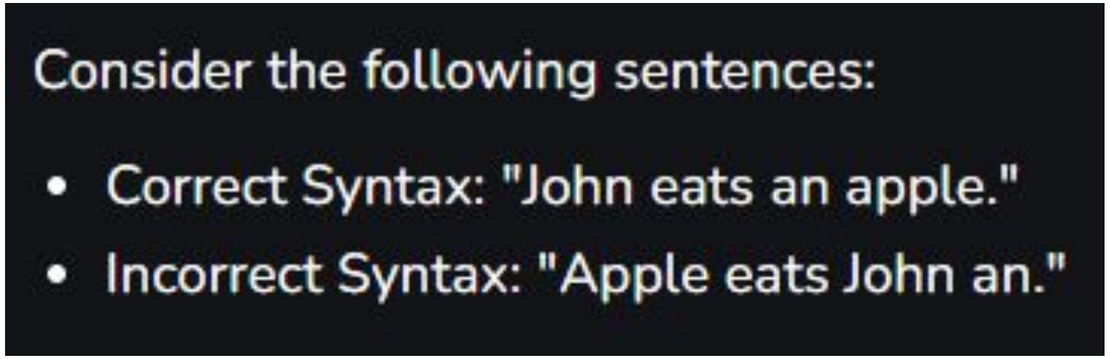


Lexical Analysis

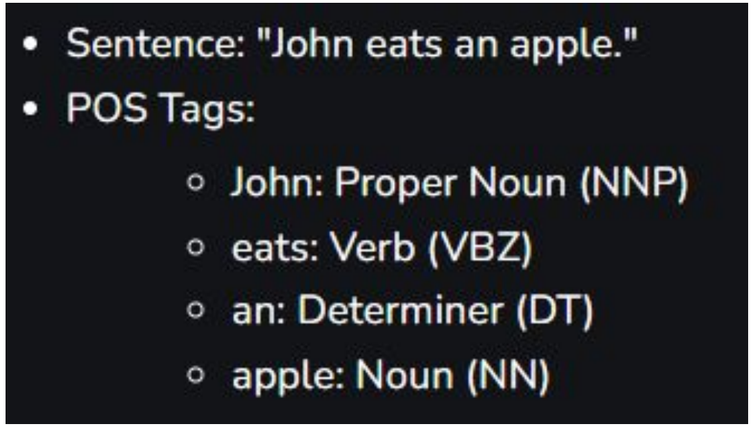
- **Lexical Analysis** : Also called lexing or tokenization, it breaks raw text into individual words or phrases (tokens) for further NLP processing.
- It defines patterns for identifying tokens, including word boundaries, character construction, and parts of speech.
- Tokenized text is further used by NLP algorithms to extract meaning through tasks like part-of-speech tagging, named entity recognition, sentiment analysis, and text summarization.
- **Importance of Lexical Analysis** : It is a crucial first step, providing the foundational building blocks for understanding language structure and meaning.

Syntactic Analysis

- **Syntactic Analysis (Parsing)** : Analyzes sentence structure to check grammatical correctness and identify relationships between words.
- **Importance of Syntactic Analysis** : It provides structural and grammatical information essential for understanding sentence meaning.



Syntax analysis



Part of Speech Tagging

Semantics Analysis

- **Semantic Analysis** : Focuses on analyzing the meaning of words, phrases, and sentences to determine their intended significance and relationships.
- It holds symbolic depiction of sentence meaning, expressing relationships between words and phrases in a way that a computer can understand.
- **Importance of Semantic Analysis** : It provides essential information about sentence meaning, necessary for understanding intent and significance.

Literal Interpretation: "What time is it?"

- This phrase is interpreted literally as someone asking for the current time, demonstrating how semantic analysis helps in understanding the intended meaning.

Discourse Analysis

- **Discourse Analysis** : This focuses on mapping the overally context of a text over a longer form corpus.
- This helps understand the relationship between words and the overall flow of its meaning across sentences

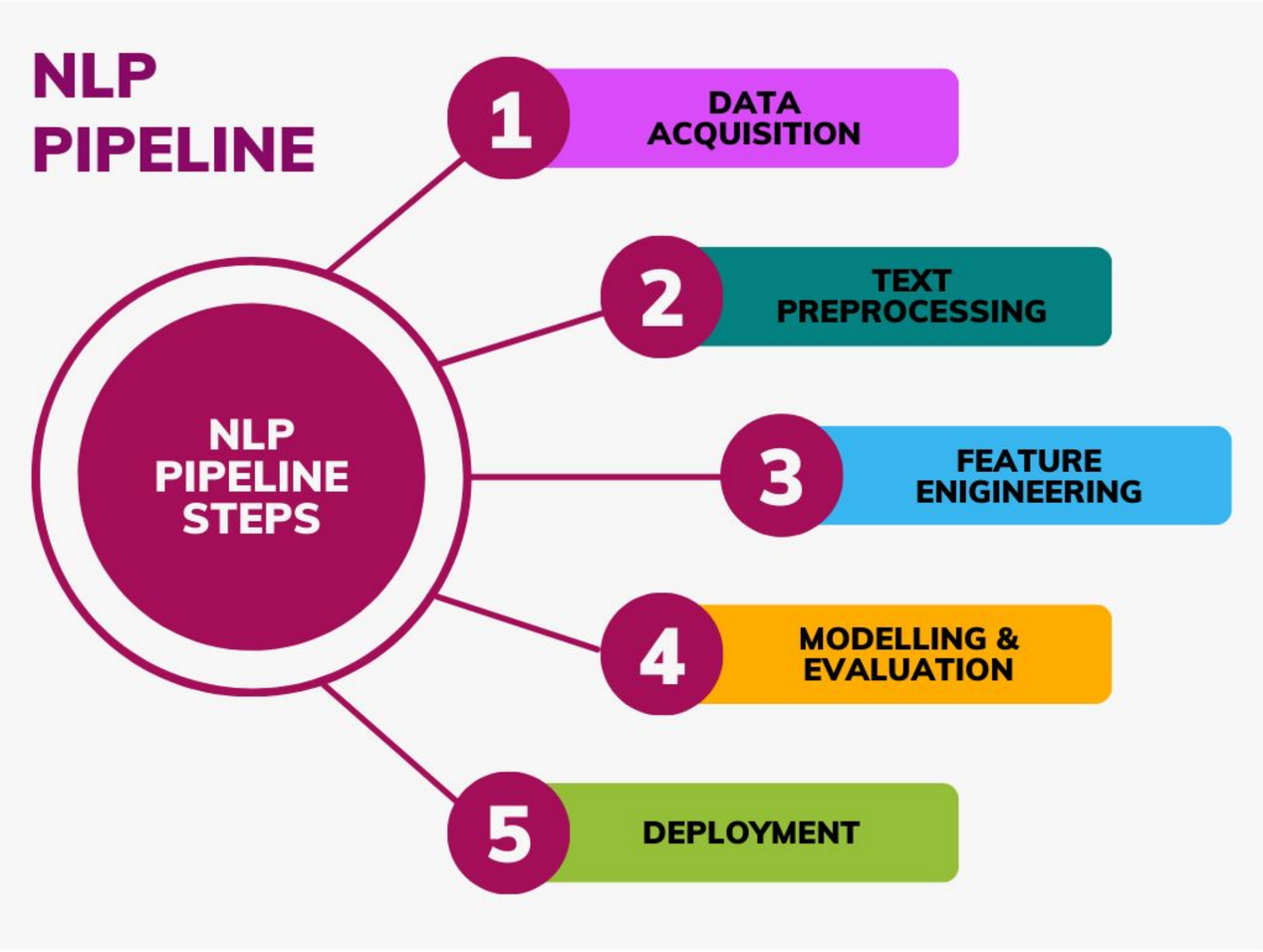
- **Contextual Reference**: "This is unfair!"
    - To understand what "this" refers to, we need to examine the preceding or following sentences. Without context, the statement's meaning remains unclear.

Pragmatic Analysis

- **Pragmatic Analysis** : A subfield of NLP focused on understanding the intended meaning of communication in its broader context, beyond just words and sentences.
- Includes speaker intentions, context, and audience knowledge and beliefs.
- **Applications** : Useful for tasks like dialogue systems, where understanding intent is key to providing appropriate responses, and sentiment analysis, especially for detecting irony or sarcasm.

- **Contextual Greeting**: "Hello! What time is it?"
  - "Hello!" is more than just a greeting; it serves to establish contact.
  - "What time is it?" might be a straightforward request for the current time, but it could also imply concern about being late.
- **Figurative Expression**: "I'm falling for you."
  - The word "falling" literally means collapsing, but in this context, it means the speaker is expressing love for someone.

NLP Pipelines



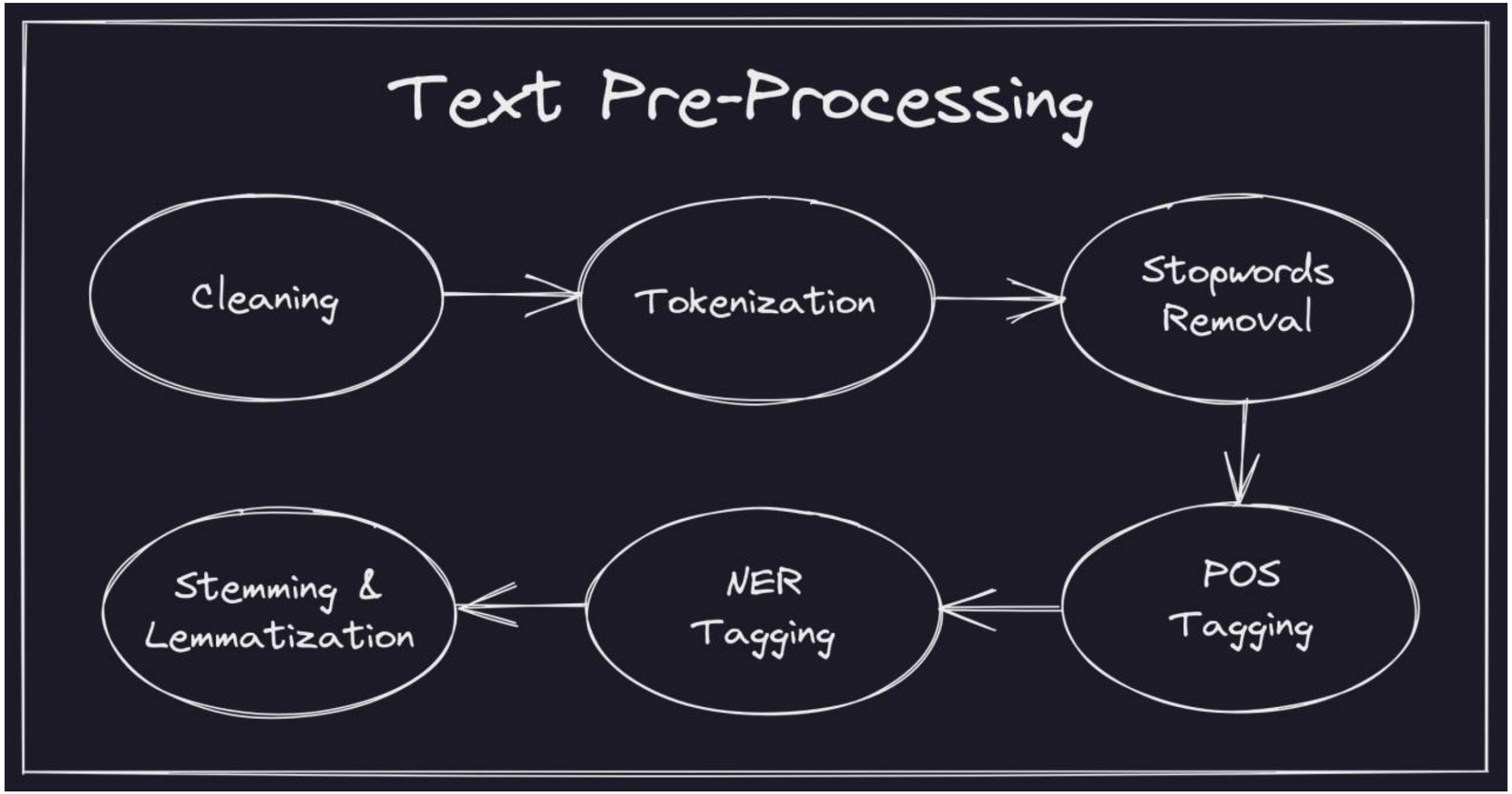
NLP Pipelines (Generic)

1. Text Processing : take raw input text, clean it, normalize it, and convert it into a form that is suitable for feature extraction.

2. Feature Extraction: 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.

3. Modeling: Design a model, fit its parameters to training data, use an optimization procedure, and then use it to make predictions about unseen data.

Step 1: Text processing



Step 1: Text processing

Cleaning — The first step in text processing is to clean the data. i.e., removing irrelevant items, such as HTML tags. This can be done in many ways. Example includes using regular expressions, beautiful soup library, CSS selector, etc.

Normalization — The cleaned data is then normalized by converting all words to lowercase and removing punctuation and extra spaces.

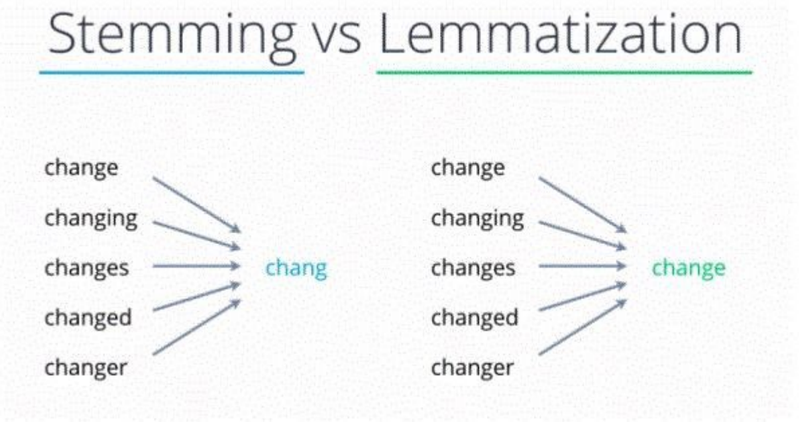
Tokenization — The normalized data is split into words, also known as tokens.

Stop Words removal — After splitting the data into words, the most common words (a, an, the, etc.), also known as stop words are removed.

Step 1: Text processing

Parts of Speech Tagging — The parts of speech are identified for the remaining words

Stemming and Lemmatization — Converting words into their canonical / dictionary forms, using stemming and lemmatization.



One-hot encoding

Fruit	Categorical value of fruit	Price
apple	1	5
mango	2	10
apple	1	15
orange	3	20

Converts categorical data into binary vectors where each category is represented by a unique position marked with "1" and others as "0".

Fruit_apple	Fruit_mango	Fruit_orange	price
1	0	0	5
0	1	0	10
1	0	0	15
0	0	1	20

- **Advantages** : Simple to implement and preserves all category information without loss.
- **Disadvantages** : Inefficient for large vocabularies, increases dimensionality, and does not capture relationships between categories.

Label encoding

- **Definition** : Converts categorical data into numerical values by assigning a unique integer to each category.
    - *Example:* "Red" → 0, "Blue" → 1, "Green" → 2
- **Advantages** : Memory-efficient and simple to implement for ordinal data.
- **Disadvantages** : Implies an unintended order between categories, which may mislead some machine learning models.

Height	Height
Tall	0
Medium	1
Short	2

Step 2: Feature Extraction

After performing these steps, the text will look very different from the original data, but it captures the essence of what was being conveyed in a form that is easier to work with.

Now the text is normalized, can it be fed into a statistical or machine learning model? Not exactly.

Here's why:

Computers cannot understand words/languages, It only understands numbers.

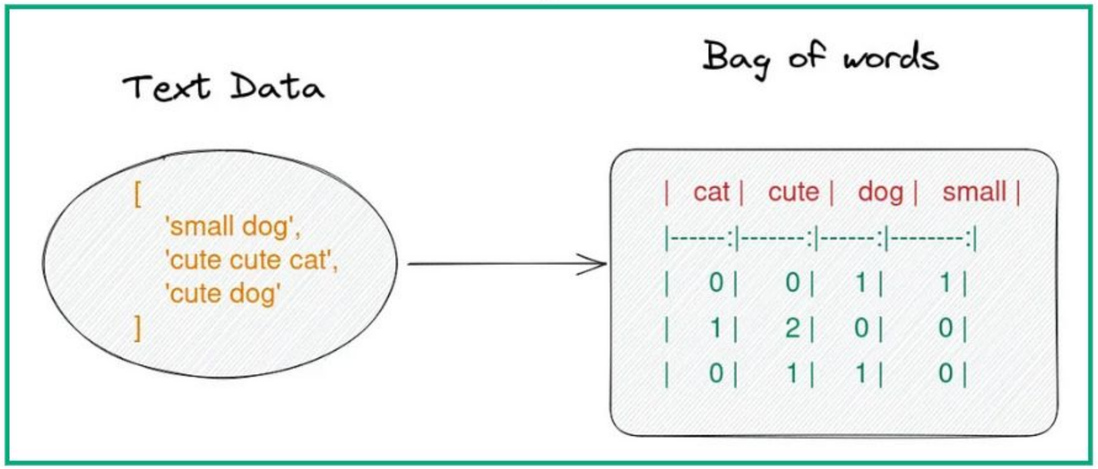
Step 2: Feature Extraction

There are many ways to represent textual information, and only through practice one can learn and use respective techniques based on tasks.

- **Bag of words (BOW) model**
- **Term Frequency — Inverse Document Frequency (TF-IDF)**
- **Encoding**
- **Word Embeddings**

Bag of Words (BOW) model

- This model treats each document as an unordered list or bag of words.
- **Document** refers to a unit of text that is being analyzed.For example, while performing a sentiment analysis on tweets, each tweet is considered as a document.
- The approach is to turn each document into a vector of numbers representing how many times each word occurs in a document.
- A set of words is a corpus and this gives the context for vectors to be calculated.



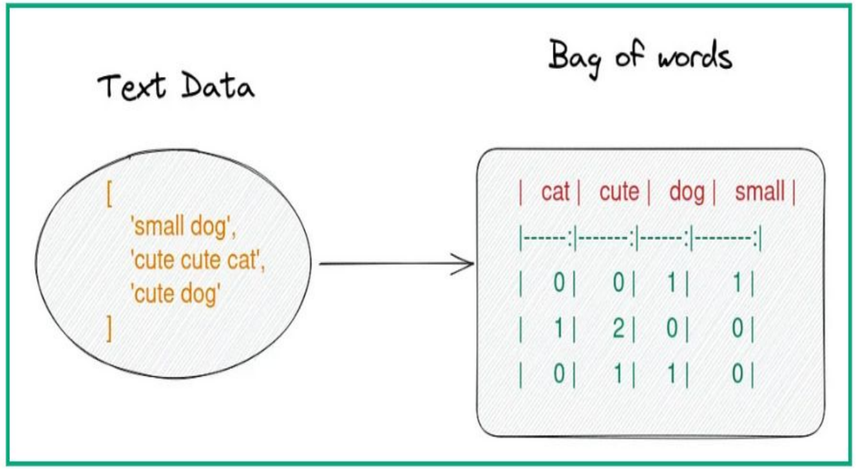
Bag of Words (BOW) model

Step 1 : First, all the unique words in the corpus are collected to form a **vocabulary**

Step 2 : Arrange these words in some order to form vector element positions or columns of a table and each row is assumed as a document

Step 3 : Count the occurrence of each word in each document and enter the value in respective column. This can be called as Document-term matrix which contains documents in rows and terms in columns, interpreting each element as term frequency.

Application : Comparison of documents based on term frequencies. This can be done by calculating dot product/cosine similarity between the two row vectors.



2.2 Term Frequency -Inverse Document Frequency (TF-IDF)

TERM FREQUENCY

$$TF(t, d) = \frac{\text{Number of times term t appears in document d}}{\text{Total number of terms in document d}}$$

This is the measure of word frequency per document. The assumption here is that the higher frequency indicates higher importance.

Limitation : TF ignores a term's global importance, giving common words like "the" or "and" high scores despite their low significance in distinguishing documents.

2.2 Term Frequency -Inverse Document Frequency (TF-IDF)

INVERSE DOCUMENT FREQUENCY

$$IDF(t, D) = \log \frac{\text{Total number of documents in corpus D}}{\text{Number of documents containing term t}}$$

This measures a term's importance by reducing the weight of common words and increasing the weight of rare ones across a corpus.

Note - A higher IDF value indicates a term is rare and more significant, while a lower value means the term is common.

2.2 Term Frequency -Inverse Document Frequency (TF-IDF)

TERM FREQUENCY - INVERSE DOCUMENT FREQUENCY

TFID is computed as the product of tf and idf

$$TF(t, d) = \frac{\text{Number of times term t appears in document d}}{\text{Total number of terms in document d}}$$

$$IDF(t, D) = \log \frac{\text{Total number of documents in corpus D}}{\text{Number of documents containing term t}}$$

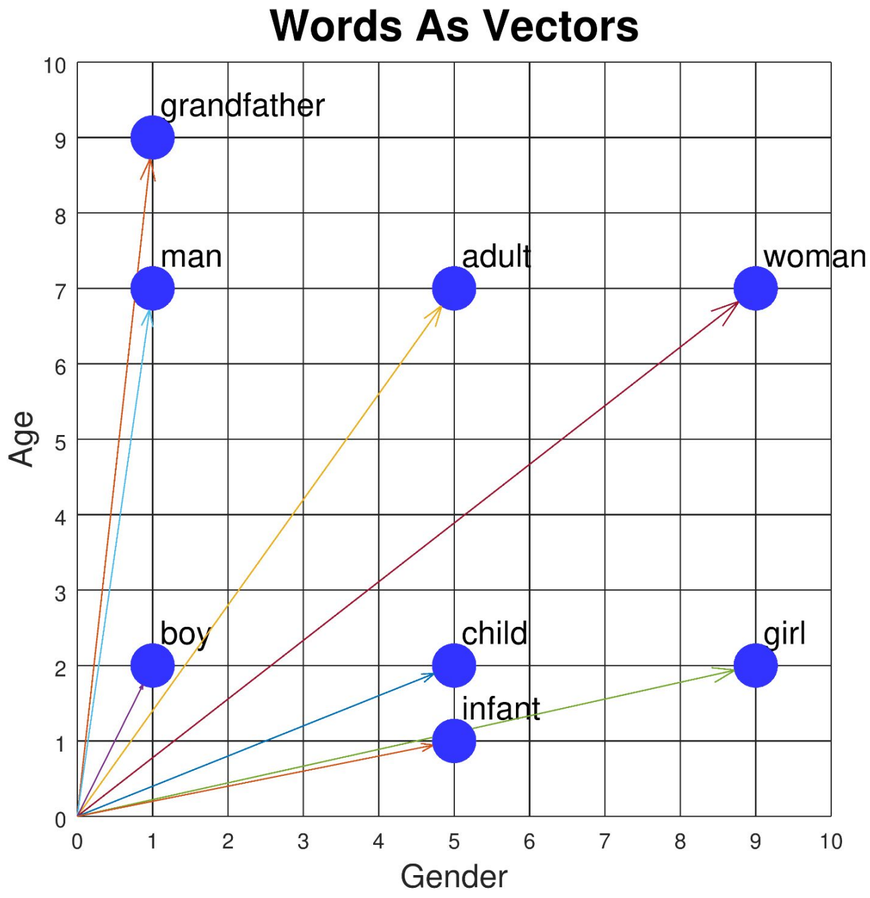
Measures the importance of a term in a document by balancing how often it appears (TF) with how rare it is across all documents (IDF).

Index →	0	1	2	3	4	5	6	7	8
	and	**document**	**first**	**is**	**one**	**second**	**the**	**third**	**this**
"This is the first document."	0	0.46979139	0.58028582	0.38408524	0	0	0.38408524	0	0.38408524
"This document is the second document."	0	0.6876236	0	0.28108867	0	0.53864762	0.28108867	0	0.28108867
"And this is the third one."	0.51184851	0	0	0.26710379	0.51184851	0	0.26710379	0.51184851	0.26710379
"Is this the first document?"	0	0.46979139	0.58028582	0.38408524	0	0	0.38408524	0	0.38408524

Word Embeddings

- **Word Embeddings** : Represent words as vectors meaning—similar words are closer, and relation reflected by their relative positions.
- **Limitations of Encoding** : Inefficient for large v size grows with the number of words.

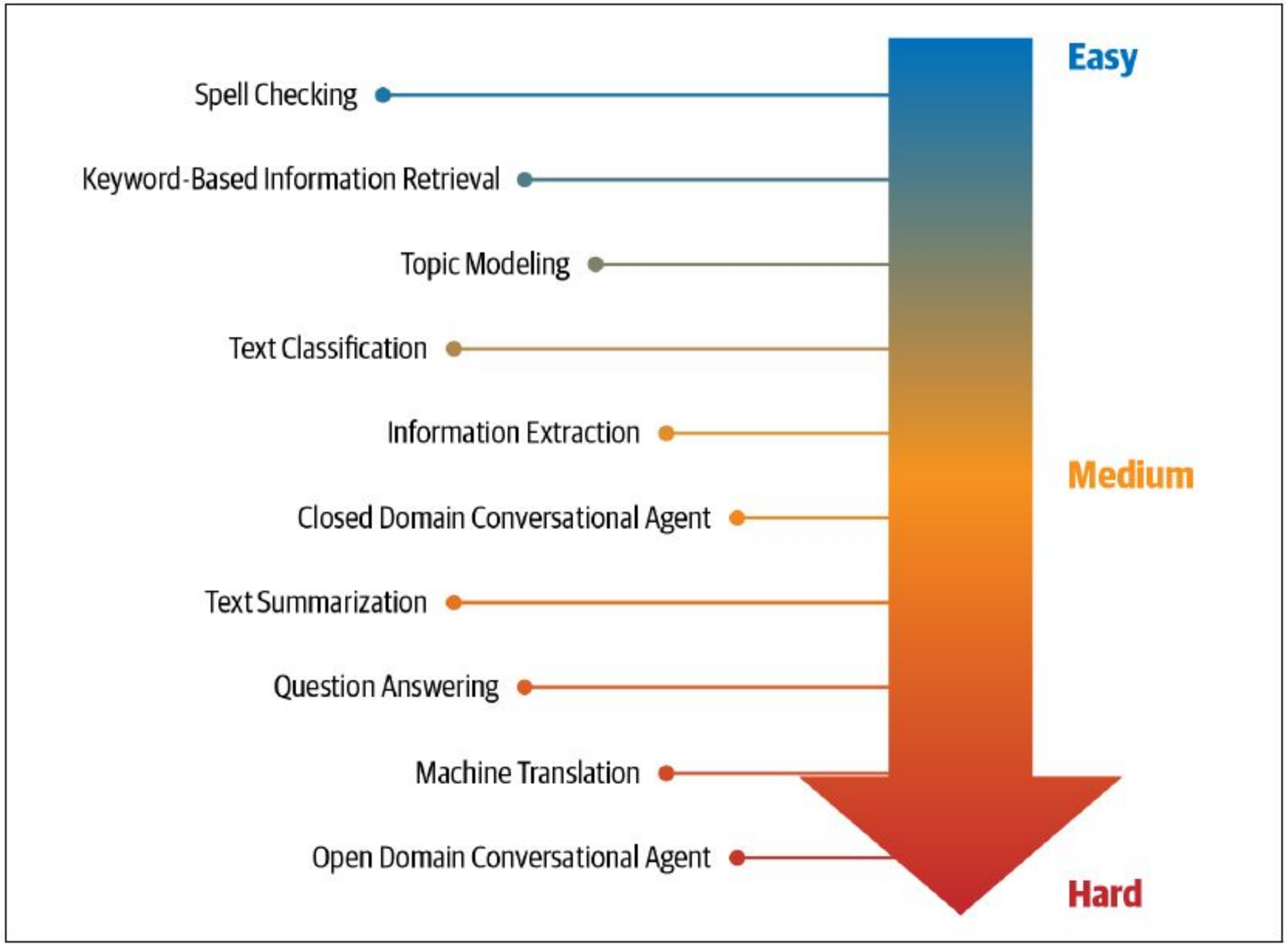
- **Applications** : Useful for tasks like finding analc antonyms, and classifying words by sentiment (



Modeling

- **Modeling in NLP** : Involves designing and training a statistical or machine learning model to make predictions on unseen data.
- **Numerical Features Advantage** : Enable the use of various machine learning models or their combinations.
- **Deployment** : Models can be deployed as web apps, mobile apps, or integrated into other services.

NLP tasks organized according to their relative difficulty



Thank You