Introduction to NLP

03/02/2025

GODS 4.0
AS NEVER BEFORE

By Mohammed Arbi Nsibi

MOHAMED ARBI NSIBI

- Final year ICT engineering student@ SUP'COM
- GDG Carthage member
- Former lead of GDSG SUP'COM 23/24
- GDGoC SUP'COM & ISAMM Mentor

mohammedarbinsibi@gmail.com

https://huggingface.co/Goodnight7

https://www.linkedin.com/in/mohammed-arbi-nsibi-584a43241/

Content

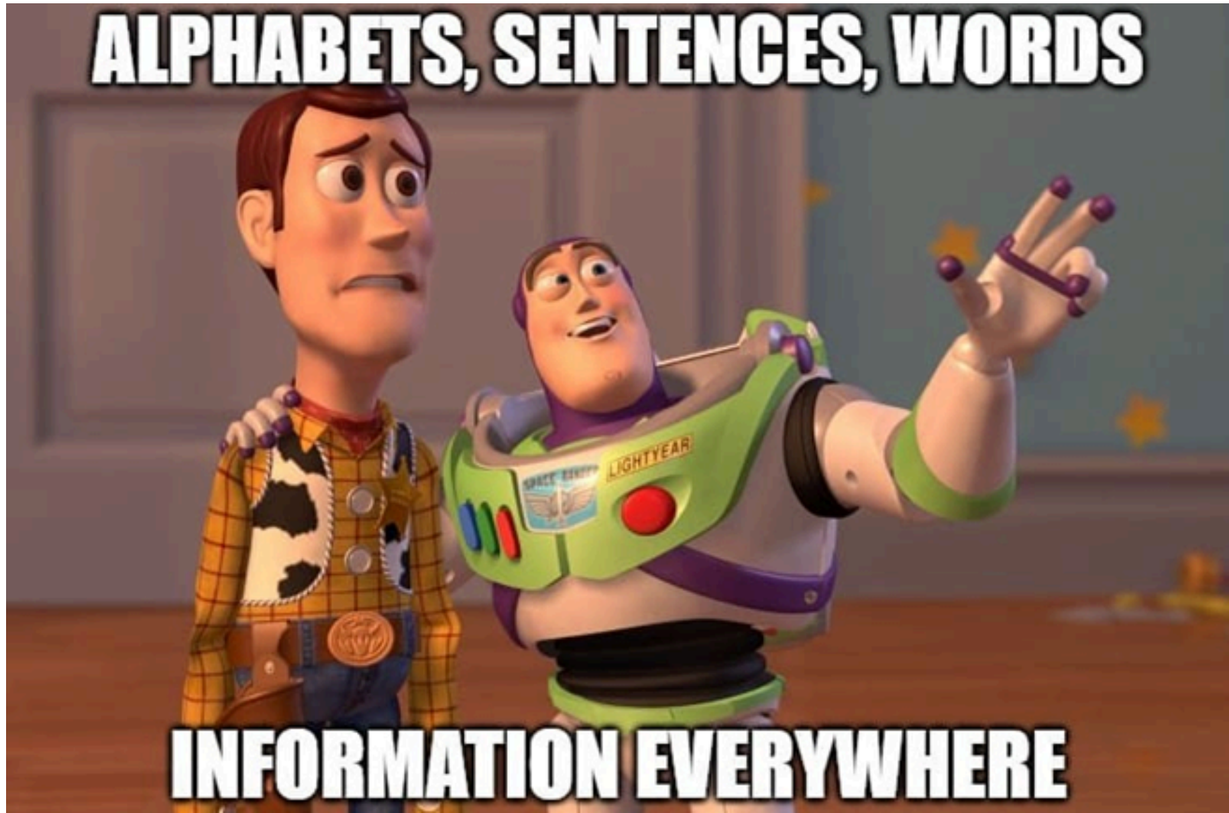
- What is NLP
- NLP History
- Components of NLP
- NLP Roadmap
- Pipeline
- try with llm
- Let’s code
- QUIZ

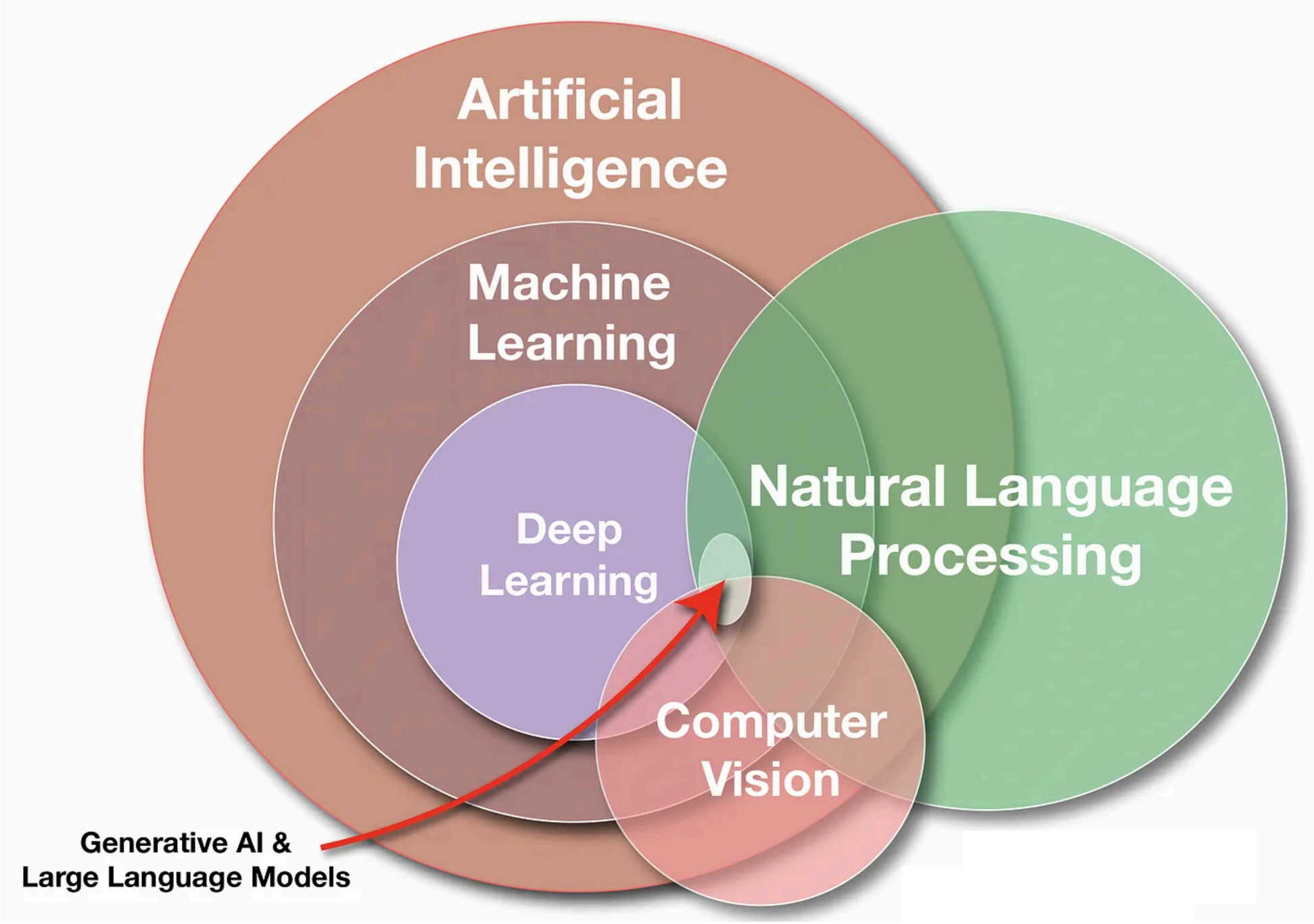
By Mohammed Arbi Nsibi

WHAT IS NLP ?

WHAT IS NLP ?

is referred to as NLP. It is a **subset** of AI that enables machines to comprehend and analyze **human languages**. Text or audio can be used to represent human languages.




Artificial Intelligence
Machine Learning
Deep Learning
Natural Language Processing
Computer Vision
Generative AI & Large Language Models

How does Natural Language Processing work?

- Converting unstructured data into computer-readable language by NLP attributes
- Complex algorithms to break down any text content to extract meaningful information from it.
- Collected data is then used to further teach machines the logic of natural language

By Mohammed Arbi Nsibi

Why natural language processing is important?

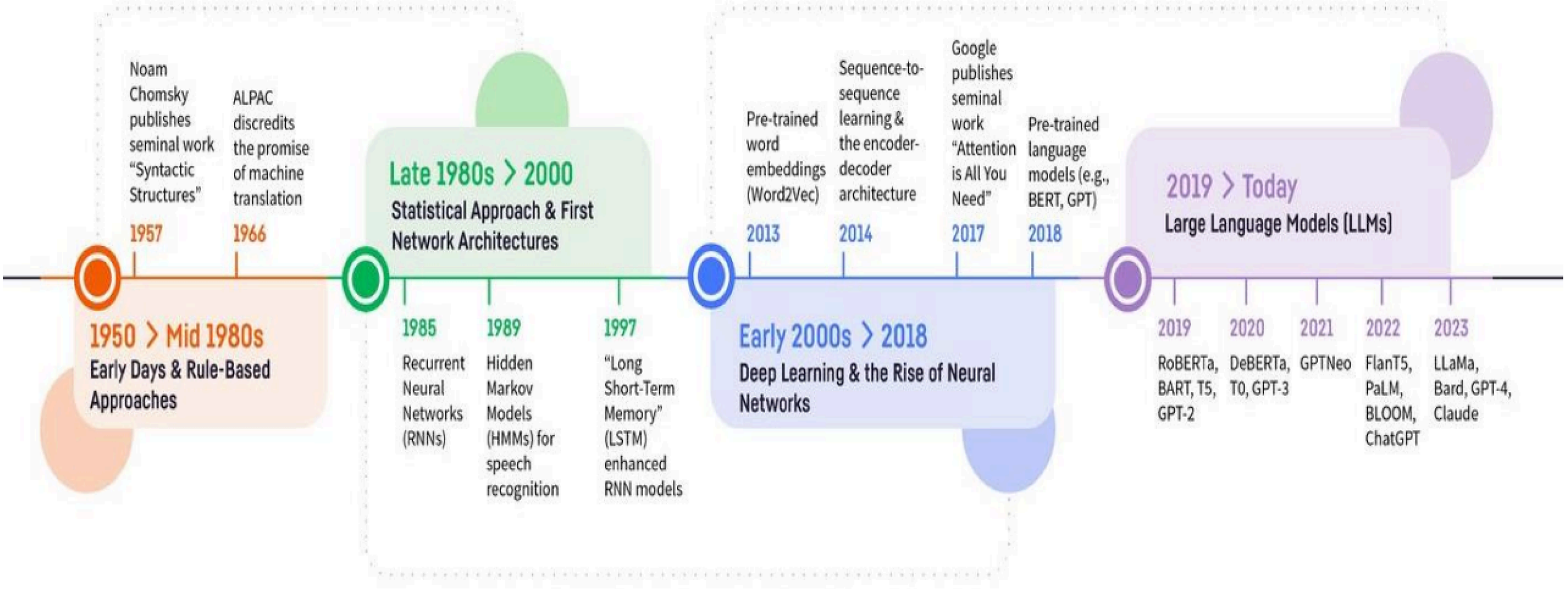
- Facilitates Human-Computer Interaction: users to communicate with devices, applications, and systems using spoken or written language

- Automates Routine Tasks: asks such as text summarization, sentiment analysis, document classification, and information extraction, improving efficiency and productivity.

- Empowers Chatbots and Virtual Assistants: understand user queries, provide information, perform tasks, and offer personalized recommendations, enhancing customer service and user experience.

By Mohammed Arbi Nsibi

NLP History


1950 > Mid 1980s
Early Days & Rule-Based Approaches
1957
Noam Chomsky publishes seminal work "Syntactic Structures"
1966
ALPAC discredits the promise of machine translation
Late 1980s > 2000
Statistical Approach & First Network Architectures
1985
Recurrent Neural Networks (RNNs)
1989
Hidden Markov Models (HMMs) for speech recognition
1997
"Long Short-Term Memory" (LSTM) enhanced RNN models
Early 2000s > 2018
Deep Learning & the Rise of Neural Networks
2013
Pre-trained word embeddings (Word2Vec)
2014
Sequence-to-sequence learning & the encoder-decoder architecture
2017
Google publishes seminal work "Attention is All You Need"
2018
Pre-trained language models (e.g., BERT, GPT)
2019 > Today
Large Language Models (LLMs)
2019
RoBERTa, BART, T5, GPT-2
2020
DeBERTa, T0, GPT-3
2021
GPTNeo
2022
FlanT5, PaLM, BLOOM, ChatGPT
2023
LLaMa, Bard, GPT-4, Claude

Components of NLP

Natural Language Understanding (NLU)

- NLU focuses on interpreting and extracting meaning from human language input.
- It involves techniques such as text parsing, entity recognition, sentiment analysis, and intent detection.
- NLU systems aim to comprehend the content of text or speech input to extract relevant information and understand the user's intentions or queries
- Examples of applications : chatbots that understand user queries, sentiment analysis tools that analyze emotions in text, and voice assistants.

By Mohammed Arbi Nsibi

Components of NLP

Natural Language Generation (NLG)

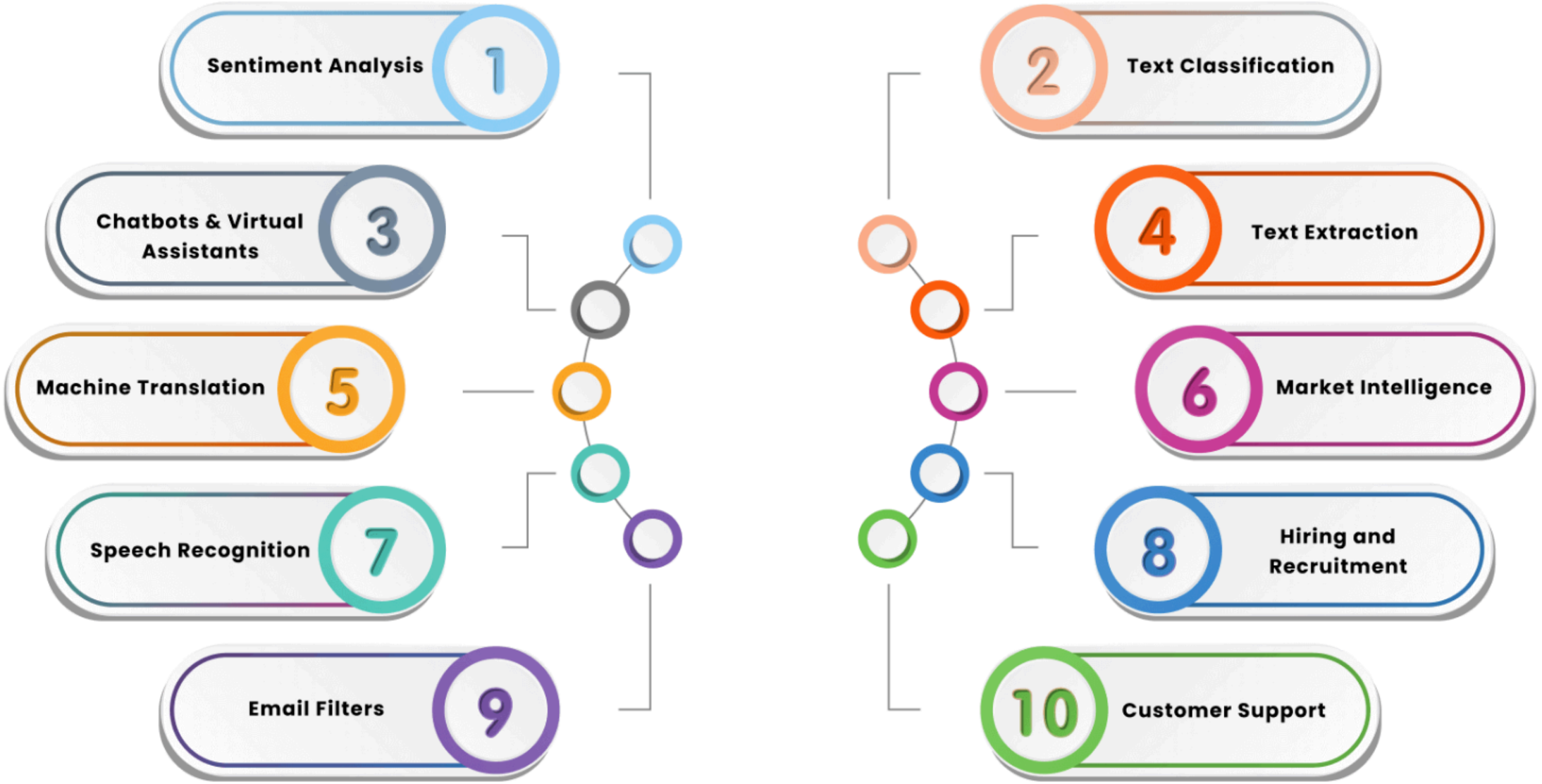
- NLG, on the other hand, deals with creating human-like text or speech output based on structured data or input from NLU systems

- NLG systems generate coherent and contextually relevant text or speech by combining linguistic rules, templates, and sometimes machine learning models

- NLG applications include text summarization, language translation, chatbot responses, and content generation for news articles or reports

By Mohammed Arbi Nsibi

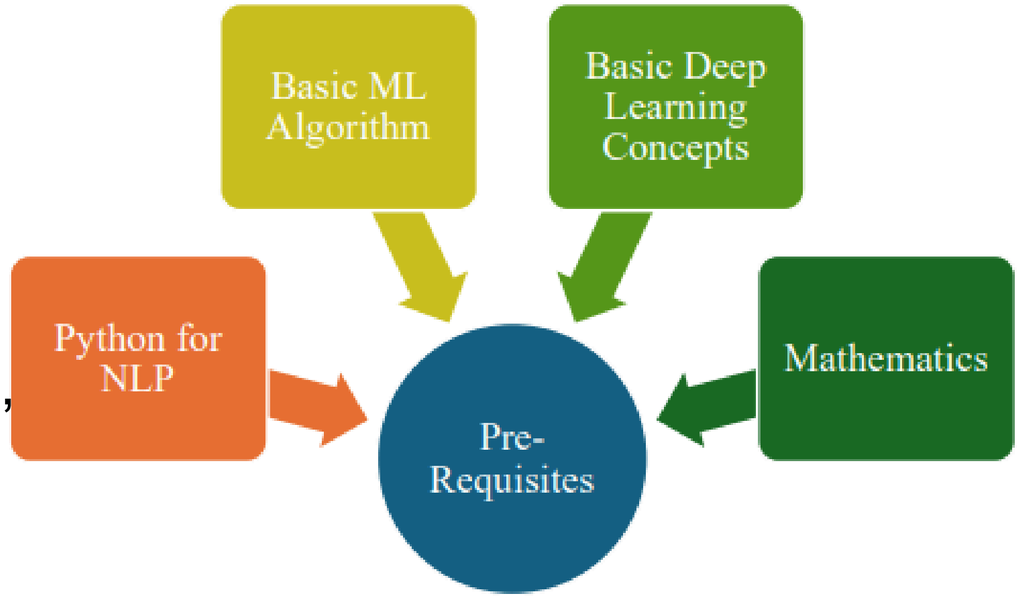
NLP applications



Things We need to know for learning NLP

Natural Language Generation (NLG)

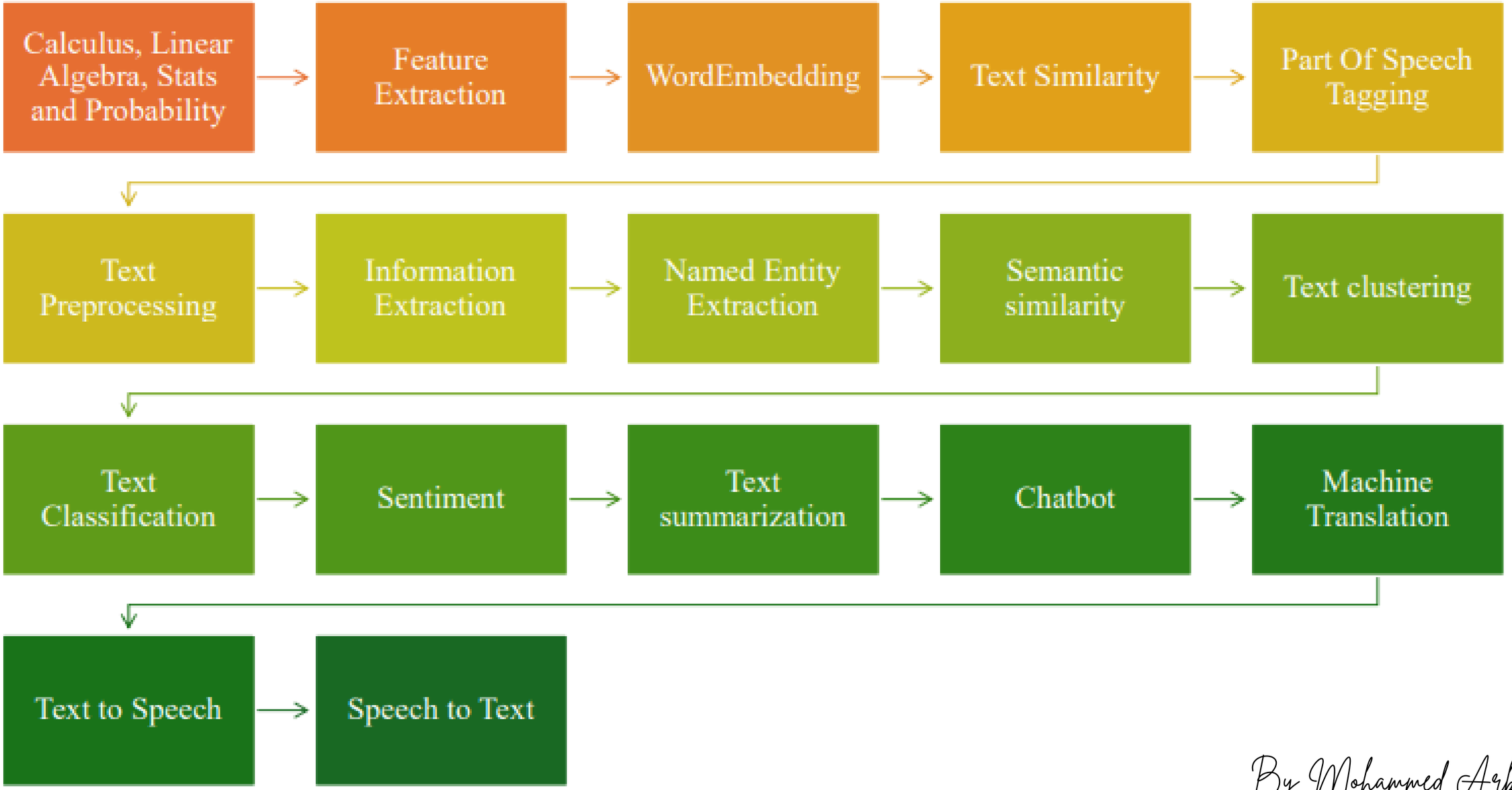
- Text processing techniques
- Word Embedding
- Deep Learning Network for NLP (CNN, LSTM, GRUs, Encoder and Decoder)
- Transformers (BERT, GPT, ALBERT and so on)
- Fine Tuning NLP task
- Large Language Model (LLM)



PyTorch TensorFlow

By Mohammed Arbi Nsibi

Roadmap List for NLP



By Mohammed Arbi Nsibi

NLP Pipeline

Data Acquisition → Text Cleaning → Text Preprocessing → Feature Engineering → Model Building → Evaluation → Deployment

By Mohammed Arbi Nsibi

Text Preprocessing Pipeline

- Sentence Segmentation : Breaks the paragraph into sentence
- Word Tokenization
- Stemming
- Lemmatization
- Identifying Stop Words
- POS tags
- Name Entity Recognition
- Chunking

By Mohammed Arbi Nsibi

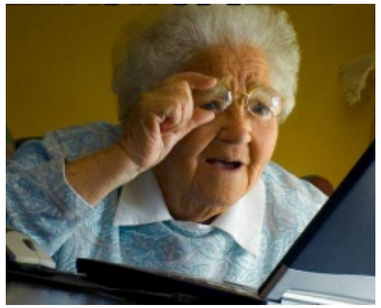
Text Cleaning

Regex or Regular Expression

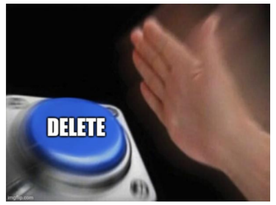
Spelling corrections

used for searching the string of specific patterns. Suppose our data contain phone number, email-Id, and URL. we can find such text using the regular expression. After that either we can keep or remove such text patterns as per requirements.

Text Preprocessing



- Lowercasing
- Stop word removal
- Stemming or lemmatization
- Removing digit/punctuation
- POS tagging
- Named Entity Recognition (NER)



By Mohammed Arbi Nsibi

Text Preprocessing

- Stop word removal

Stopword Removal

Input Text: "The quick brown fox jumps over the lazy dog."

Output Text: "quick brown fox jumps lazy dog."

By Mohammed Arbi Nsibi

Text Preprocessing

Stemming

Input Text: "running"

Output Text: "run"

Lemmatization

Input Text: "better"

Output Text: "good"

By Mohammed Arbi Nsibi

"Elon Musk founded SpaceX in 2002."

Named Entity Recognition (NER)

[Elon Musk: PERSON],
[SpaceX: ORG],
[2002: DATE].

Entity Recognition and Masking

Input Text: "John Smith works at Google."

Output Text: "[PERSON] works at [ORGANIZATION]."

POS tagging

Elon/NNP (Proper noun)
Musk/NNP
founded/VBD
SpaceX/NNP
in/IN
2002/CD (Cardinal number)

Feature Engineering = Text Representation = Text Vectorization.

Our main agenda is to represent the text in the numeric vector in such a way that the ML algorithm can understand the text attribute.

Traditional Approach

- One Hot Encoding

Traditional Approach

- One Hot Encoding

id	color
1	red
2	blue
3	green
4	blue

id	color_red	color_blue	color_green
1	1	0	0
2	0	1	0
3	0	0	1
4	0	1	0

Traditional Approach

- TF-IDF (Term Frequency – Inverse Document Frequency) 1972

  Give more weight to rare words and less to common terms

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

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

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

- Neural Approach (Word embedding)

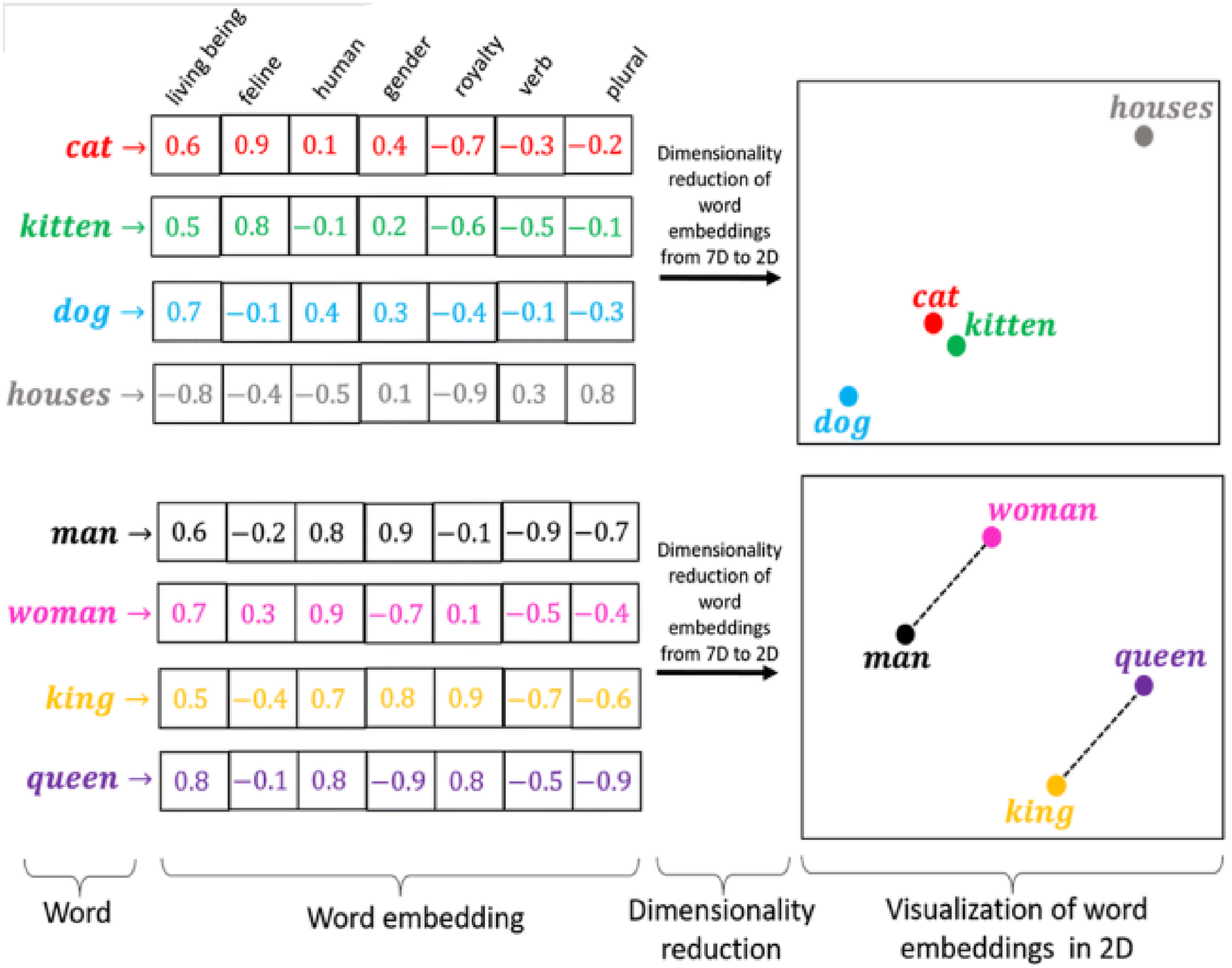
car =[0.8, 0.9, 0.9, 0.01, 0.75]
bike =[0.8, 0.7, 0.2, 0.01, 0.5]

not interpretable for humans

try to incorporate the contextual meaning of the words.

Word	car	bike
Road	0.8	0.8
Speed	0.9	0.7
Fuel	0.9	0.2
Animal	0.01	0.01
Price	0.75	0.5

How can we get these word embedding vectors?


living being
feline
human
gender
royalty
verb
plural
cat → 0.6 0.9 0.1 0.4 −0.7 −0.3 −0.2
kitten → 0.5 0.8 −0.1 0.2 −0.6 −0.5 −0.1
dog → 0.7 −0.1 0.4 0.3 −0.4 −0.1 −0.3
houses → −0.8 −0.4 −0.5 0.1 −0.9 0.3 0.8
Dimensionality reduction of word embeddings from 7D to 2D
houses
cat
kitten
dog
man → 0.6 −0.2 0.8 0.9 −0.1 −0.9 −0.7
woman → 0.7 0.3 0.9 −0.7 0.1 −0.5 −0.4
king → 0.5 −0.4 0.7 0.8 0.9 −0.7 −0.6
queen → 0.8 −0.1 0.8 −0.9 0.8 −0.5 −0.9
Dimensionality reduction of word embeddings from 7D to 2D
woman
man
queen
king
Word
Word embedding
Dimensionality reduction
Visualization of word embeddings in 2D

How can we get these word embedding vectors?

Train our own embedding layer

- **CBOW (Continuous Bag of Words)**
- **SkipGram**

Pre-Trained Word Embeddings

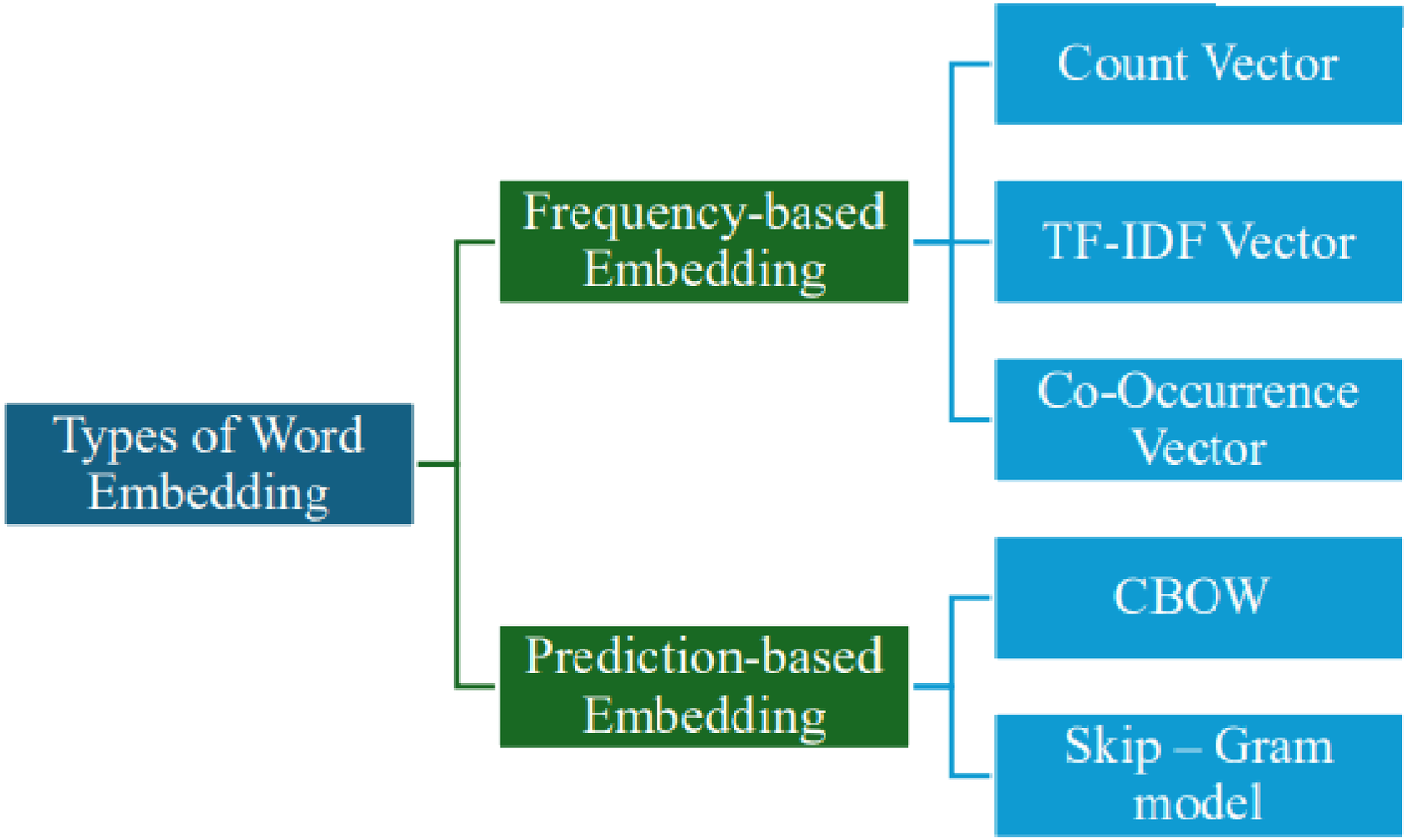
- **Word2vec by Google 2013**
- **GloVe by Stanford (Global Vectors for Word Representation)**
- **fasttext by Facebook**
- **BERT (Bidirectional Encoder Representations from Transformers)**

https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/

Visualizing High-Dimensional Space: https://www.youtube.com/watch?v=wvsE8jm1GzE&ab_channel=GoogleforDevelopers

By Mohammed Arbi Nsibi

To summarize

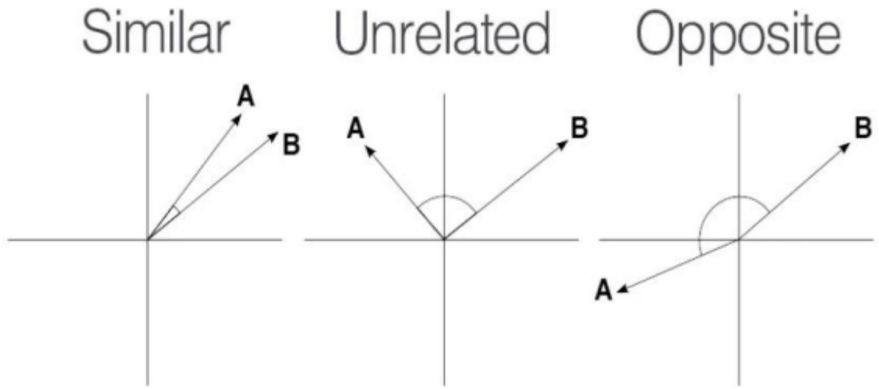


Feature Extraction Techniques summary

Techniques	Main Feature	Use case
CountVectorizer	Coverts text to matrix of word counts	Text classification, topic modeling
TF-IDF (Term Frequency-Inverse Document Frequency)	Assign weights to words based on importance	Information retrieval, text classification
Word embeddings	Vector representation of words based on semantics and syntax	Text classification, Information Retrieval
Bag of words	Represents text as a vector of word frequencies	Text classification, Sentimental Analysis
Bag of n-grams	Capture frequency of sequences of n words	Text classification, Sentimental Analysis
Principal Component Analysis (PCA) , t-SNE	Reduces dimensionality of documents-term matrix	Text visualization, text compression
Part-of-speech (POS) tagging	Assigns parts of speech tag to each word in text	Named entity recognition, text classification
N-grams	sequences of contiguous words or characters, capturing local word dependencies	Captures context and word order information, useful in language modeling, machine translation, and text generation
Named Entity Recognition (NER)	identifies and classifies named entities (e.g., person names, organizations, locations) in text	Information extraction, entity linking, and improving search engine results

Text simmilarity

The dog bites the man == The man bites the dog?



- **According to the lexical similarity, those two phrases are very close and almost identical because they have the same word set.**

- **For semantic similarity, they are completely different because they have different meanings despite the similarity of the word set.**

Techniques are commonly used to compute text similarity

Cosine Similarity

Measures the cosine of the angle between two vectors representing the text.

Used with word embeddings or TF-IDF vectors to compute similarity.

Jaccard Similarity

Measures the similarity between two sets by dividing the size of their intersection by the size of their union.

BERT and Similar Models

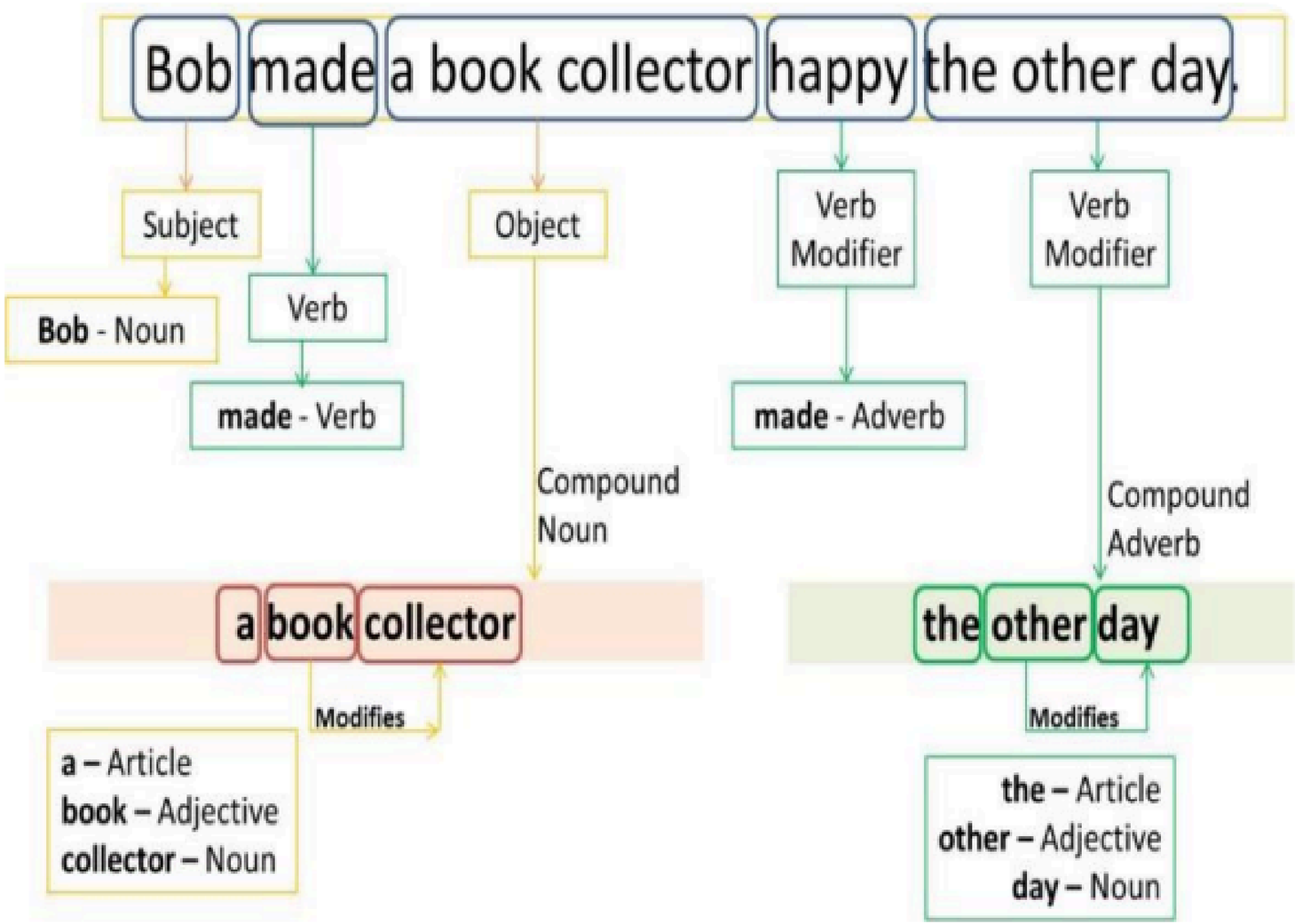
Pre-trained language models like BERT (Bidirectional Encoder Representations from Transformers) can be fine-tuned for text similarity tasks.

Part-of-Speech Tagging

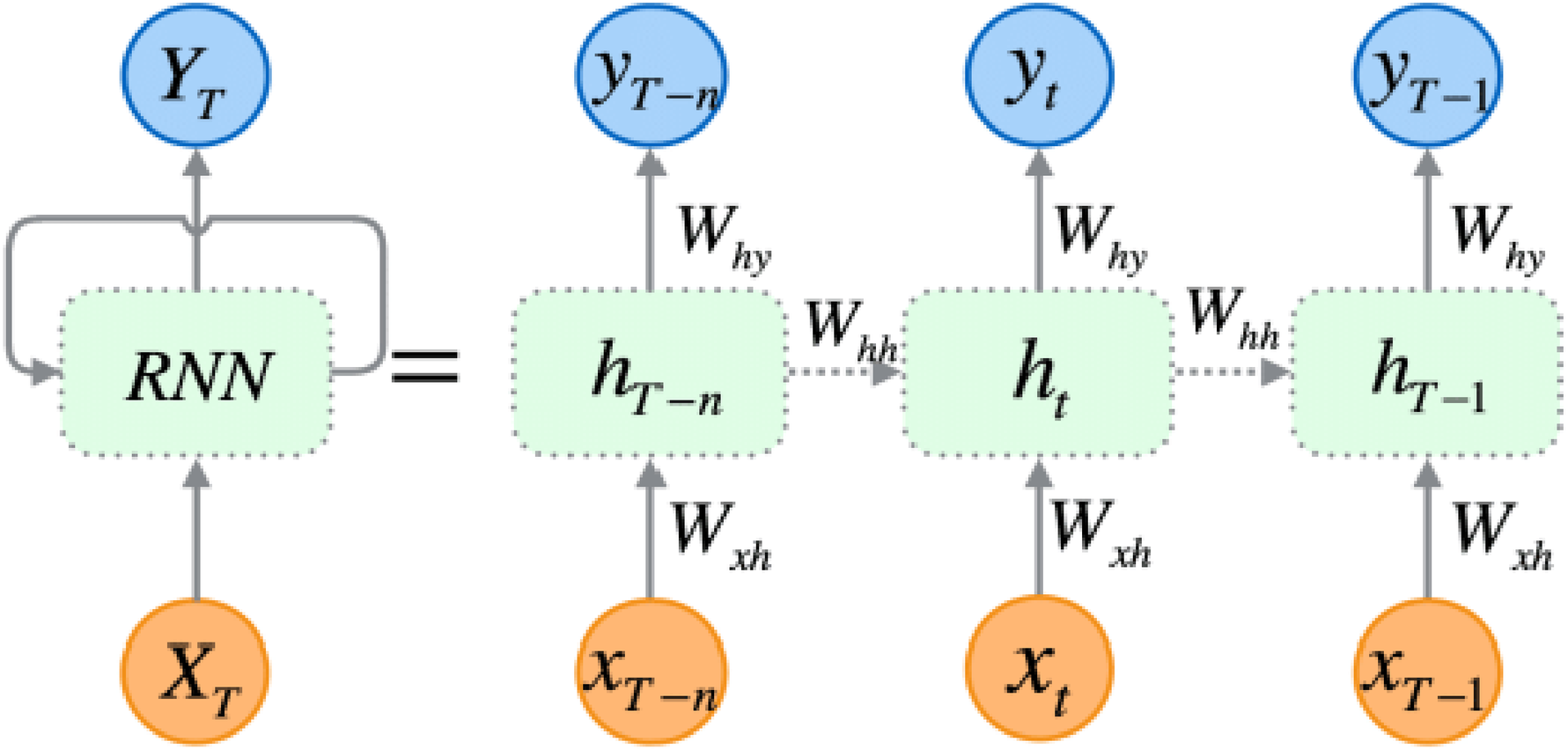
- Label each word in a sentence with its corresponding part of speech, such as noun (NN), verb (VB), adjective (JJ), adverb (RB), pronoun (PRP), preposition (IN), conjunction (CC), interjection (UH), Determiner (DT), etc.,

- Machine translation, sentiment analysis, information retrieval, and NER, POS tagging are essential

Part-of-Speech Tagging

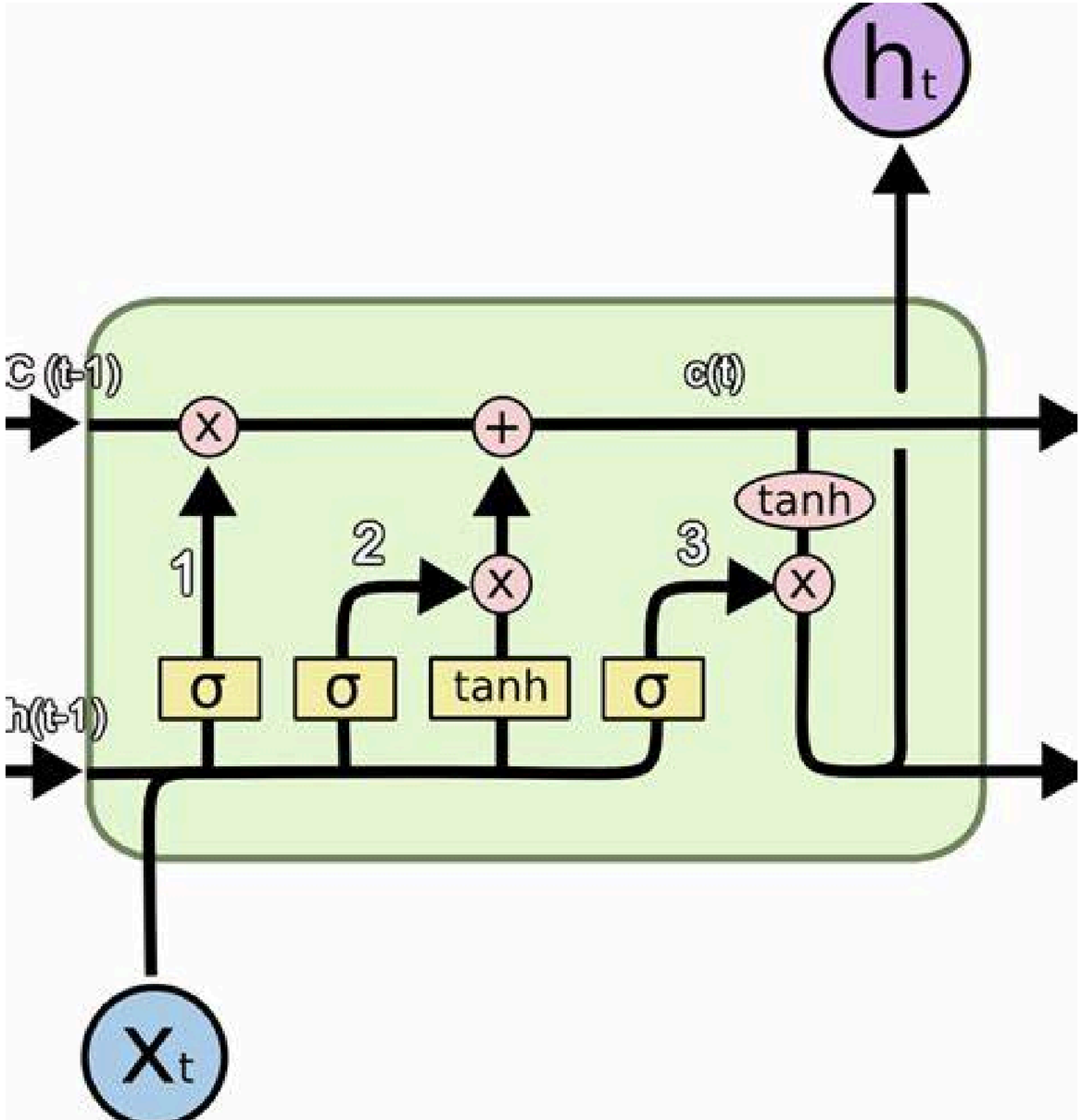


RNNs : RNN is a type of neural network designed to work with sequential data, like sentences or time series. The key feature is that it **remembers information from previous steps**



Prolem: vanishing gradient problem

LSTM : is a special type of RNN designed to solve the problem of remembering things over long sequences. It has a more complex internal structure that lets it **decide what to remember and what to forget** more effectively.



NLP vs LLM

By Mohammed Arbi Nsibi

Where to find pretrained LLMs ?

Hugging Face

By Mohammed Arbi Nsibi

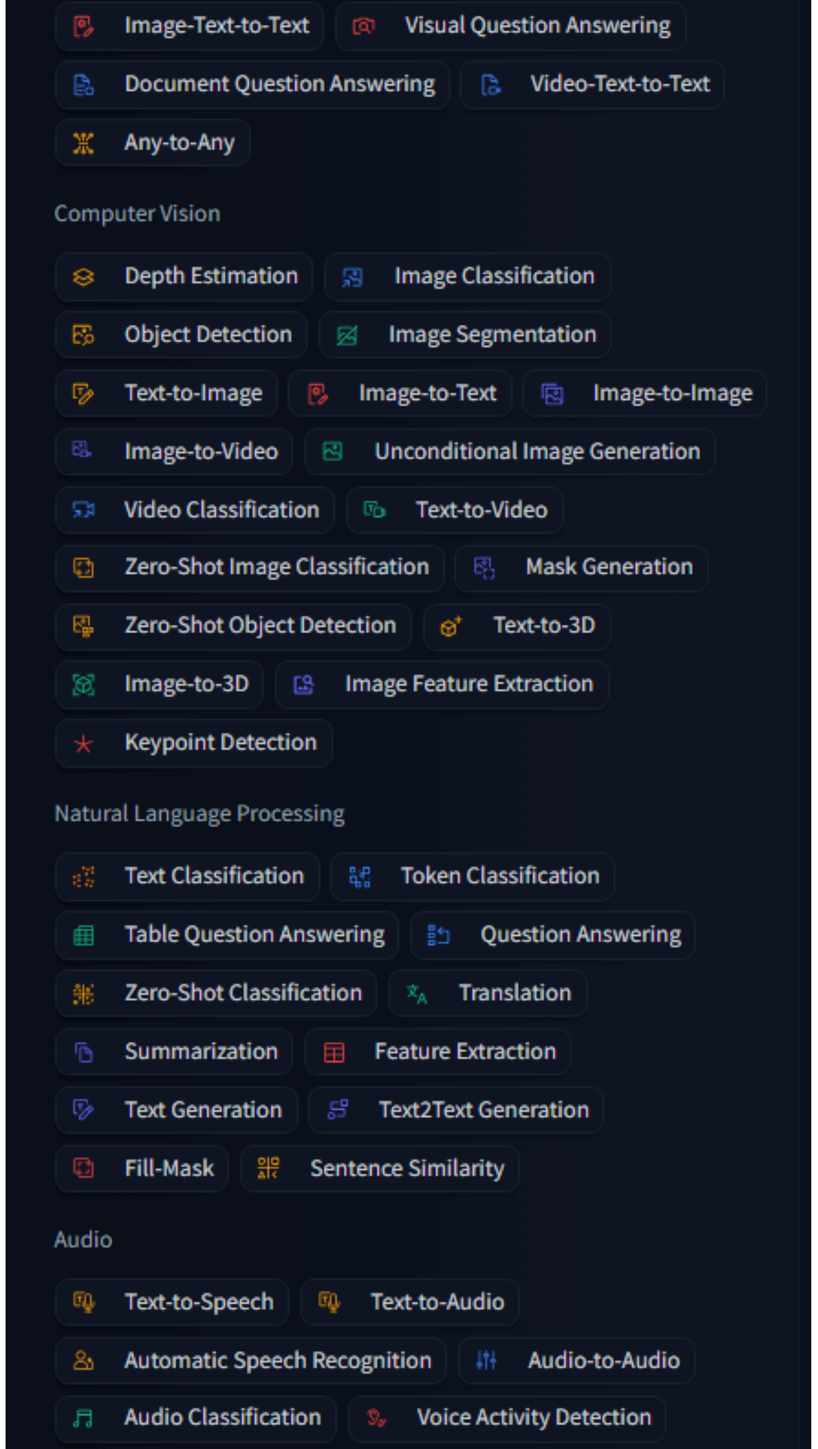
Model types

Text-to-text

Text-to-image

Text-to-video

Sentence-similarity



How to create my own application

```
import gradio as gr
from transformers import pipeline

pipe = pipeline("sentiment-analysis")

def predict(new_input):
  out = pipe(new_input)
  out = out[0]["label"]
  return out

gr.Interface(predict,inputs=["text"],outputs=["text"]).launch()
```

https://huggingface.co/learn/nlp-course/chapter1/3#working-with-pipelines

LET'S CODE 

Projects we should try for a better understanding of NLP

- Sentiment Analysis
- Question Answering System
- Named Entity Recognition
- Fake News Detection
- Topic Modeling
- Text Similarity
- Text summarization and machine translation
- Next word Prediction
- LLM applications using RAG
- Fine-tune a model for a specific NLP task
- Constructing your own LLM, inspired by models like Llama 2

Important Reading to know about NLP approaches

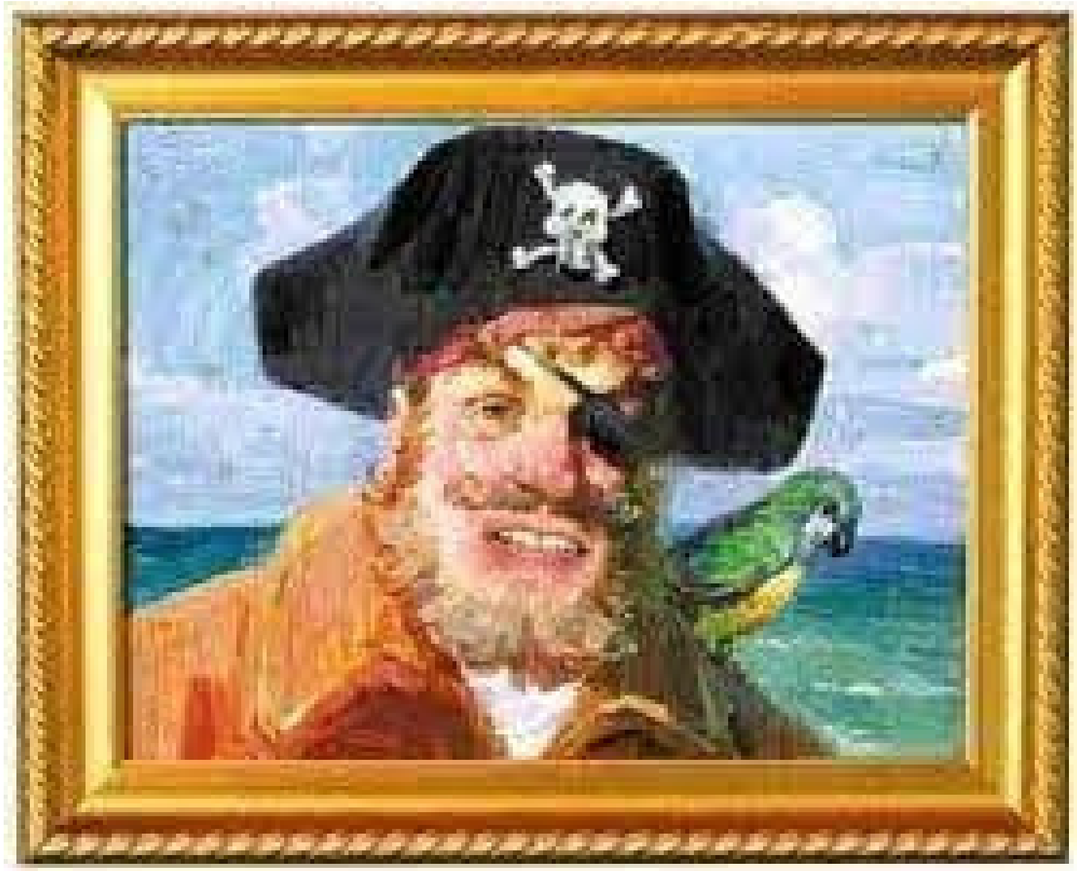
Level	Topic	Weblink
Beginning Level	RNN and LSTM	https://arxiv.org/pdf/1808.03314.pdf
	Word2Vec	https://arxiv.org/pdf/1301.3781
	Attention for Translation Task	https://arxiv.org/abs/1409.0473
	Attention is all you Need	https://arxiv.org/abs/1706.03762
	BERT	https://arxiv.org/abs/1810.04805
	GPT-1	https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf
	GPT-2	https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
	Research Papers	https://www.kaggle.com/discussions/general/236973
Advanced Level	RLHF	https://arxiv.org/abs/2203.02155
	LIama2	https://arxiv.org/abs/2307.09288
	PaLM2	https://blog.google/technology/ai/google-palm-2-ai-large-language-model/
	Mistral : Mixture of Experts	https://mistral.ai/news/mixtral-of-experts/
	GPT -4	https://openai.com/research/gpt-4
	Gemini AI	https://blog.google/technology/ai/google-gemini-ai

Q&A

By Mohammed Arbi Nsibi

THANK YOU for you attention!!

QUIZ TIME



By Mohammed Arbi Nsibi