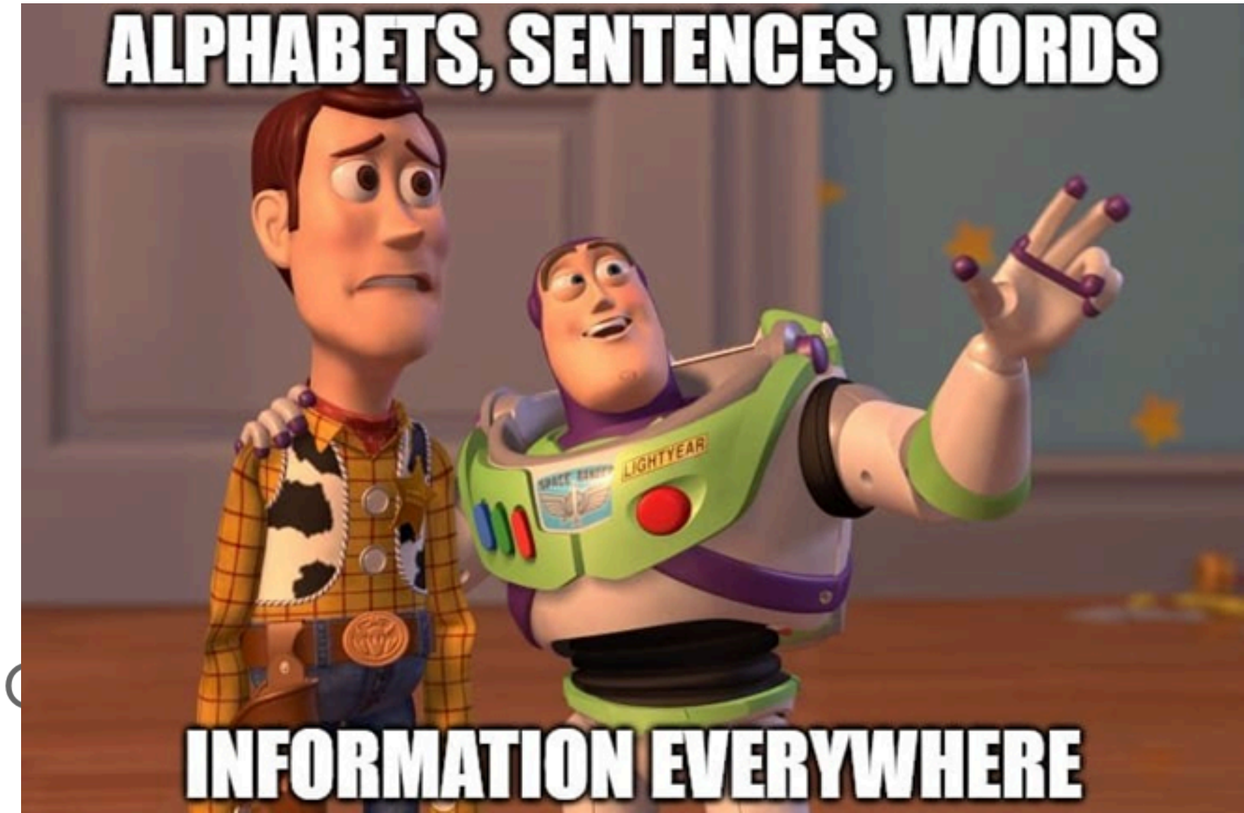
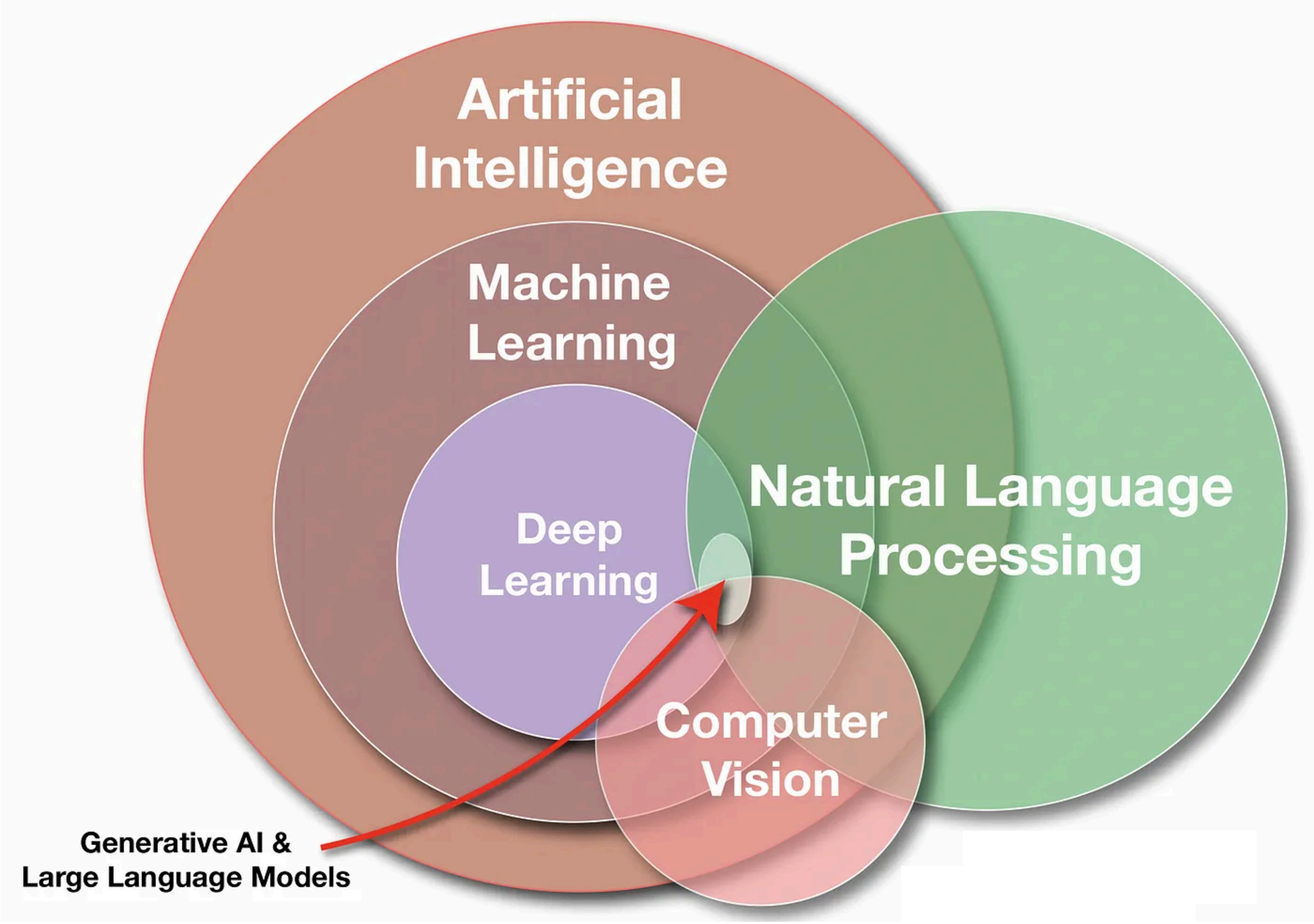
WHAT IS NLP ?

WHAT IS NLP ?

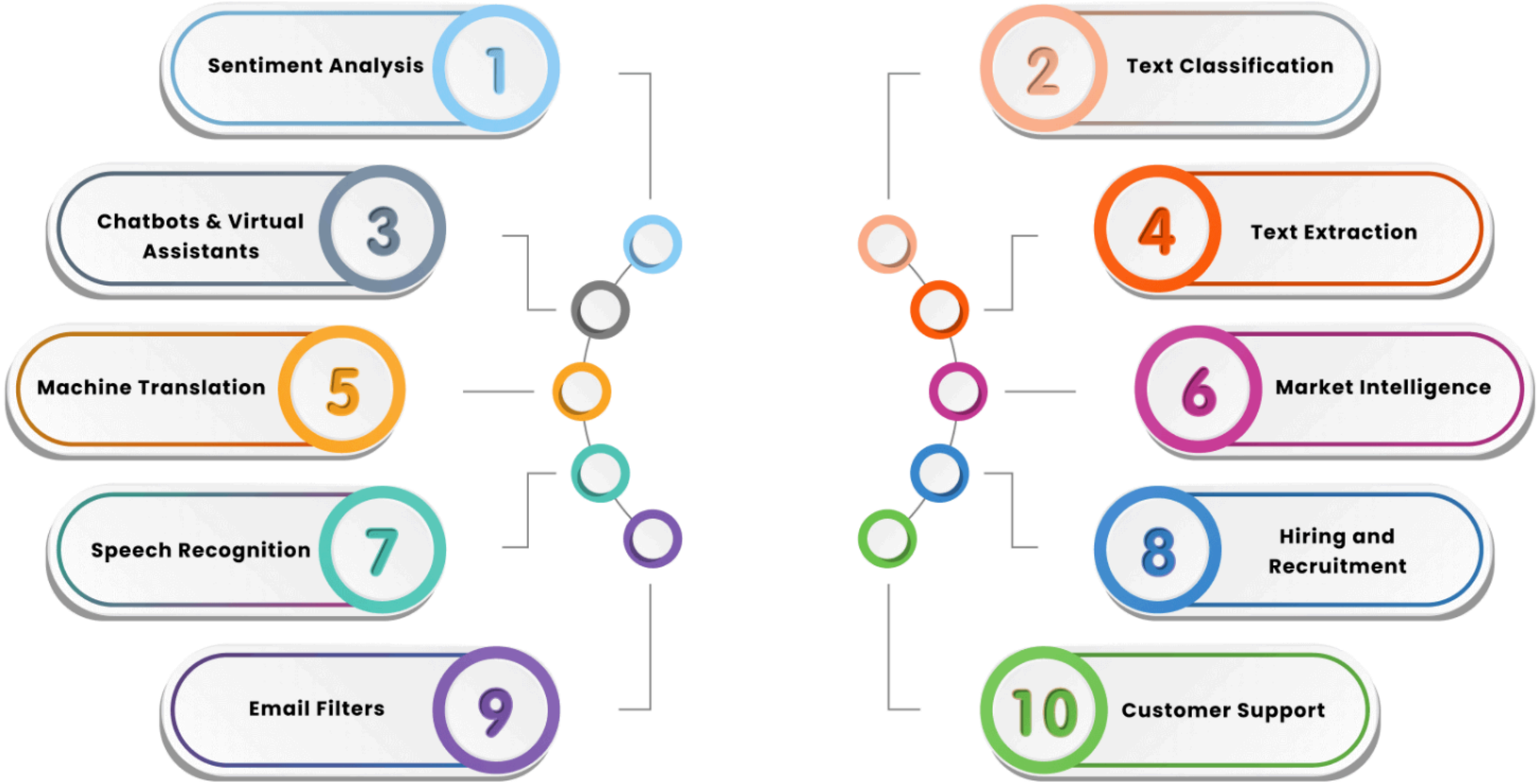
is referred to as NLP. It is a subset of AI that enables machines to comprehend and analyze human languages. Text or audio can be used to represent human languages.



GDG C

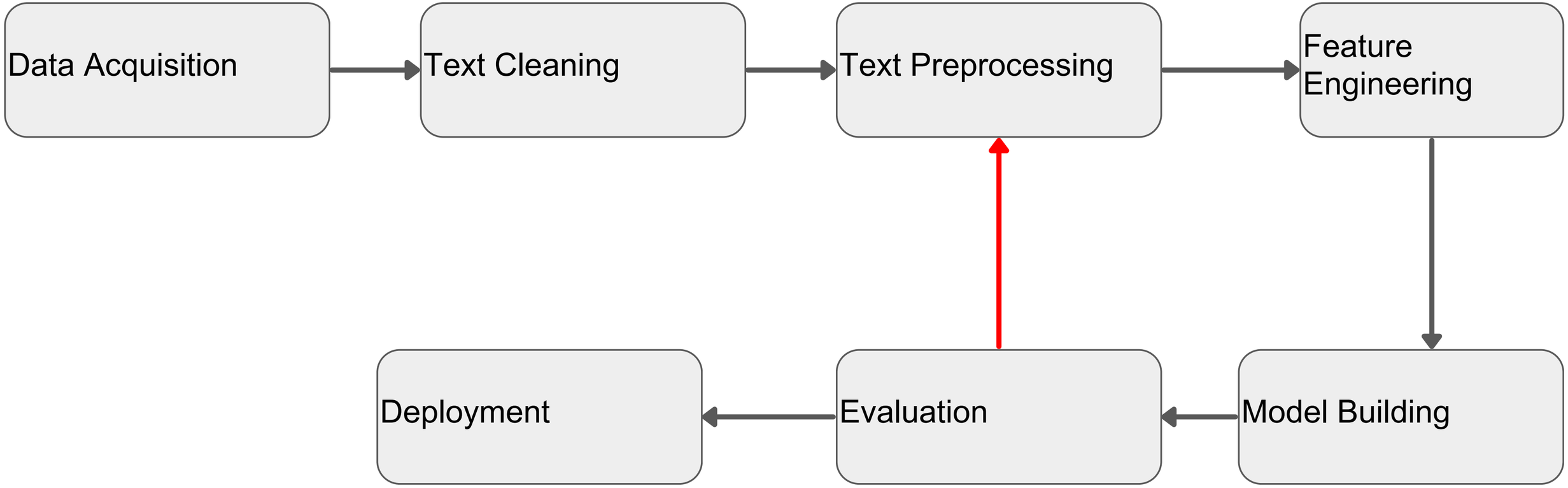

Artificial Intelligence
Machine Learning
Deep Learning
Natural Language Processing
Computer Vision
Generative AI & Large Language Models

NLP applications



GDG Carthage

NLP Pipeline



GDG Carthage

Text Cleaning

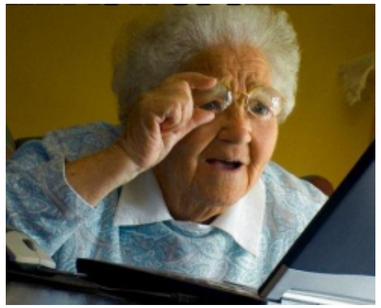
Regex or Regular Expression

Spelling corrections

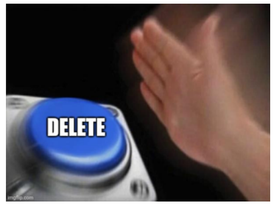
used for searching the string of specific patterns. Suppose our data contain phone number, email-Id, and URL. we can find such text using the regular expression. After that either we can keep or remove such text patterns as per requirements.

GDG Carthage

Text Preprocessing



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



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

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

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

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

Traditional Approach

- 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?

GDG Carthage

How can we get these word embedding vectors?

Train our own embedding layer:

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

Pre-Trained Word Embeddings

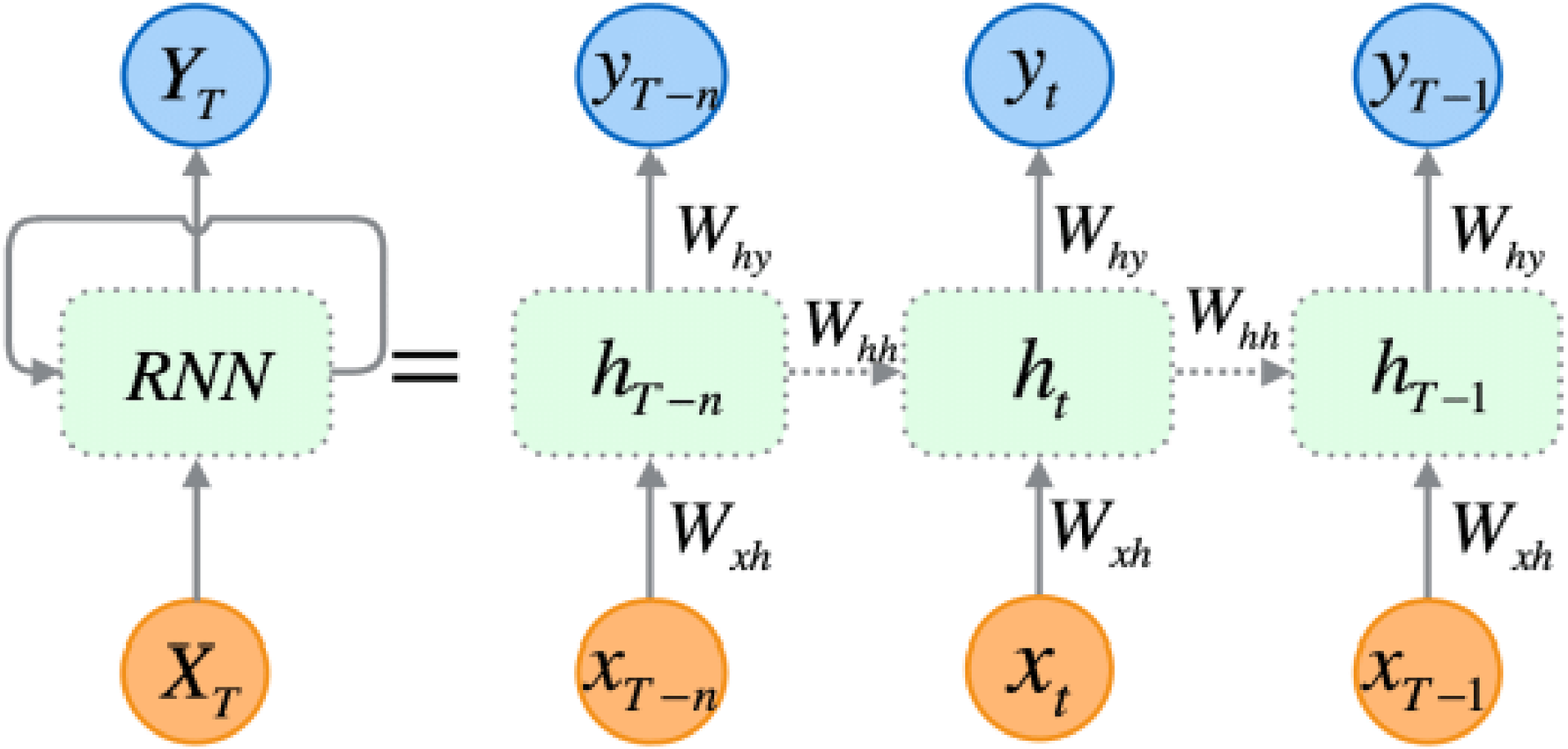
- **Word2vec by Google 2013**
- **GloVe by Stanford**
- **fasttext by Facebook**

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

LET'S CODE 

GDG Carthage

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.