# Natural Language Processing

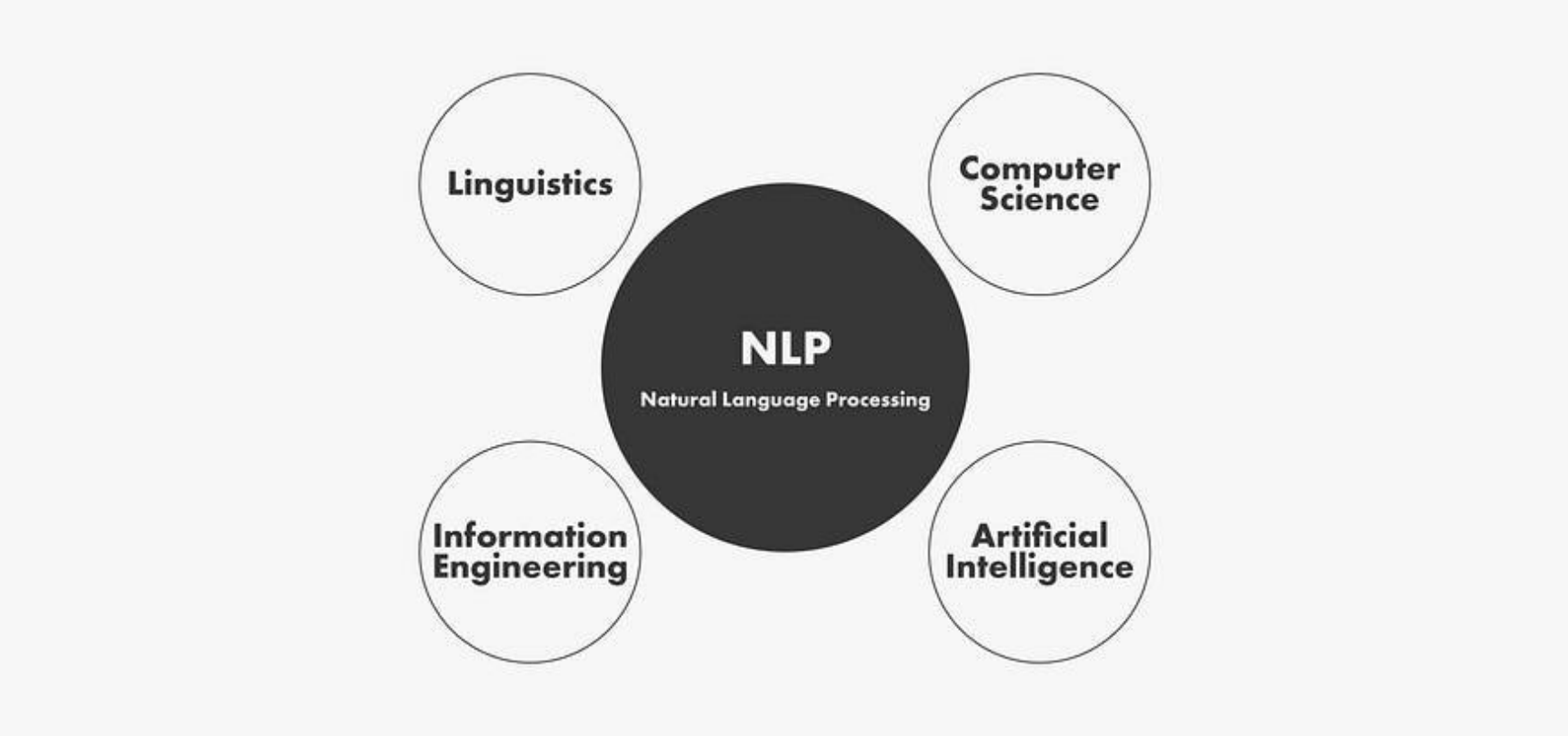
AIGC 5501

Introduction

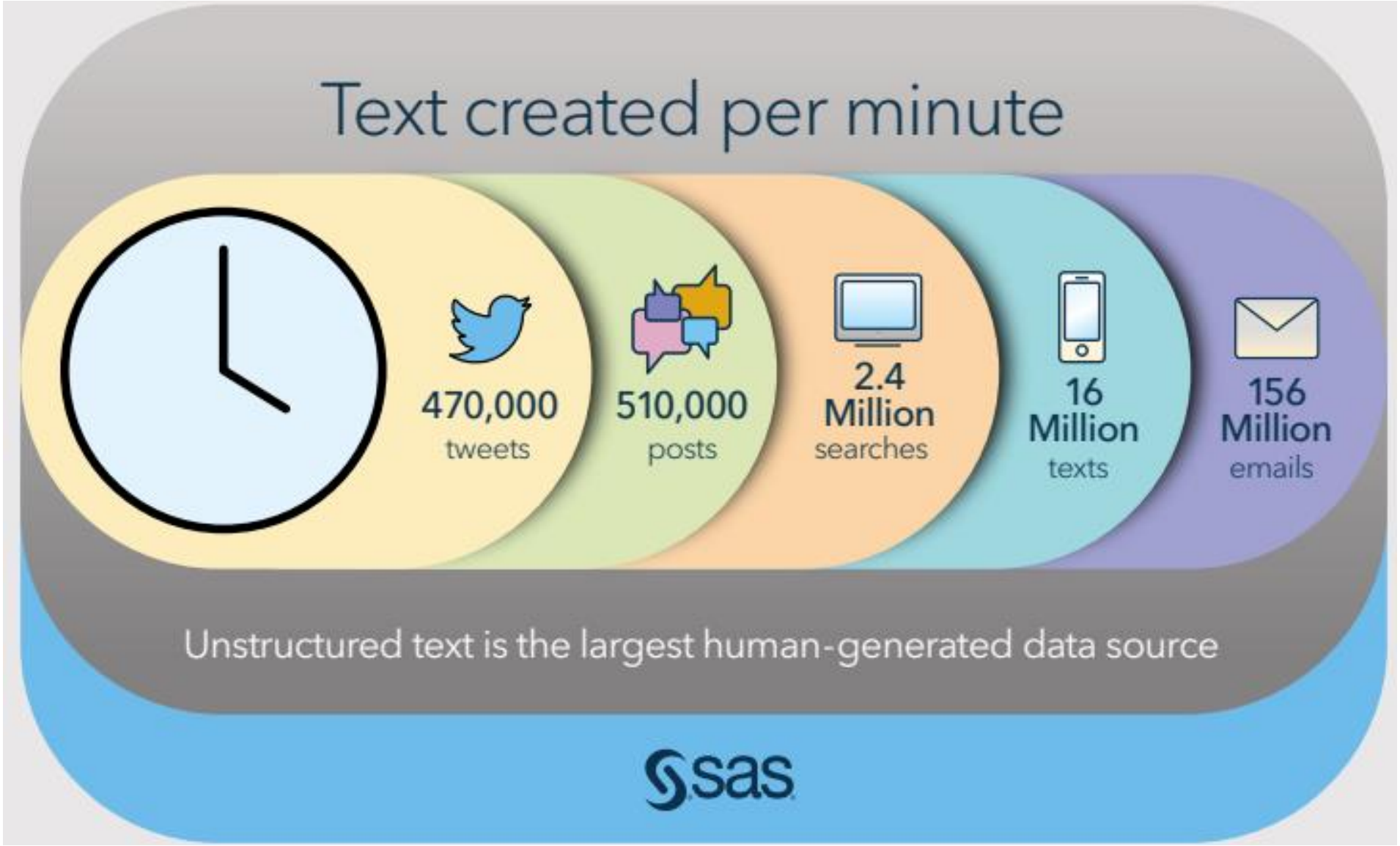
Instructor: Ritwick Dutta
Email: ritwick.dutta@humber.ca

# What is NLP?

Natural language processing (NLP) is a branch of artificial intelligence that helps computers **understand**, **interpret** and **manipulate** human language. NLP draws from many disciplines, including **computer science** and **computational linguistics**, in its pursuit to fill the **gap** between human communication and computer understanding.



# Current State

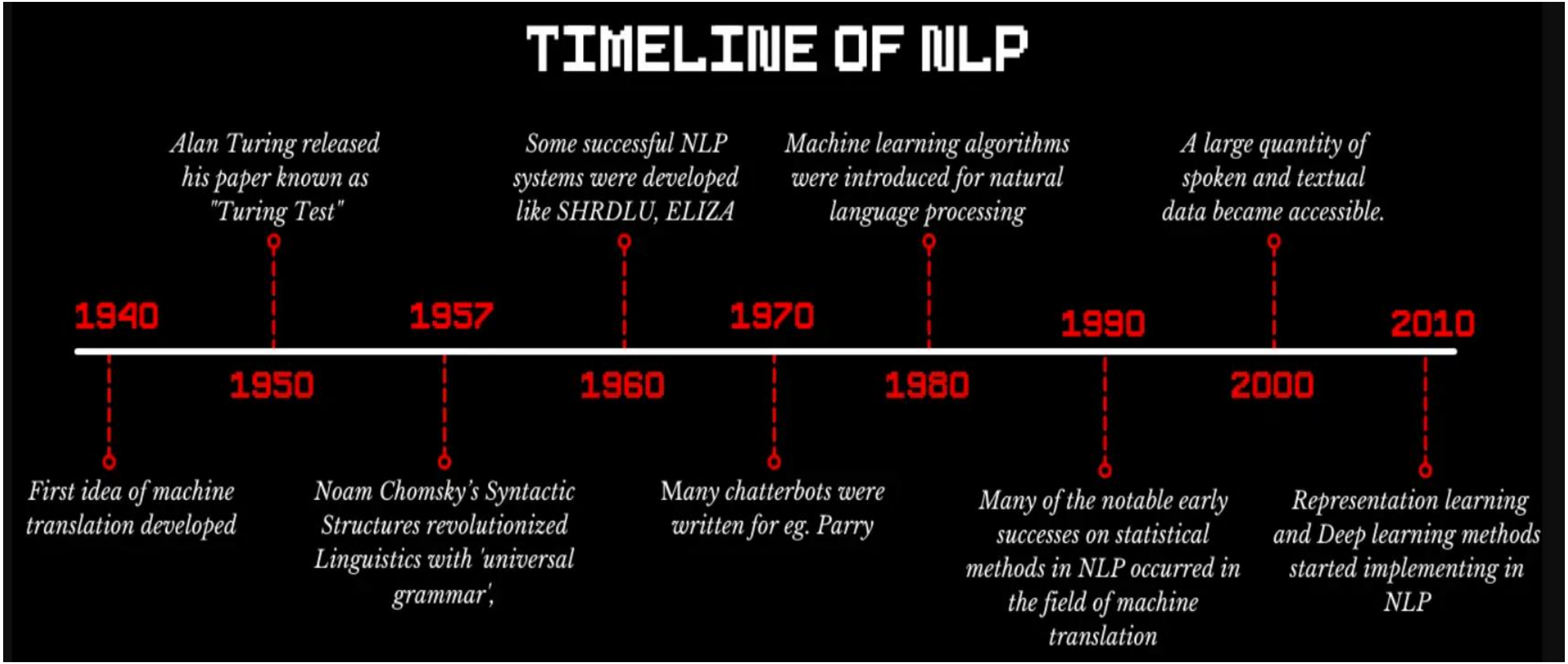


# Natural Language Processing (NLP) — Why NLP ?

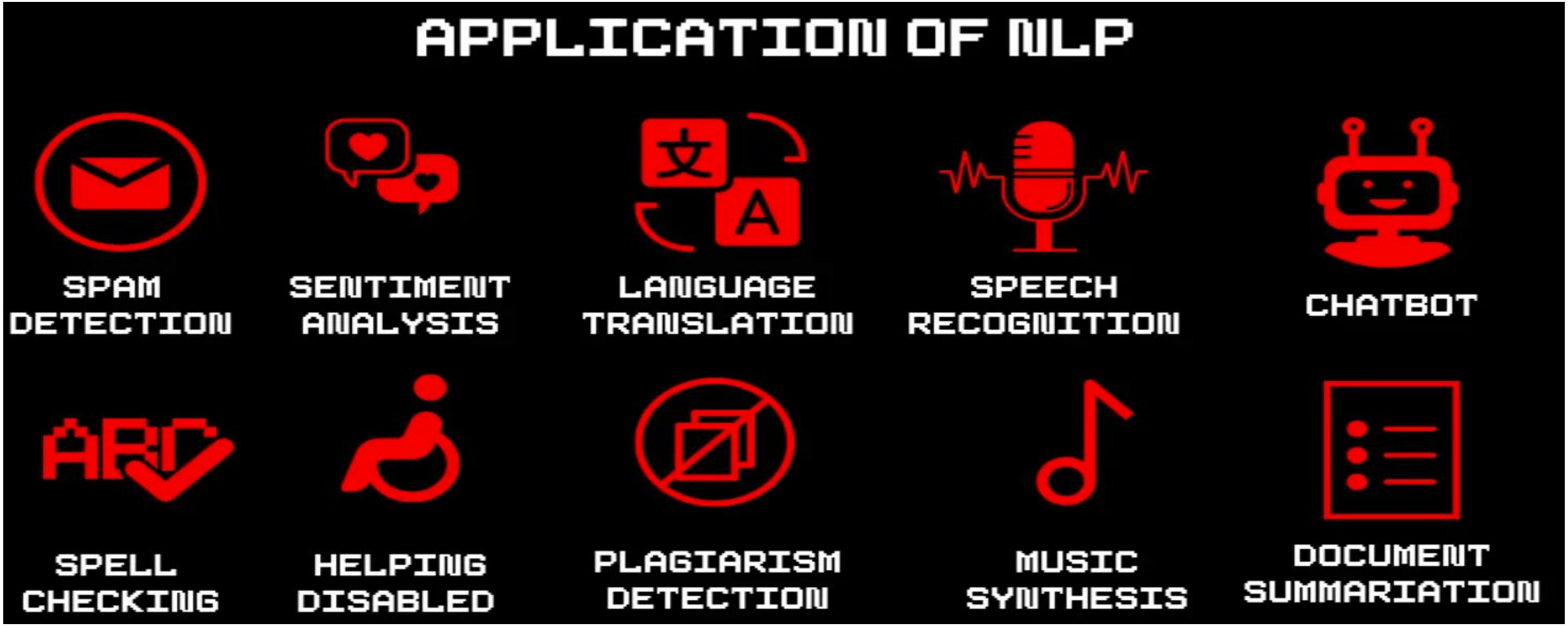
NLP is about making computers **understand** how humans speak and **communicate** with each other in order to help the computers to engage in communication using natural **human language** in many forms, including but not limited to speech and writing.

We **talk to each other in a number of different ways**, whether it's through different grammatical structures or the regional idioms we use, so it's essential that NLP is in place so that **this gap in communication can be bridged**.

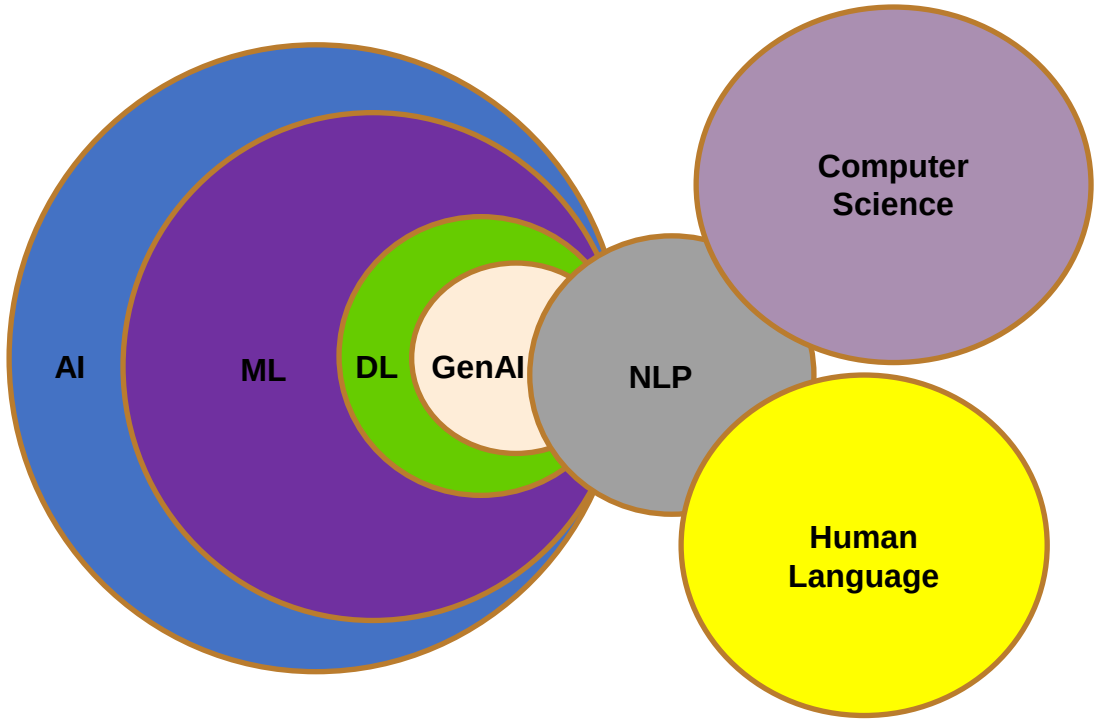
# History of Natural language processing

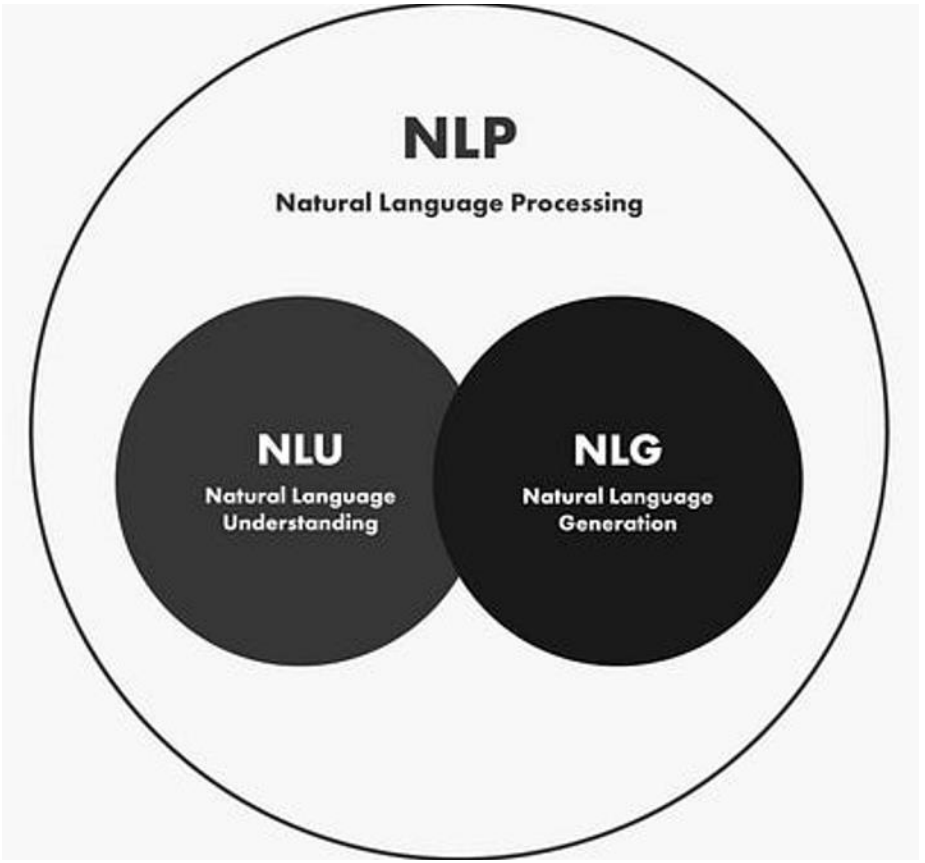


# Applications of NLP

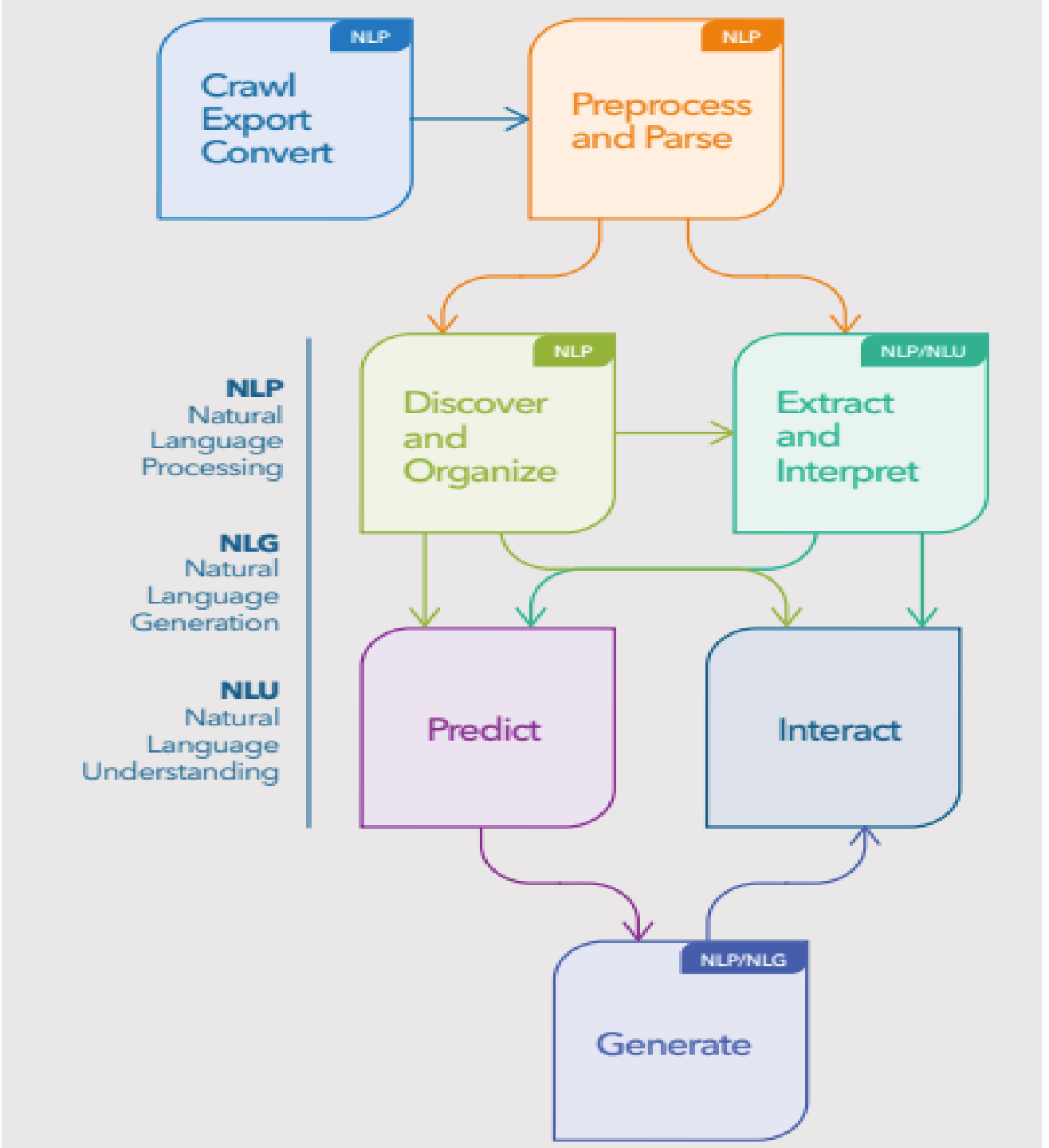


# Components of NLP



# Stages of Text Processing



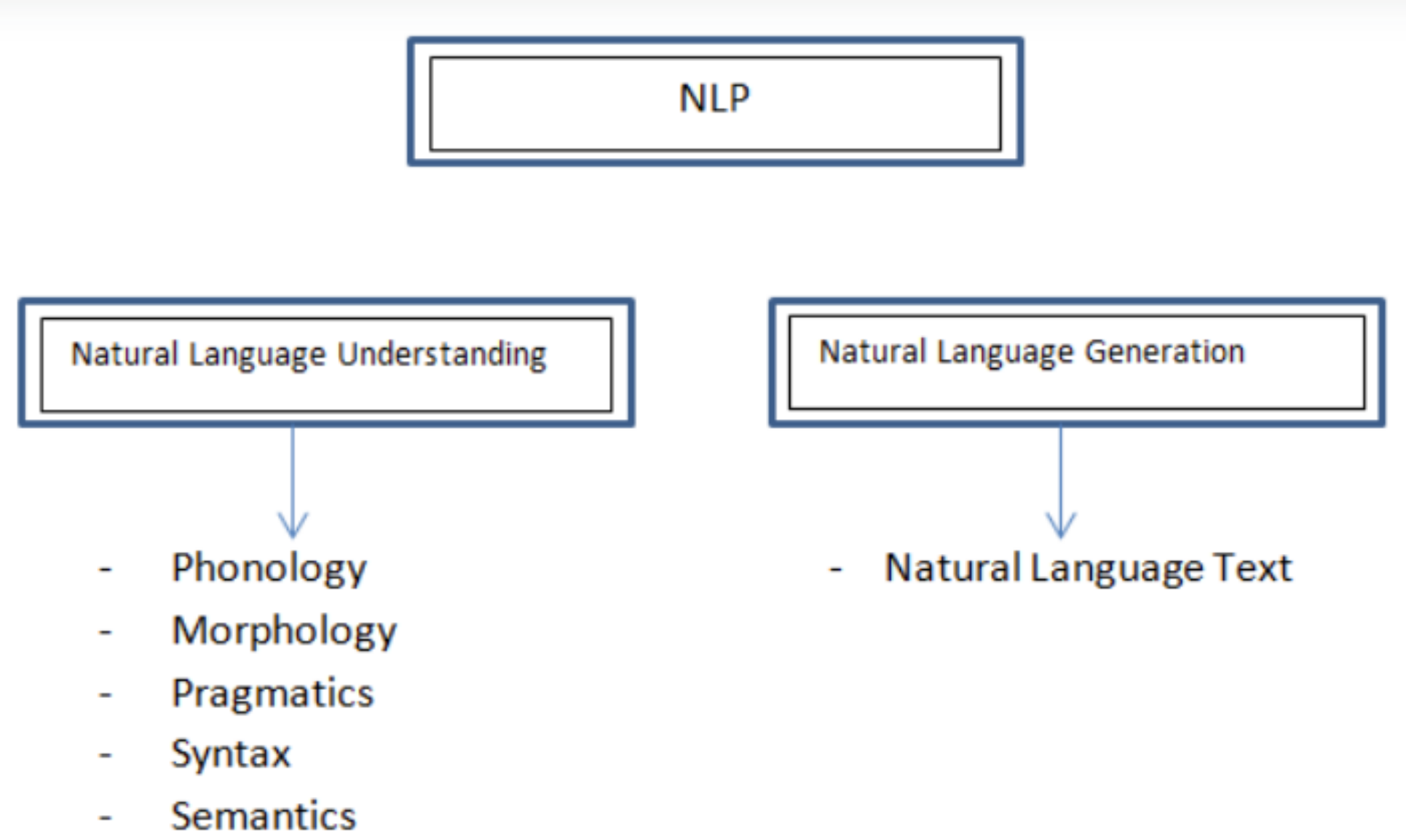
# Natural Language Understanding (NLU)

NLU or Natural Language Understanding is a subset of NLP that is responsible for the **semantic parsing and analysis, entities extracting…etc**. NLU tries to structure the input data to understand the meaning behind written text.

# Natural Language Generation (NLG)

**NLG** or **Natural Language Generation** is the step when the machine tries to transform the structured (from NLU) data into human-readable language, so NLG does exactly the opposite of NLU.

# Broad Classification of NLP



# Natural Language Understanding

**Phonology** refers to the science of understanding sound
**Morphology** refers to word formation
**Syntax** refers to the structure
**Pragmatics** refers to understanding

When humans talk to each other, the first thing that other humans do is **understand the context**. Later **formulate the response** accordingly that makes sense.

This is what the two terms try to say with **Natural language understanding**, it means to understand the context, and **Natural language generation** relates to sensible response to the context.

# Different meaning of any thing

•**Lexical (Word Level):** Lexical work at the word level; imagine any word used as **a verb** and used as **a noun**. These are crucial to deciding for NLP. E.g. BAT (Mammal/action of striking)

•**Syntactical (Parsing):** Parsing is a kind of synonym for syntactical as per NLP is concerned. E.g., "Call me a cab" this sentence has two implications if you think.

One is a request to get a cab while the other implementation says;

My name is cab so call me a cab

This is syntactical, which lays its role at a sentence level.

•**Referential:** Let see a new scenario to understand this better. "Alex went to Dave; he said that he was hungry". This is just an explanation statement to demonstrate how complex the interpretations can be for the computers to understand in their initial NLP phase. So, in the above statement, the confusion for a computer to understand two he's is meant for which person (means Alex or Dave).

# Natural Language Generation

So, the machine has understood that we asked them to do something, now come to their turn to provide a proper response or feedback. NLG does the same thing.

**Text Planning:** This means plain text from the knowledge base, just like we humans have a vocabulary that helps us to frame sentences.

**Sentence Making:** To arrange all the words and make an arrangement in a meaningful pattern.

**Text Realization:** To process all the sentences in a proper sequence or order and give the output is called text realization.

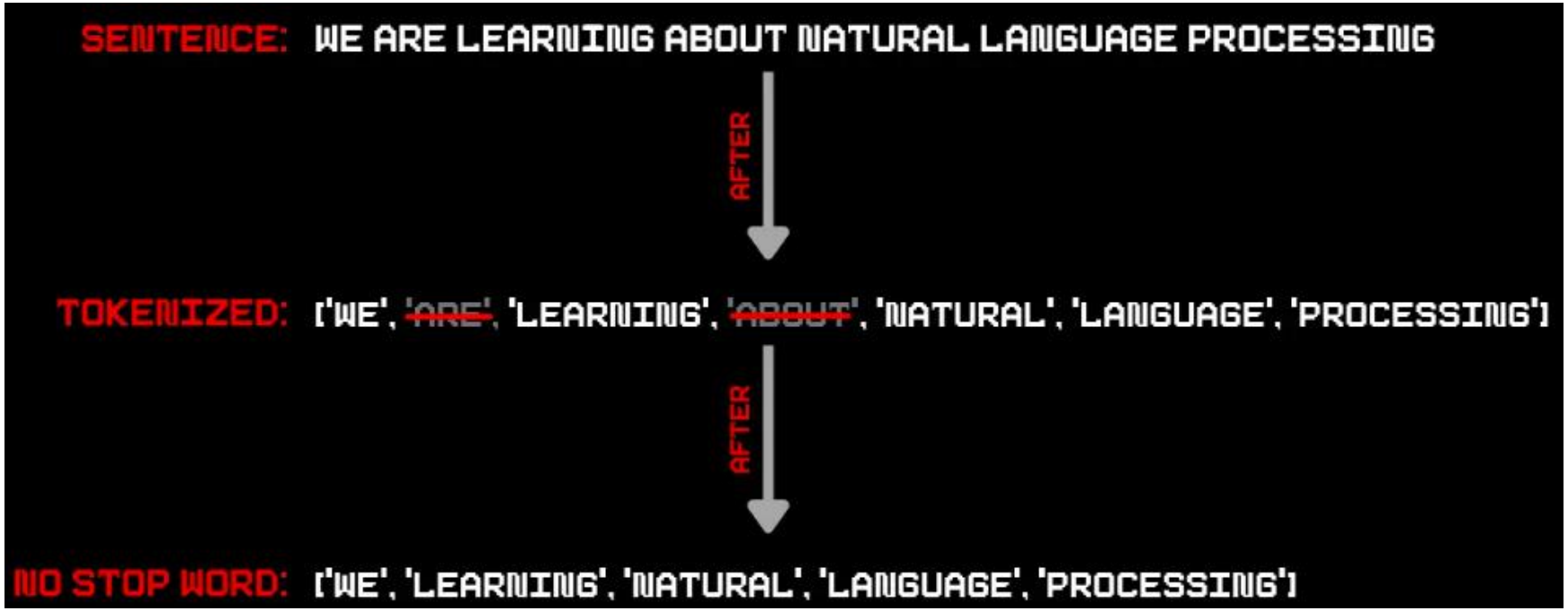
# Steps to understand any sentence or document

**1. Tokenization:** We separate each individual word, punctuation marks, etc from a sentence/phrase/document, this process is known as Tokenization.
These all-separated words from a text or documents are together called **bags of words**.

So, remember "**tokenization**" is a term for the **process of separating words** while "**bag of words**" is the term for its **output**.

# Steps to understand any sentence or document

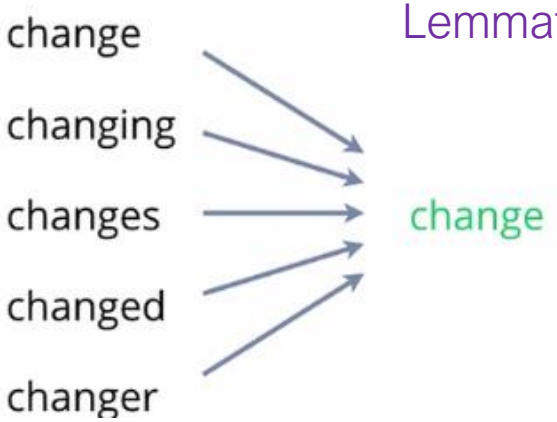
**2. Stop words:** After separating each word from the sentence we remove the word or element which do not add any value to the sentence like, numbers, punctuation marks, URLs, and stop words (for example, in English, “the”, “is” and “and”,). This not only saves processing power but also increases model accuracy significantly.



# Steps to understand any sentence or document

**3. Stemming/Lemmatization:** Words like "eating", "eat", "ate" or "better", "good", "best" are forms of the same word "eat" and "good". This different form of a word means the same at some level and doesn't add some extra value to a sentence. Keeping this increases the processing power and decreases the accuracy so we convert it to its root word like for running we convert it back to run.

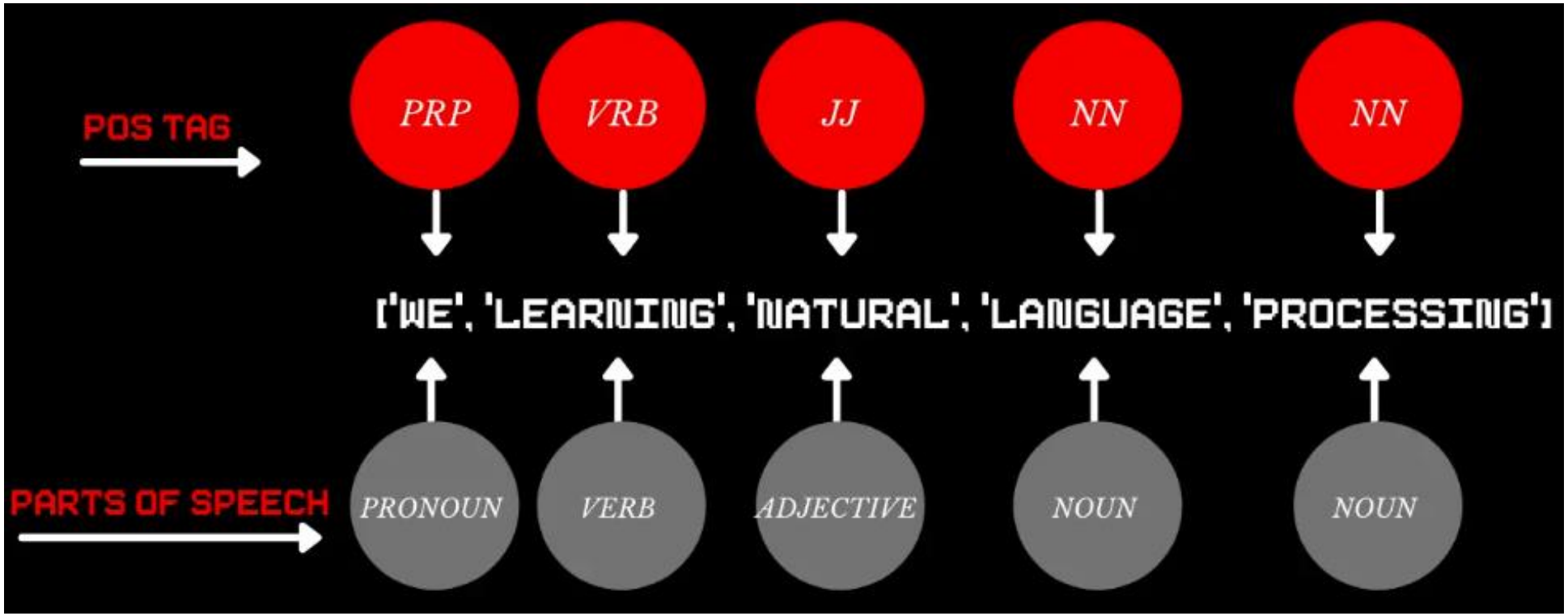
## Stemming vs Lemmatization


Stemming - reduces words to their word root.
Lemmatization - reduces words to their base word

# Steps to understand any sentence or document

**4. POS Tagging (Parts of Speech tagging):** Parts of speech tagging, also known as grammatical tagging or word category disambiguation. It's used to categorize words in a text or a document to their corresponding parts of speech based on both its definition and its context (with its adjacent and related words in a phrase). For example, see the image below.



# Steps to understand any sentence or document

**5. Extracting information:** After performing all the previous steps (pre-processing), we finally head to extract meaningful insight from the text or a document. These all steps are performed with the help of various python libraries which helps in all text pre-processing and information extraction.

# Common Terminologies in Natural language processing

**1.Corpus or corpora:** It is a term used for the collection of text such as product review, conversation, etc.

**2.Lexicons**: It contains the meaning or vocabulary of different words; you can think of it as your dictionary.

**3.Word embedding**: It is a term used for the representation of words for text analysis, basically, it is a mathematical representation of words as a vector created by analyzing a corpus or document and representing each word, phrase, or entire document as a vector in a high-dimensional space.

**4.Word2vector**: It is a technique for creating such word embeddings

**5.N-grams**: It is basically an "n" numbers of sequence of the word for example a single word "run" we call it "uni-gram" (not 1-gram) we use Latin numerical prefixes, and for two words it is "bi-gram" for 3-word "tri-grams, and for four words "four-gram"…."five-gram" and so on.

# Common Terminologies in Natural language processing

**6. Normalization**: It is all about putting text in a standard format, for example putting all text in either upper or lower case or transforming similar text like “goooood” or “gud” to “good”.

**7. Named Entity Recognition:** It helps you easily identify key elements of the text like names of people, places, brands, etc.

**8. Transformers:** It is an architecture introduced in 2017 which solves sequence – to – sequence tasks while handling long-range dependencies with ease.

# What topics will we cover?

- Introduction to Natural Language Processing
- Language Modeling/N-grams, Text Processing / Classification
- Sequence Tagging / HMMs / Viterbi / MEMMs, Viterbi Step-Through / NER Intro
- Chunking / SRL, Word Embeddings / Vector Semantics
- FFNNs, Designing NNs
- Pytorch tutorial , Training NNs
- Neural Language Models / Recurrent Neural Networks, RNNs and Bidirectionality
- Backprop, LSTMs
- Encoder-Decoders, Attention / Contextualized Embeddings / ELMo
- Transformers / BERT, Discourse, Implementing Transformers
- Coreference, Summarization, NN Probing / Interpretability
- LLMs

+ machine learning methods
– Statistical
– Probabilistic
– Neural networks

# Top NLP Python Libraries

- NLTK- Natural Language Toolkit
- SpaCy
- Genism
- CoreNLP
- TextBlob
- PyNLPl
- Polyglot
- Pattern
- Scikit-Learn
- Flair
- Vocabulary
- Pytorch

# NLTK- Natural Language Toolkit

Python NLTK is the most used Python NLP library, an open-source NLP library. POS tagging, phrase frequencies, NLTK sentiment analysis, etc. are the most powerful functions that it offers.

NLTK's user-friendly interfaces provide more than 50 linguistics assistants such as WordNet, corpora, linguistic, etc. Where those textual content processing libraries are used for class, tokenization, stemming, and so forth.

Tokenize and tag some text:

```
>>> import nltk
>>> sentence = """At eight o'clock on Thursday morning
... Arthur didn't feel very good."""
>>> tokens = nltk.word_tokenize(sentence)
>>> tokens
['At', 'eight', "o'clock", 'on', 'Thursday', 'morning',
'Arthur', 'did', "n't", 'feel', 'very', 'good', '.']
>>> tagged = nltk.pos_tag(tokens)
>>> tagged[0:6]
[('At', 'IN'), ('eight', 'CD'), ("o'clock", 'JJ'), ('on', 'IN'),
('Thursday', 'NNP'), ('morning', 'NN')]
```

For more information, check out the official document of Natural Language Toolkit. (https://www.nltk.org/)

# SpaCy

The library comes up with spaCy projects like entity recognition, pre-trained statistical model, dependency parsing, text classification, word vector, tokenization deep learning integration, etc. It's spaCy sentiment analysis this tool emphasizes art speed and accuracy, CNN models for tagging and translating.

Along with the compelling features & intuitive interface, spaCy boasts the "industrial-strength".

```
# pip install -U spacy
# python -m spacy download en_core_web_sm
import spacy

# Load English tokenizer, tagger, parser and NER
nlp = spacy.load("en_core_web_sm")

# Process whole documents
text = ("When Sebastian Thrun started working on self-driving cars at "
        "Google in 2007, few people outside of the company took him "
        "seriously. “I can tell you very senior CEOs of major American "
        "car companies would shake my hand and turn away because I wasn’t "
        "worth talking to,” said Thrun, in an interview with Recode earlier "
        "this week.")
doc = nlp(text)

# Analyze syntax
print("Noun phrases:", [chunk.text for chunk in doc.noun_chunks])
print("Verbs:", [token.lemma_ for token in doc if token.pos_ == "VERB"])

# Find named entities, phrases and concepts
for entity in doc.ents:
    print(entity.text, entity.label_)
```

Get more on the official **spaCy documentation**!

# Lab -1

**Getting Familiar**

Install Jupyter Notebook / Anaconda
Python Demo