INTRO TO NLP

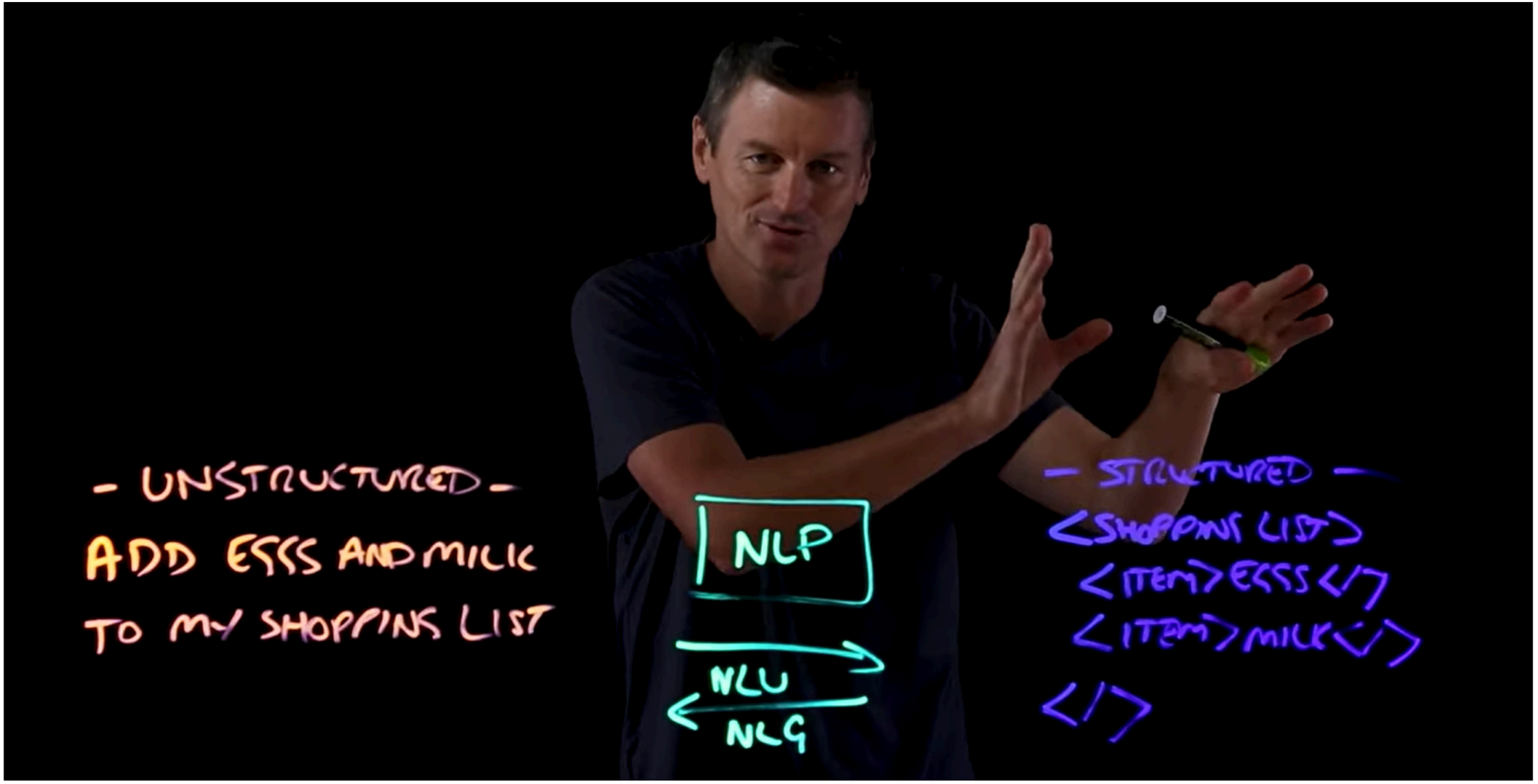
Architecting Intelligence

What is NLP

Natural language processing is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data.

Need for NLP

In neuropsychology, linguistics, and the philosophy of language, a **natural language** or **ordinary language** is any language that has evolved naturally in humans through use and repetition without conscious planning or premeditation. Natural languages can take different forms, such as speech or signing. They are distinguished from constructed and formal languages such as those used to program computers or to study logic


- UNSTRUCTURED -
ADD EGGS AND MILK
TO MY SHOPPING LIST
NLP
NLU
NLG
- STRUCTURED -
<SHOPPING LIST>
<ITEM> EGGS </>
<ITEM> MILK </>
</>

Natural Language Understanding

Interpreting the meaning of the text.

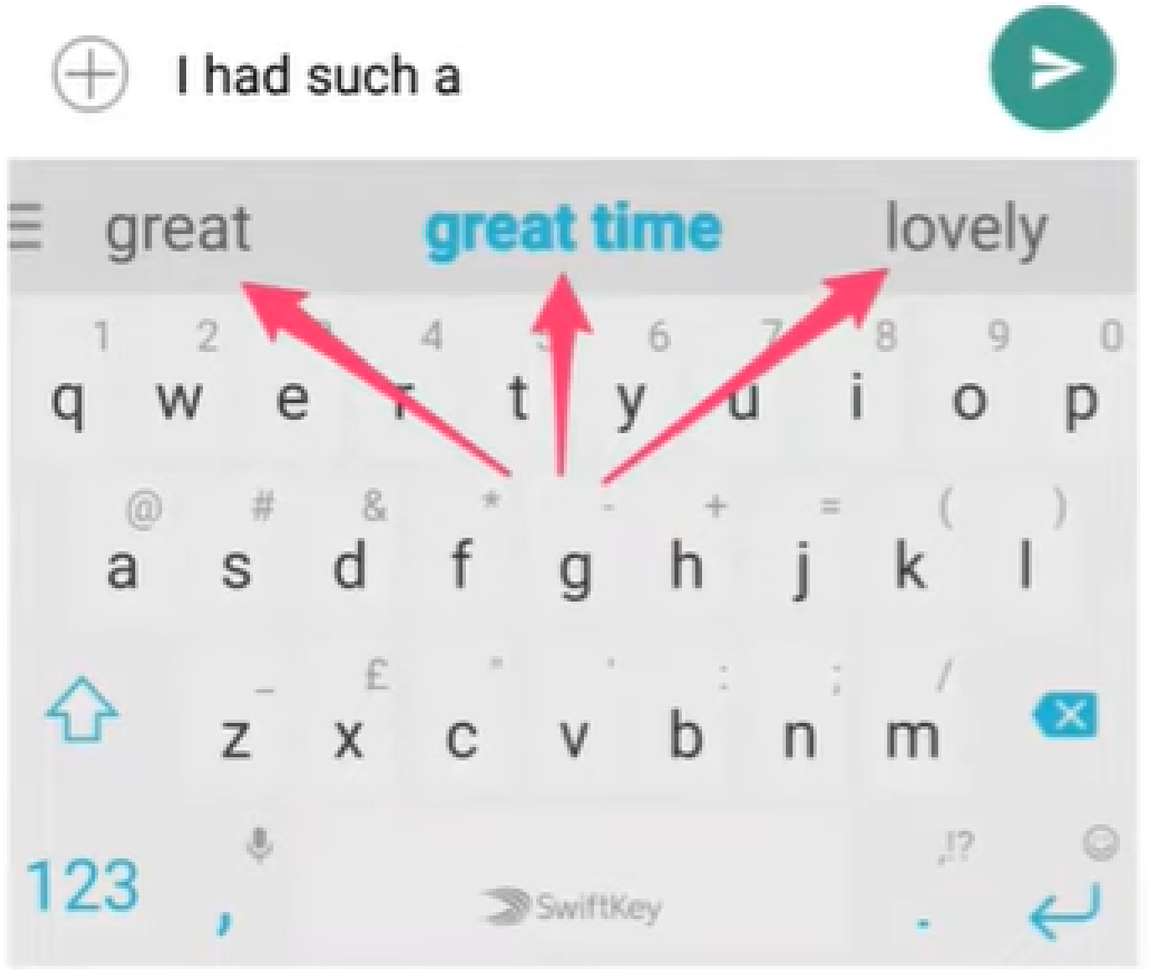
NLP Components

Natural Language Generation

Creating human-like text based on data.

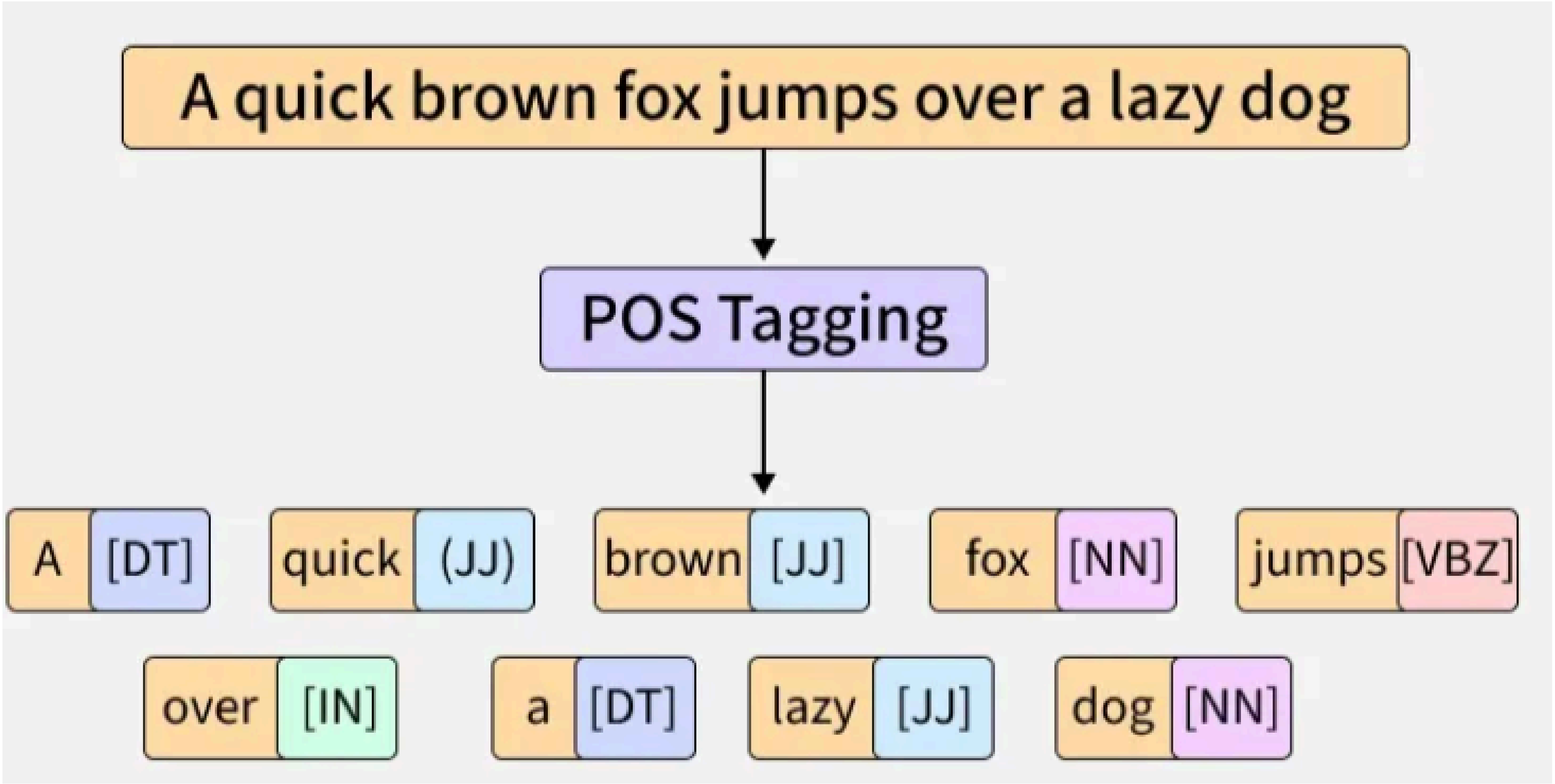
Applications of NLP

1. **Voice Assistants:** Alexa, Siri and Google Assistant use NLP for voice recognition and interaction.
2. **Grammar and Text Analysis:** Tools like Grammarly, Microsoft Word and Google Docs apply NLP for grammar checking.
3. **Information Extraction:** Search engines like Google and DuckDuckGo use NLP to extract relevant information.
4. **Chatbots:** Website bots and customer support chatbots leverage NLP for automated conversations.

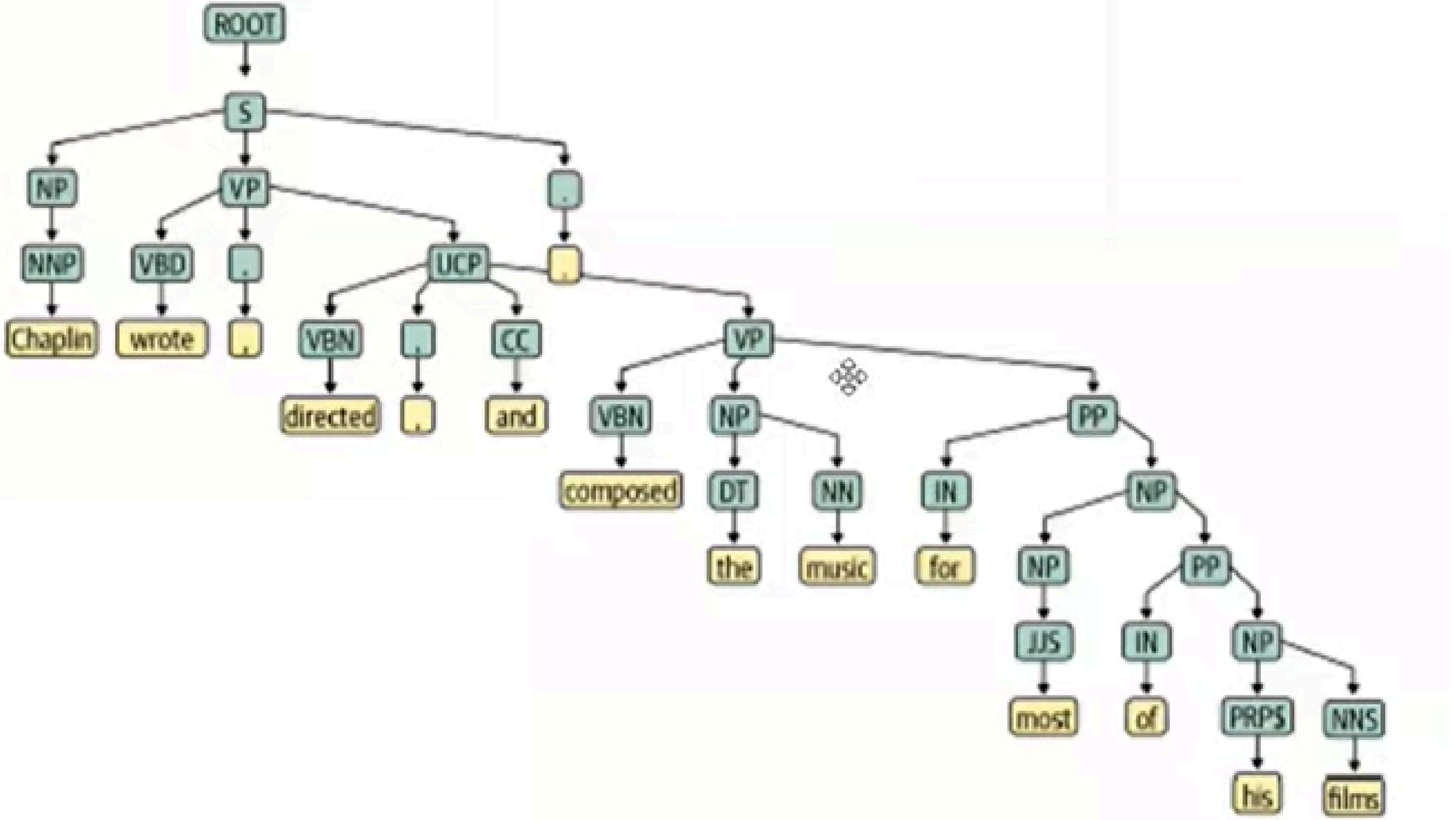


Text Generation

Speech To
Text


A quick brown fox jumps over a lazy dog
POS Tagging
A [DT]
quick (JJ)
brown [JJ]
fox [NN]
jumps [VBZ]
over [IN]
a [DT]
lazy [JJ]
dog [NN]

Parse Tree


ROOT
S
NP
VP
.
NNP
VBD
.
UCP
.
Chaplin
wrote
,
VBN
,
CC
VP
directed
,
and
VBN
NP
PP
composed
DT
NN
IN
NP
the
music
for
NP
PP
JJS
IN
NP
most
of
PRP$
NNS
his
films

Challenges in NLP

Ambiguity

I saw the boy on the beach with my binoculars.
I have never tasted a cake quite like that one before!

Contextual Words

*I **ran** to the store because we **ran** out of milk.*

Colloquialisms and slang

Piece of cake, pulling your leg

Synonyms

Irony, Sarcasm and tonal diff

That's just what I needed today!

Spelling Errors

Creativity

Poems, dialogue, scripts

Diversity

Pipeline followed for building NLP Applications

Data Acquisition

Text Preparation

Text Cleanup
Basic Preprocessing
Advance Preprocessing

Feature Engineering

Modelling

Model Building
Evaluation

Deployment

Deployment
Monitoring
Model Update

Step 1

EDA (Exploratory Data Analysis), primary processing, and data visualization :

- **How does the data look?**
- **Are there any duplicate entries? If yes, then what to do?**
- **Are there any missing entries? If yes, then what to do?**
- **Is the data inconsistent somewhere? Is there any other value apart from 0/1 in the sentiment column.**
- **What is the distribution between positive and negative reviews? Is the data imbalanced or not?**
- **What does the length of reviews vs the sentiment graph look like?**

Step 2

Data Pre-Processing: Getting your data ready to be fed into your classification algorithms. Which involves cleaning your data for any noise or unnecessary elements.

- **Removing Stopwords**
- **Removing special characters like emojis, hashtags, etc**
- **Convert all the text into Lowercase for generalisation.**
- **Removing punctuations and any other thing you may think does not affect the text's sentiment.**

Step 3

Feature Extraction: Now you have cleaned text data, but the machine doesn't understand English, so you'll have to convert the text into meaningful numerical representation, which can be fed to your machine for predictions.

Creating Embeddings: You can use different techniques to generate word embeddings, such as

- **One hot encoding**
- **Bag of Words**
- **Count Vectorizer**
- **TF-IDF vectorizer**
- **Word2vec**

Step 4

Model Selection: Now we have our input data ready to be fed to a neural network, but which? We'll try out different classification algorithms (this is a binary classification problem), from simple ones to complex ones, and then evaluate which one gives the best results. Models such as:

- **Logistic Regression**
- **Bernoulli Naive Bayes Classifier**
- **SVM (Support Vector Machine)**
- **Random Forests**
- **A simple Neural network with a few dense layers**

Step 5

Evaluation: To evaluate how well you performed(or your model), we will assess them on a set of metrics:

- **Accuracy Score**
- **ROC-AUC**
- **F1 Score**
- **Confusion Matrix**

Libraries for NLP

Some of natural language processing libraries include:

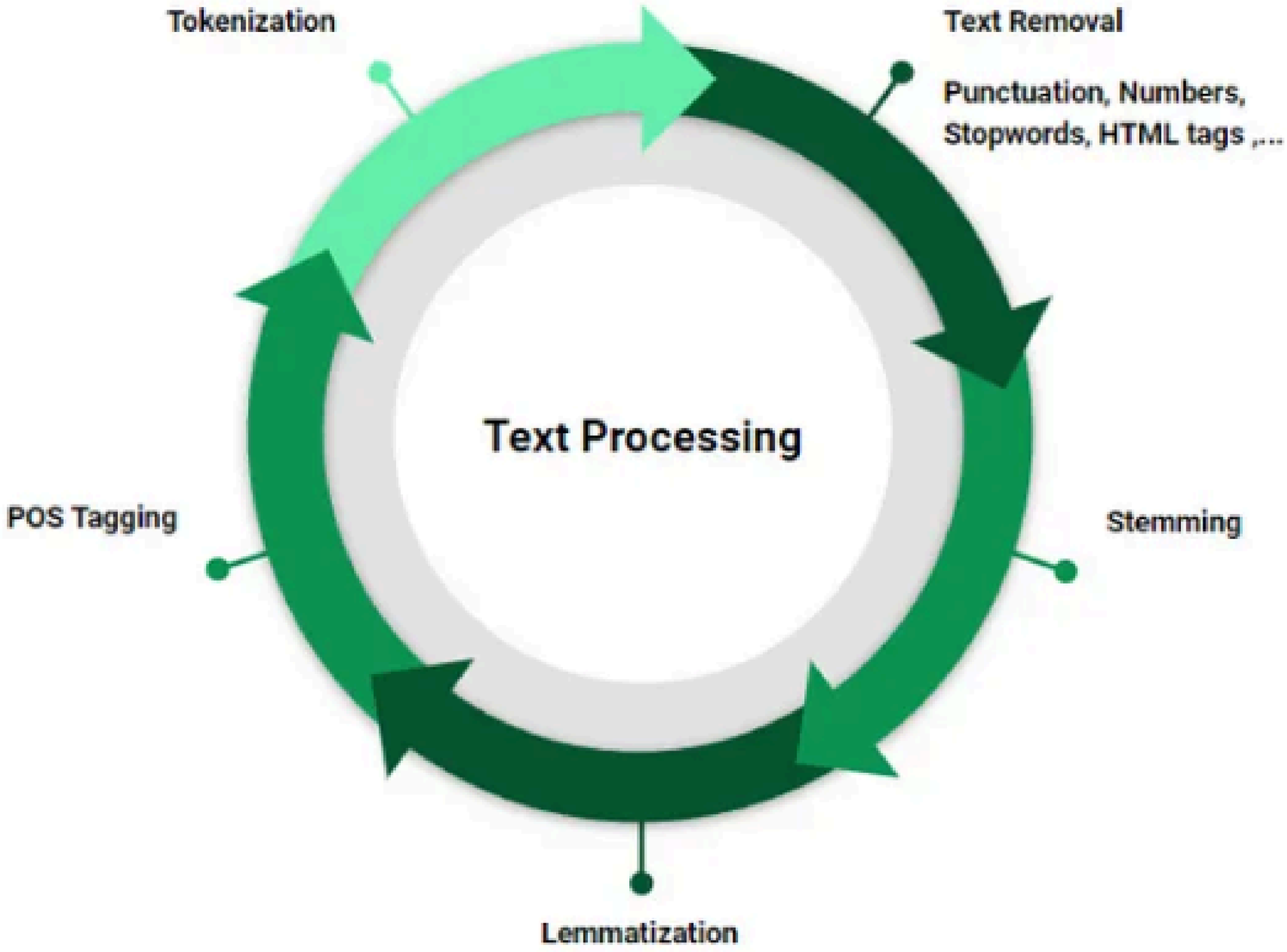
- NLTK (Natural Language Toolkit)
- spaCy
- TextBlob
- Transformers (by Hugging Face)
- Gensim

Text Preprocessing in NLP

Text preprocessing is a crucial step in Natural Language Processing (NLP) that involves cleaning and transforming raw text data into a format suitable for analysis and machine learning models. This process is vital for enhancing the performance and accuracy of NLP tasks.

One key reason for text preprocessing is to remove noise and irrelevant information from the text, such as special characters, punctuation, and stop words. This helps in reducing the dimensionality of the data and improves the efficiency of subsequent analysis. Additionally, text normalization techniques, such as stemming and lemmatization, ensure that words are represented in their base or root form, reducing redundancy and enhancing the consistency of the dataset.

For example, consider the sentence: "The quick brown foxes are jumping over the lazy dogs." After preprocessing, it might become: "quick brown fox jump lazy dog." This simplification facilitates better feature extraction and enables NLP models to focus on the essential linguistic elements.


Tokenization
Text Removal
Punctuation, Numbers,
Stopwords, HTML tags ,...
Text Processing
POS Tagging
Stemming
Lemmatization

Moreover, text preprocessing addresses issues like :

- Lowercase letters.
- Removing HTML tags.
- Removing URLs.
- Removing punctuation.
- Chat Words Treatment.
- Spelling Correction.
- Removing stop words
- Handling Emojies
- Tokenization
- Stemming
- Lemmatization

Handling StopWords

In NLP text preprocessing, removing stop words is crucial to enhance the quality and efficiency of analysis. Stop words are common words like "the," "is," and "and," which appear frequently in text but carry little semantic meaning. By eliminating stop words, we reduce noise in the data, decrease the dimensionality of the dataset, and improve the accuracy of NLP tasks such as sentiment analysis, topic modeling, and text classification. This process streamlines the analysis by focusing on the significant words that carry more meaningful information, leading to better model performance and interpretation of results.

Handling Emojis

Handling emojis in NLP text preprocessing is essential for several reasons. Emojis convey valuable information about sentiment, emotion, and context in text data, especially in informal communication channels like social media. However, they pose challenges for NLP algorithms due to their non-textual nature. Preprocessing involves converting emojis into meaningful representations, such as replacing them with textual descriptions or mapping them to specific sentiment categories. By handling emojis effectively, NLP models can accurately interpret and analyze text data, leading to improved performance in sentiment analysis, emotion detection, and other NLP tasks.

Tokenization

Tokenization is a crucial step in NLP text preprocessing where text is segmented into smaller units, typically words or subwords, known as tokens. This process is essential for several reasons. Firstly, it breaks down the text into manageable units for analysis and processing. Secondly, it standardizes the representation of words, enabling consistency in language modeling tasks. Additionally, tokenization forms the basis for feature extraction and modeling in NLP, facilitating tasks such as sentiment analysis, named entity recognition, and machine translation. Overall, tokenization plays a fundamental role in preparing text data for further analysis and modeling in NLP applications.

We Generally do 2 Type of tokenization 1. Word tokenization 2. Sentence Tokenization

Stemming

Stemming is a text preprocessing technique in NLP used to reduce words to their root or base form, known as a stem, by removing suffixes. It helps in simplifying the vocabulary and reducing word variations, thereby improving the efficiency of downstream NLP tasks like information retrieval and sentiment analysis. By converting words to their common root, stemming increases the overlap between related words, enhancing the generalization ability of models.

However, stemming may sometimes result in the production of non-existent or incorrect words, known as stemming errors, which need to be carefully managed to avoid impacting the accuracy of NLP applications.

Lemmatization

Lemmatization is performed in NLP text preprocessing to reduce words to their base or dictionary form (lemma), enhancing consistency and simplifying analysis. Unlike stemming, which truncates words to their root form without considering meaning, lemmatization ensures that words are transformed to their canonical form, considering their part of speech. This process aids in reducing redundancy, improving text normalization, and enhancing the accuracy of downstream NLP tasks such as sentiment analysis, topic modeling, and information retrieval. Overall, lemmatization contributes to refining text data, facilitating more effective linguistic analysis and machine learning model performance.

Stemming V/s Lemmatization

Stemming and lemmatization both reduce words to a base form, but stemming uses simple rules to chop off endings, creating potentially nonsensical stems (e.g., "running" -> "runn"), making it fast but less accurate, while lemmatization uses context and dictionaries (morphological analysis) to find the actual dictionary word (lemma), like "better" -> "good," making it slower but more accurate for tasks needing meaning. Stemming is for speed (search indexing), lemmatization for accuracy (sentiment analysis, translation).

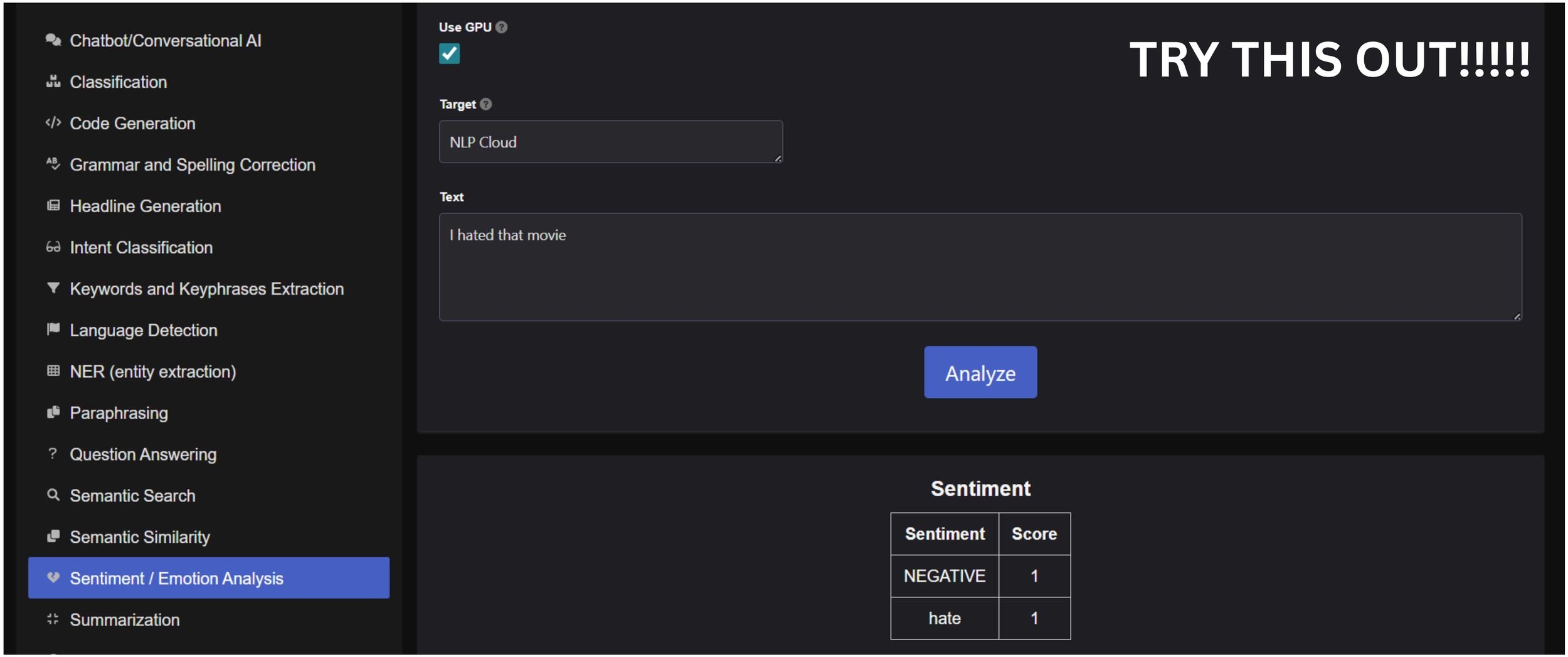
Stemming

- **Method:** Rule-based chopping of suffixes (e.g., "-ing", "-ed", "-s").
- **Output:** A "stem" that might not be a real word (e.g., "flies" -> "fli").
- **Speed:** Very fast.
- **Use Case:** Large datasets, quick search indexing, reducing dimensionality where exact word meaning isn't critical.
- **Example:** "Caring" -> "Car," "Studies" -> "Studi".

Lemmatization

- **Method:** Uses dictionaries (like WordNet) and part-of-speech (POS) tagging for context.
- **Output:** A valid dictionary word (lemma).
- **Speed:** Slower, more computationally intensive.
- **Use Case:** Tasks requiring grammatical accuracy, sentiment analysis, topic modeling, machine translation.
- **Example:** "Caring" -> "Care," "Better" -> "Good," "Meeting" (verb) -> "Meet".

API



https://nlpcloud.com/home/playground/sentiment-analysis

Word2Vec Implementation

About Dataset

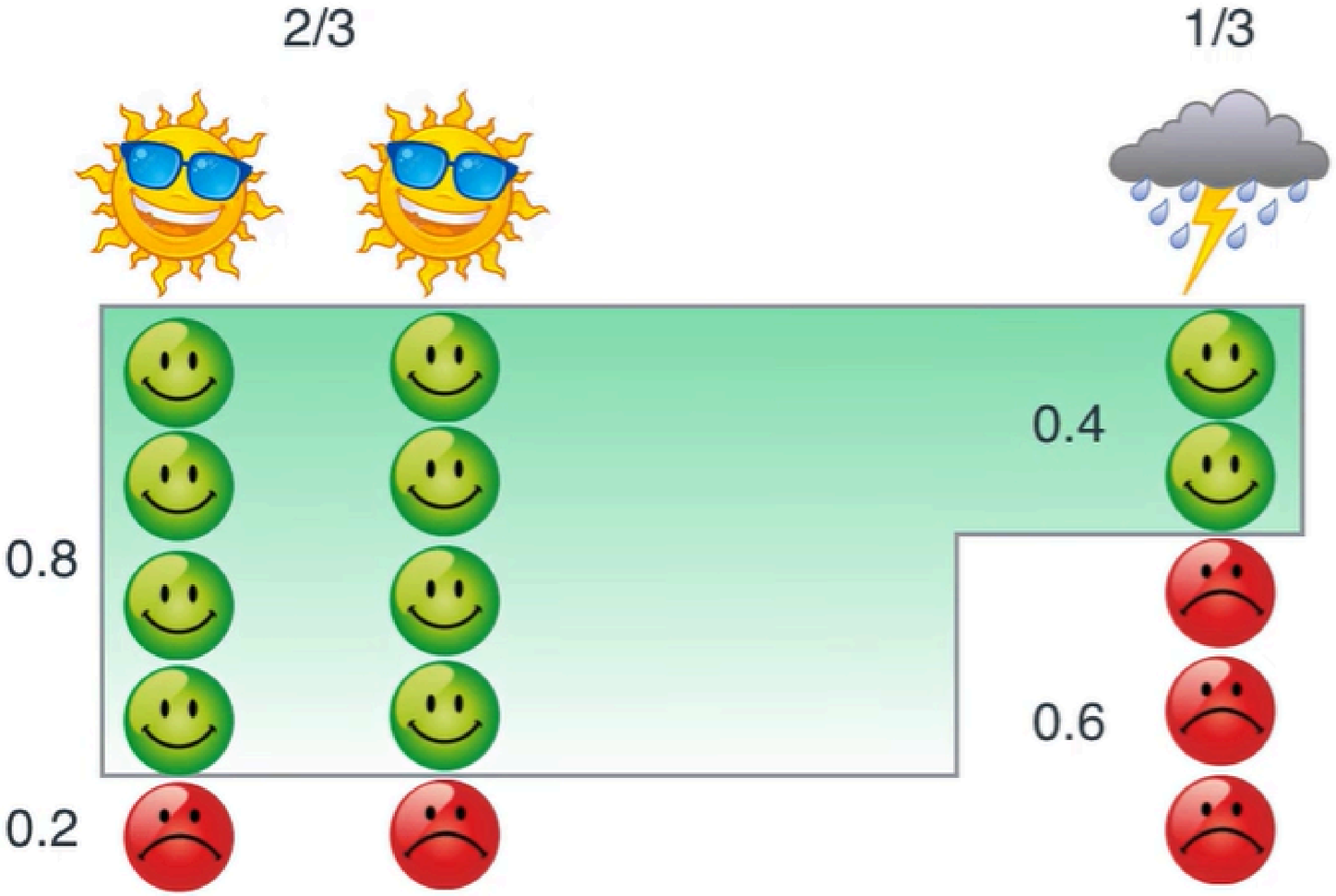
This dataset contains the full text of all seven books from the Harry Potter series by J.K. Rowling. Each book is provided as a separate text file. This dataset is intended for educational and research purposes, allowing for text analysis, natural language processing (NLP), sentiment analysis, and other data science projects.

Books Included

Harry Potter and the Philosopher's Stone (Sorcerer's Stone in the US)
Harry Potter and the Chamber of Secrets
Harry Potter and the Prisoner of Azkaban
Harry Potter and the Goblet of Fire
Harry Potter and the Order of the Phoenix
Harry Potter and the Half-Blood Prince
Harry Potter and the Deathly Hallows

Download the dataset from here:
https://www.kaggle.com/datasets/rupanshukapoor/harry-potter-books

Hidden Markov Model

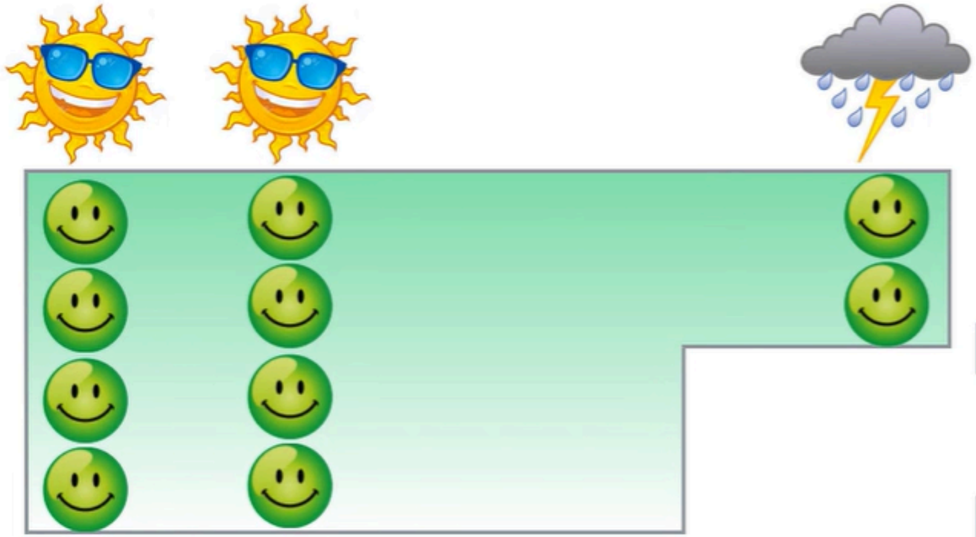


Bayes Theorem

Visualization

Fig.1
When we are checking the conditional probabilities

Conditional Probabilities Continued



Bayes Theorem

If 🙂

$$P(☀ \mid 🙂) = \frac{8}{10}$$

$$P(⛈ \mid 🙂) = \frac{2}{10}$$

Continued

Bayes Theorem

If 🔴

$$P(☀️ | 🔴) = \frac{2}{5}$$

$$P(🌩️ | 🔴) = \frac{3}{5}$$

References:

A friendly introduction to Bayes Theorem and Hidden Markov Models
https://youtu.be/kqSzLo9fenk?si=mkx5oa16IXD9nEYr

Introduction to NLP
https://youtu.be/zlUpTlaxAKI?si=UvGRpwwLV41UAQ1R

https://www.kaggle.com/code/abdmental01/text-preprocessing-nlp-steps-to-process-text

https://www.geeksforgeeks.org/nlp/natural-language-processing-nlp-tutorial/

https://www.youtube.com/watch?v=fLvJ8VdHLA0