PyCon Ireland 2023
12.11.2023

Introduction to NLP & Text Classification

Women in AI Ireland

Nabanita Roy & Liliya Akhtyamova

The WAI TEAM today!

Nabanita Roy
Women in AI Ireland
Senior Data Scientist at EY

https://www.linkedin.com/in/nabanita-roy/

Liliya Akhtyamova
Women in AI Ireland
Senior Data Scientist & ML Engineer at iCIMS

https://www.linkedin.com/in/datawhizette/

ABOUT US

WOMEN IN AI IS A COMMUNITY WORKING TOWARDS SHAPING INCLUSIVE AI FOR OUR COMMON FUTURE.

WAI Global

- Launched in 2017
- Impact-driven
- 9 Sustainable Development Goals in focus
- 10K members (volunteers)
- 35K+ Social Media followers
- ~150 countries of coverage
- 40+ Ambassadors

Member profile

- 96% of members are women
- junior to C-level (age group 25-40 years old)
- who are working in or studying STEM, Business or Entrepreneurship
- 42% Masters Degree, 30% Bachelor Degree, 14% PHD
- 3.5k different fields of expertise among our community members.
- The most popular fields are: AI, Computer Science, Data Science, IT and Law

Agenda

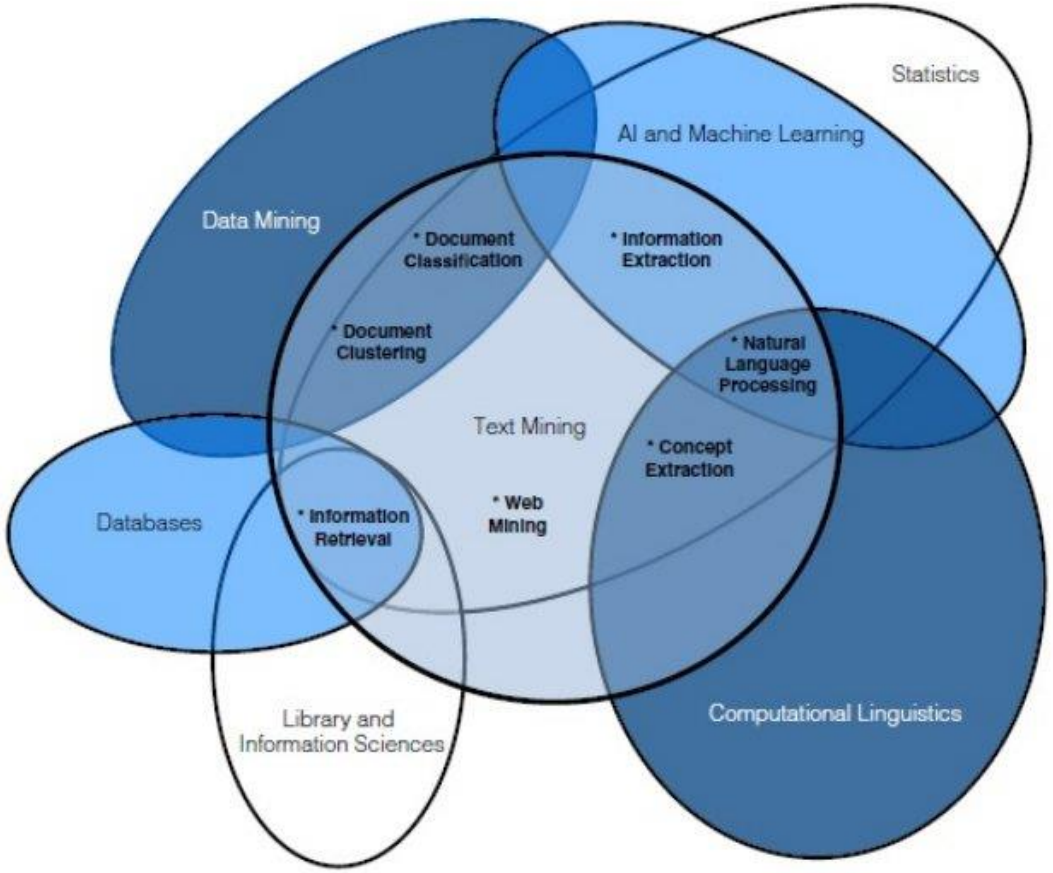
QR – Code to GitHub

- What is Natural Language Processing (NLP)
- NLP Tasks
- Text Vectorization Techniques
- Text Preprocessing
- Text Classification Methods
- Model Evaluation for Text Classification
- End-to-End Example
- Practice

Natural Language Processing

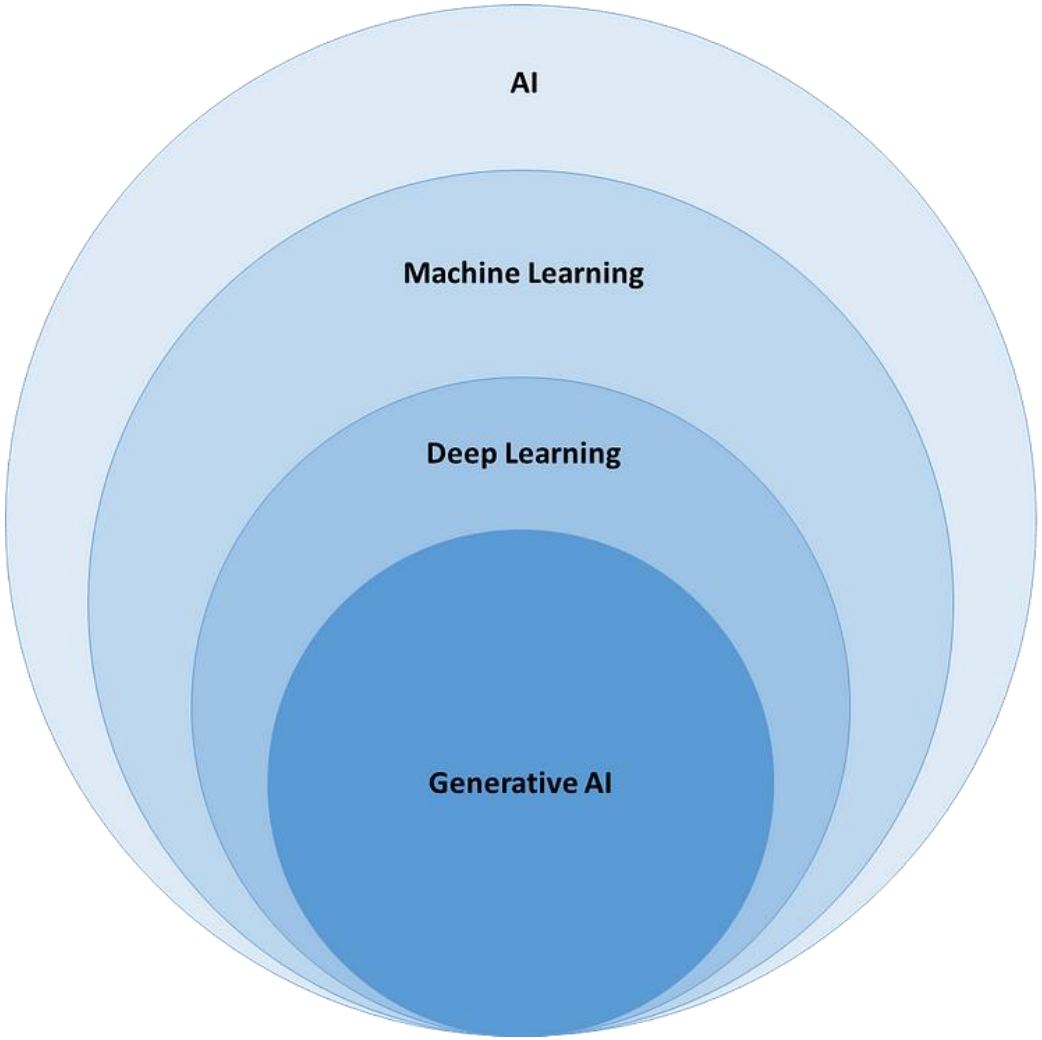
Natural Language Processing (NLP) - a section of Data Science devoted to how computers analyze natural (human) language. NLP allows you to apply machine learning algorithms to text and speech.

Natural Language Processing



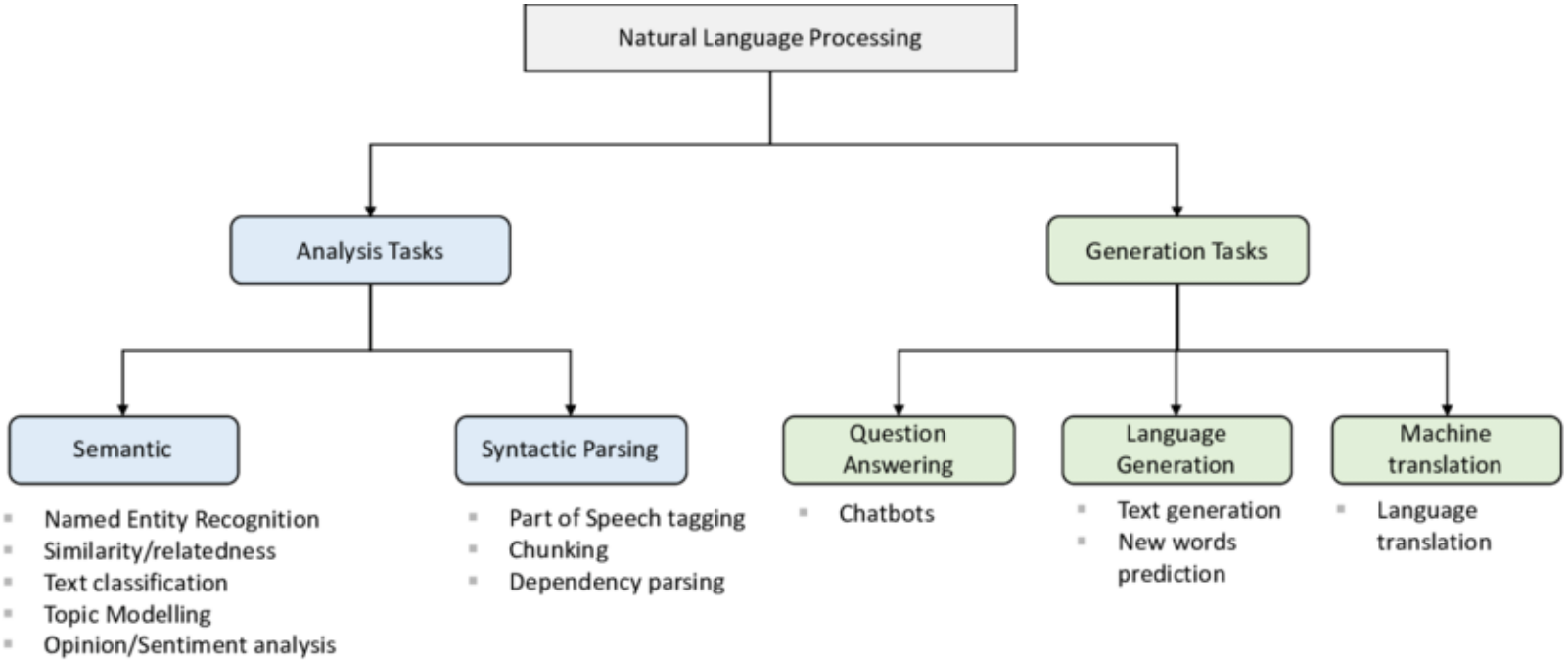
Where does NLP fit here?

Menti: 8215 9464

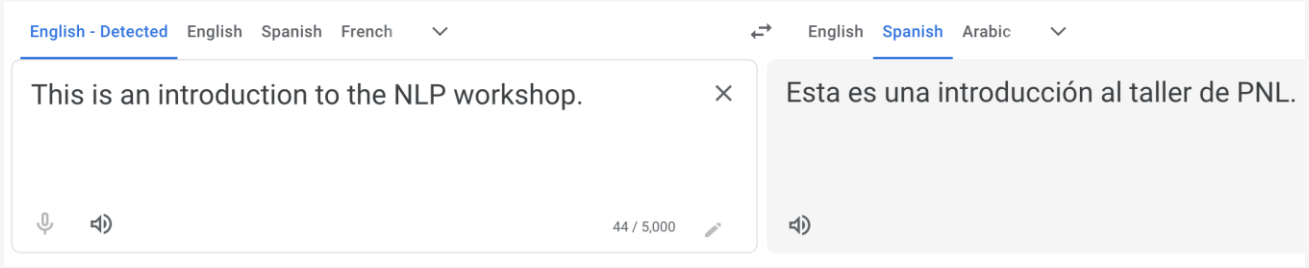


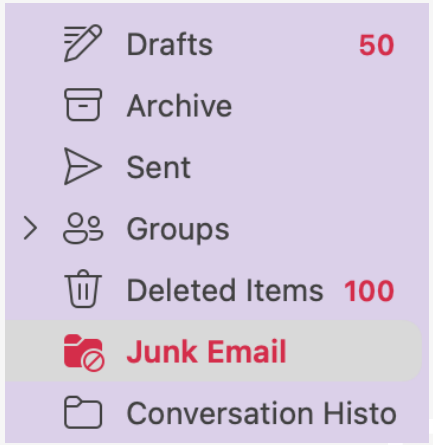
NLP Tasks Overview

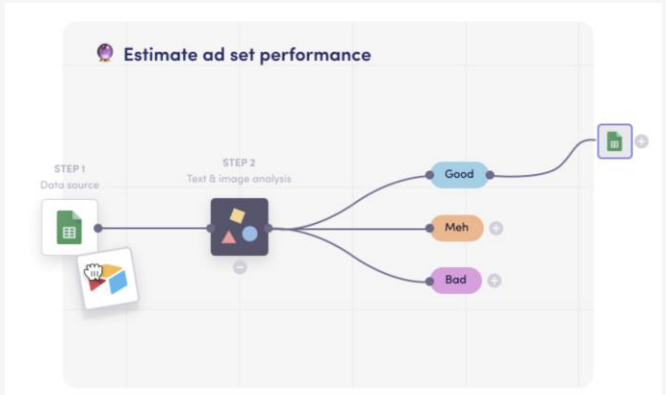
Menti: 8215 9464

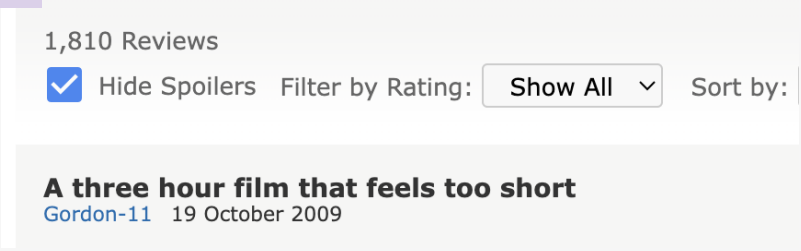


Source: https://www.researchgate.net/figure/NLP-Tasks-Categorized-in-Two-Broader-Categories-Analysis-Tasks-light-blue-and_fig4_343323519









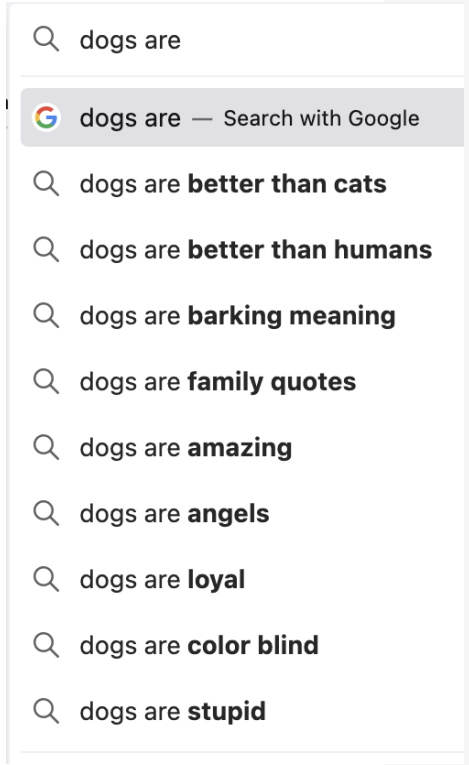
NLP Tasks

- Machine Translation

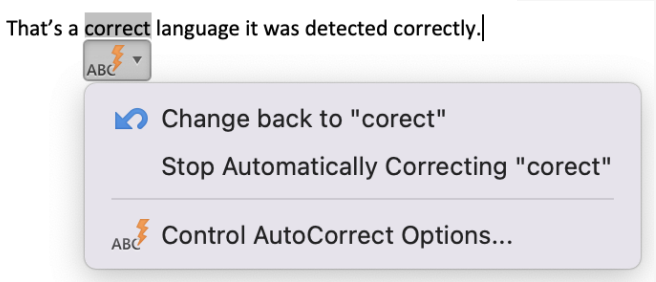
- Classification
  - Spam filtration
  - Sentiment analysis
  - Spoiler detection

NLP Tasks

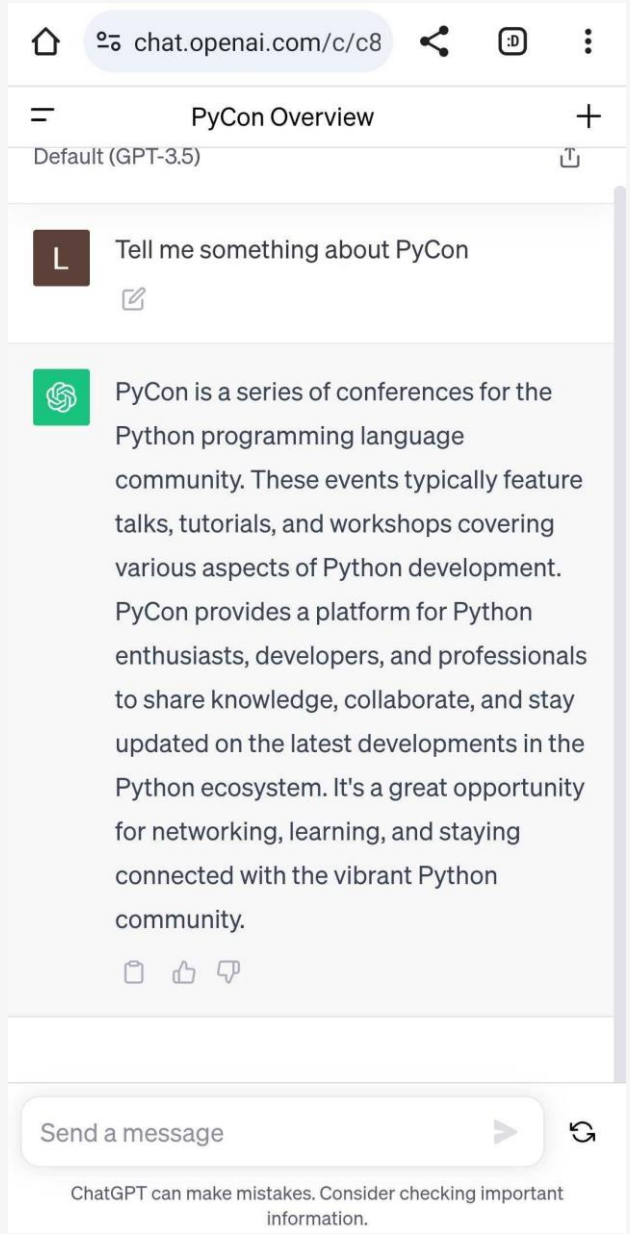
- Autocompletion



- Autocorrection

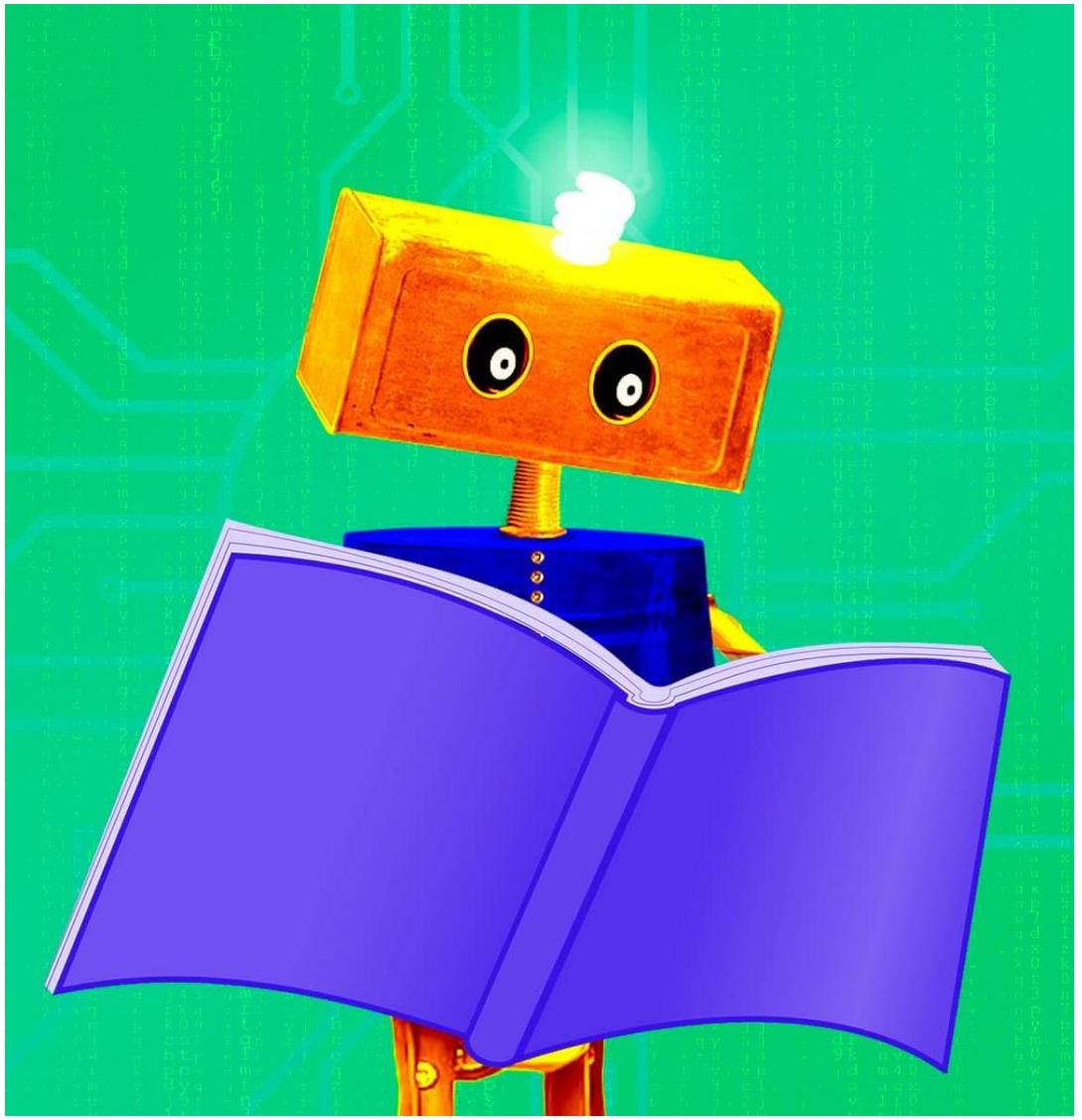


- Chatbots - GenAI (Wohoo!)



How do computers understand words?

- Through numbers (and then bits)
- Need to encode words (or parts of words, sentences, whole documents) into numbers. That process is called text vectorization, and numerical representations of texts are called embeddings
- Most popular types of embeddings are word embeddings



Funny video – how automated systems can struggle to understand human language:
Scottish Elevator - Voice Recognition - ELEVEN !

Text Vectorization Techniques

- Bag Of Words (Count Vectorizer)
- Term Frequency and Inverse Document Frequency (TF-IDF)
- Word2Vec
- Transformers

Text Vectorization Techniques

- Bag Of Words (Count Vectorizer)
- Term Frequency and Inverse Document Frequency (TF-IDF)
- Word2Vec
- Transformers

Bag of Words

- Bag of Words (BoW) is a model that is represented as an unordered set of words included in the processed text.

- Simple, but

"I love dogs, hate cats" == "I love cats, hate dogs" for BoW

- Still, may suffice for the global context: movie sentiment analysis, restaurant feedback, etc as the details of feedback message are less important

TF-IDF (Term Frequency – Inverse Document Frequency)

TF-IDF (Term Frequency - Inverse Document Frequency) is an algorithm that uses the frequency of words to determine how relevant those words are to a given document.

The weight of a word is proportional to the frequency of occurrence of this word in the document and inversely proportional to the frequency of occurrence of the word in all documents in the collection.

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

Preprocessing

Problems:

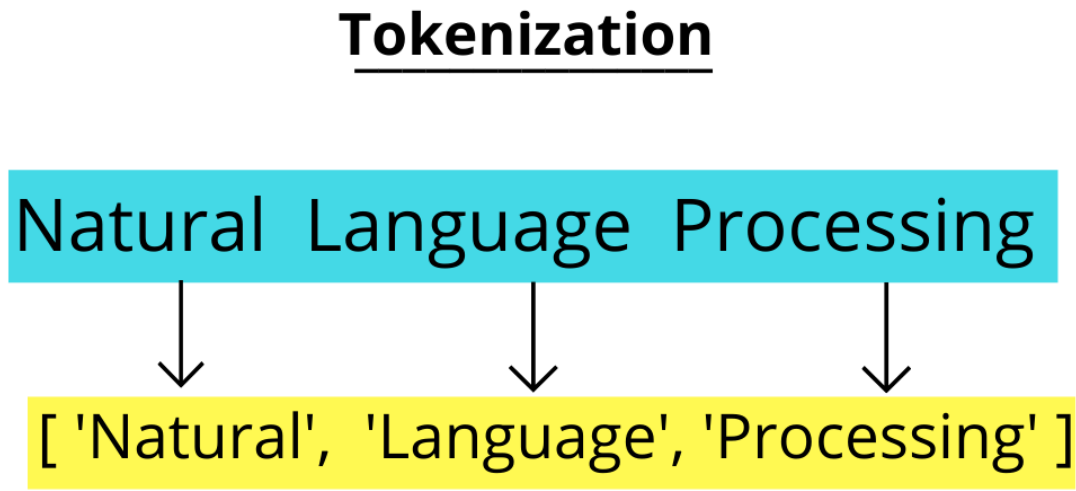
- long vectors (word embeddings) => expensive computation
- Same words with little difference in writing are considered different, i.e. “cup” vs “cups”, “finalize” vs “finalise”

Types of text preprocessing

- Tokenization,
- Noise removal (stop words, lowercasing, punctuation, etc)
- Stemming,
- Lemmatization.

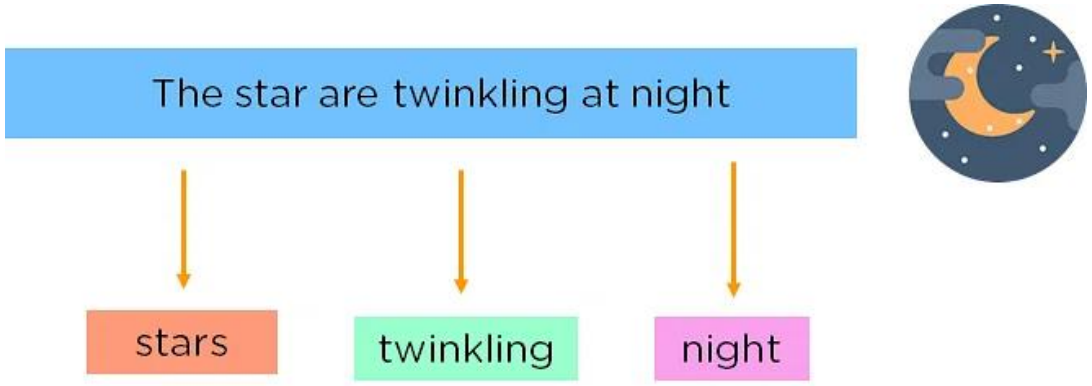
Most important – tokenization.

It’s the process of breaking a stream of textual data into words, terms, sentences, symbols, or some other meaningful elements called tokens



Stop words

- Stop words are words that are thrown out of the text during text processing. When we apply machine learning to texts, such words can add a lot of noise, so it is necessary to get rid of irrelevant words.
- Stop words are usually articles, interjections, conjunctions, etc., which do not carry a semantic meaning.
- At the same time, one must understand that there is no universal list of stop words, everything depends on the specific case.



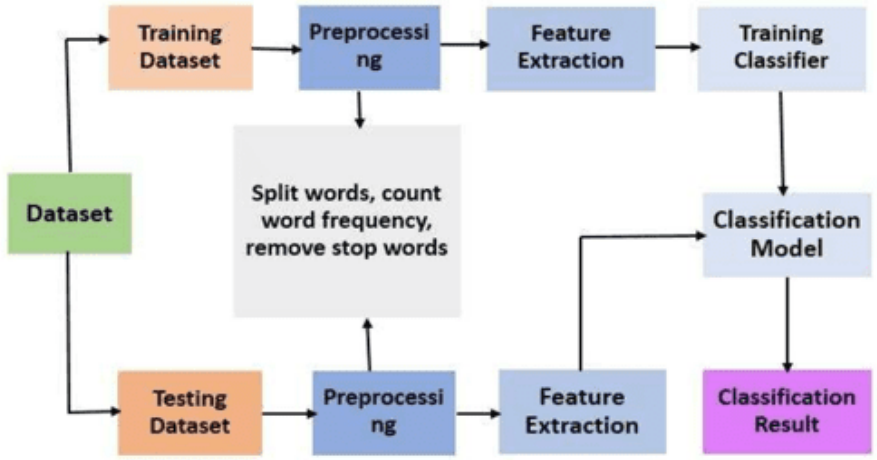
Word normalization

- Stemming is the process of finding the stem of a word for a given source word. The stem of the word does not necessarily coincide with the morphological root of the word and does not have to be an existing word in the language. Stemming is a crude heuristic process that cuts off "excess" from the root of words, often resulting in the loss of derivational suffixes.

- Lemmatization brings all occurring word forms to one, normal dictionary form. Lemmatization uses vocabulary and morphological analysis to eventually bring the word to its canonical form, the lemma.

General Text Classification steps

- Getting the dataset
- Train/(dev)/test split
- Preprocessing
- Feature extraction
- Training
- Evaluation



Model Evaluation

		Predicted: Negative	Predicted: Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Recall

Precision

$$\textbf{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

$$\textbf{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision + Recall}}$$

Walkthrough time…

Let’s get practical!

QR – Code to GitHub

Thank you!