Preprocessing for NLP

Dr. Manuel Pita & Zuil Pirola

UNIVERSIDADE
LUSÓFONA

Why preprocessing?

Preprocessing in NLP is crucial for **cleaning** and **standardising** text data. It removes noise (e.g., typos, stop words) and applies techniques like **tokenisation**, **normalisation**, and **stemming**. This improves model accuracy, reduces dimensionality, and enhances computational efficiency by focusing on **relevant patterns** in the data.

Is a Universal Solution?

Preprocessing in NLP must be **tailored to specific tasks** and data. While standard practices exist (e.g., tokenisation, stop word removal), not all are universally applicable. The process should be **adjusted based on the context**, ensuring relevant information is preserved while reducing noise.

Regex and others can help us

Regex is crucial. It efficiently removes noise like special characters. Other tools, like **NLTK** and **SpaCy**, complement regex by providing tokenisation, stemming, and lemmatisation, enhancing preprocessing for better results.

Preprocessing steps

- **Lowercasing and Text Normalisation**: Standardise formats (e.g., date formats, case normalisation).
- **Tokenisation**: Split text into tokens.
- **Stop Word Removal**: Remove common words that don't add much meaning (e.g., "and", "the").
- **Noise Removal**: Remove punctuation, special characters (#), digits and URLs that are not needed.
- **Stemming/Lemmatisation**: Reduce words to their root form (e.g., "running" to "run").

Regex and others can help us

Regex is crucial. It efficiently removes noise like special characters. Other tools, like **NLTK** and **SpaCy**, complement regex by providing tokenisation, stemming, and lemmatisation, enhancing preprocessing for better results.

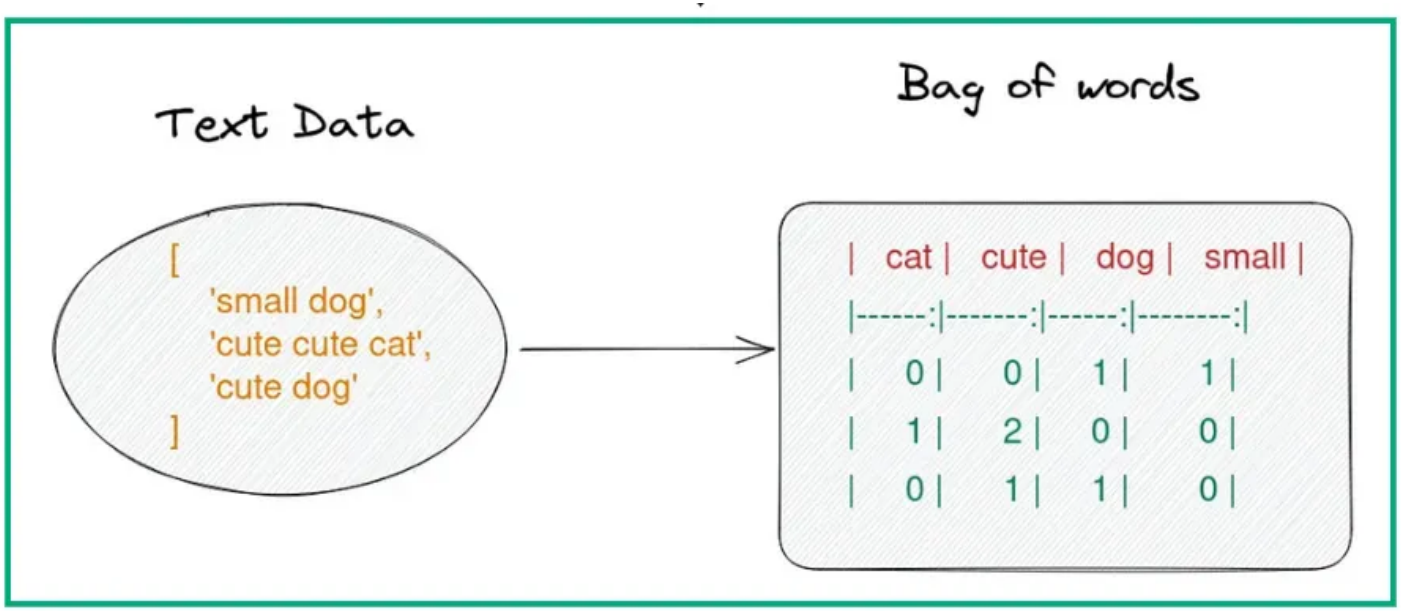
Lets try

https://github.com/zuilpirola/NLP

Preprocessing and BOW

Preprocessing is crucial for Bag of Words (BoW) as it cleans and **standardises text data**, ensuring accurate word **frequency capture**. It removes irrelevant, allowing BoW to focus on **meaningful words**. Techniques like tokenisation and stemming enhance the representation of text, **improving model performance**.

BOW



*https://ayselaydin.medium.com/4-bag-of-words-model-in-nlp-434cb38cdd1b

BOW applications

Text Classification: Identifying the category of a text document, such as spam detection in emails.

Sentiment Analysis: Determining the sentiment expressed in a piece of text, such as positive or negative reviews.

Information Retrieval: Searching and retrieving documents based on keyword matches.

BOW limitations

Loss of Context: By ignoring word order, the model may lose important contextual information that could change the meaning of phrases.

High Dimensionality: In cases with large vocabularies, the resulting vectors can be very high-dimensional, leading to increased computational complexity.

Next steps

Next, we can explore more advanced text representation techniques, such as TF-IDF. Additionally, we will discuss how to apply these concepts to practical tasks like sentiment analysis and text classification.