Logo, company name

Description automatically generated

**CSE523 Machine Learning**

**Prof. Mehul Raval**

**Weekly report**

**Group number: 17**

**Group name: The Mandelbrot set**

| **Name** | **Enrolment Number** |
| --- | --- |
| Aastha Gaudani | AU2040032 |
| Khushi Patel | AU2040068 |
| Devyash Shah | AU2040152 |
| Simran Khoja | AU1910606 |

**REPORT**

Dataset: <https://www.kaggle.com/datasets/pariza/bbc-news-summary?resource=download>

Searched for some libraries and then performed tokenization using functions from the libraries nltk, os and spacy. But we intend to perform it with the minimalistic use of libraries. For tokenization on the BBC data we first gave input as paragraphs and separated the sentences and words later on. Thenonwards removed the less relevant words in order to reduce the data.

This week we planned on doing the feature extraction part. What is feature extraction? Feature extraction is a crucial step in building a machine-learning model. It involves identifying and selecting the most relevant information (features) from the input data that will be used to train the model. The goal of feature extraction is to transform the raw input data into a set of meaningful and informative features that can effectively represent the underlying patterns in the data.

The below code uses the Natural Language Toolkit (nltk) library to perform text tokenization on a given paragraph. It first splits the paragraph into sentences using the sent\_tokenize() function and stores the resulting sentences in the "sentences" list. The code then calculates the number of sentences using the len() function and stores it in "numr".

Next, the code tokenizes the paragraph into individual words using the word\_tokenize() function and stores the words in the "words" list. The number of words is calculated using the len() function and stored in "numb".

Finally, the code creates a filtered text by removing stop words (common words such as "the", "is", "an", etc.) from the "words" list using a list comprehension and the word\_tokenize() function. The filtered text is stored in the "filtered\_text" list and the length of this list is printed.

And in the last image,

The below code uses the spaCy library to perform text analysis on a given paragraph. It starts by loading the "en\_core\_web\_sm" model, which is a small English language model for basic text analysis tasks. The loaded model is stored in the "nlp" variable.

Next, the code applies the "nlp" model to the given paragraph and stores the resulting document in the "doc" variable. This document is now a spaCy-processed representation of the original paragraph.

The code then uses a for loop to iterate over each token (i.e. word) in the document. If the current token is a stop word (a commonly occurring word such as "the", "is", "an", etc.), the loop continues to the next iteration.

For all other tokens, the code prints the token text, part-of-speech (POS) tag, and syntactic dependency label. The POS tag and dependency label provide additional information about the role of each token in the sentence and its relationship to other tokens.





