Cairo University
Faculty of Computers and Information
Department of Computer Sciences

Text Classification

Supervised by

Dr. Hesham Hassan

TA. Dalia Maher

Implemented by

ID	Name	E-mail	Group
20170022	Ahmed Sayed Mansour	ahmed.mans20719@gmail.com	CS-IS-1
20170021	Ahmed Sayed Ibrahim	ahmed111522@gmail.com	CS-DS-1
20170136	Atef Magdy Mitwally	atefmagdy12@gmail.com	CS-IS-1
20170053	Ashraf Samir Ali	ashrafsamer423@gmail.com	CS-DS-1
20170002	Ibrahim Ramadan Abdou	ibrahemramadan130@gmail.com	CS-IS-1

Graduation Project
Academic Year 2020-2021
Mid-year Short Documentation

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abstract

With the explosion of information fueled by the growth of the numbers of text categories. It is no longer feasible for a human observer to understand all the data coming in or even classify it into categories. With this growth of information and simultaneous growth of available computing power automatic classification of data, particularly textual data, gains increasingly high importance. This project provides a review of the text classification process, phases of that process and methods being used at each phase. Examples from research papers classification are provided throughout the text. Principles of operation of four main text classification algorithms “Naïve Bayesian”, “Neural networks”, “Support Vector Machines” and “Random forest”. This project will look through the state of the art in all these phases, take note of methods and algorithms used and of different ways that researchers are trying to reduce computational complexity and improve the precision of text classification process as well as how the text classification is used in practice. The paper is written in a way to avoid extensive use of mathematical formulae in order to be more suited for readers with little or no background in theoretical mathematics.

Introduction

1.1. What is classification?

Classification is a supervised machine learning technique, Classification is the process of categorizing a given set of data into classes, and it can be performed on both structured and unstructured data. The process starts with predicting the class of given data points. The classes are often referred to as target, label or categories.

1.2. What is text classification?

Text classification is the process of categorizing a text into organized groups. By using Natural Language Processing (NLP), text classifiers can automatically analyze text and then assign a set of pre-defined tags or categories based on its content. As shown in figure 1.

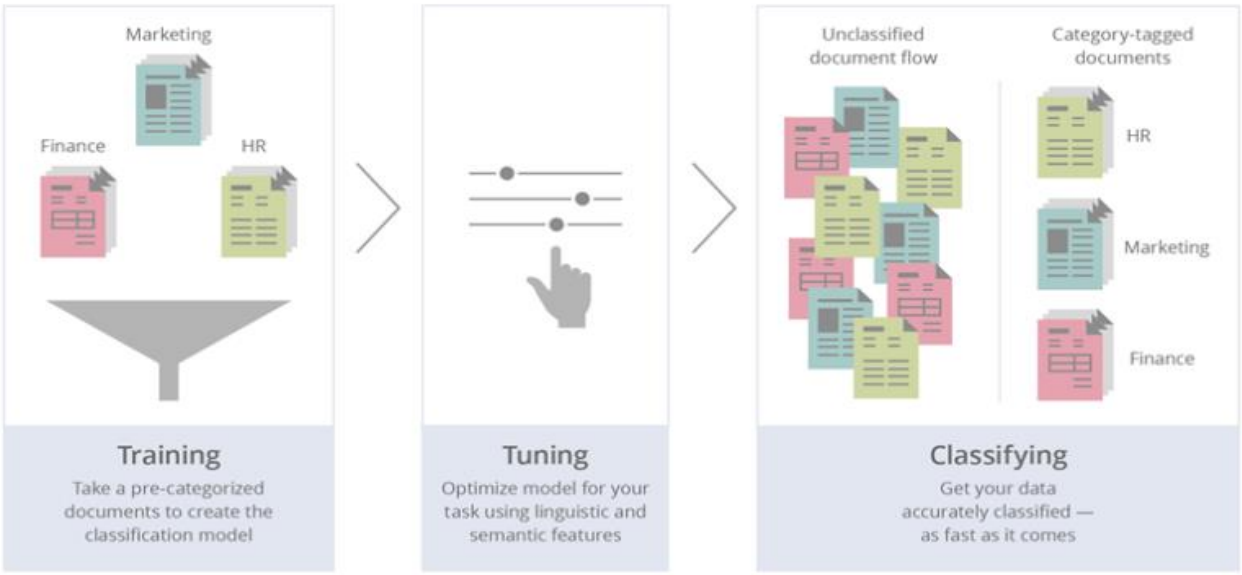

Figure 1 – Text Classification Processing

1.3. Variants of text classification

- **Binary categorization:** only two categories
  - Retrieval: {relevant-doc, non-relevant-doc}
  - Span filtering: {spam, not-spam}
  - Opinion: {positive, negative}
- **K-category categorization:** more than two categories
  - Topic categorization: {sports, science, travel, business}
  - Email routing: {folder1, folder2, folder3 …}
- **Hierarchical categorization:** categories from hierarchy.
- **Joint categorization:** multiple related categorization tasks done in joint matter.

1.4. Why is text classification important?

It's estimated that around 80% of all information is unstructured, with text being one of the most common types of unstructured data. Because of the messy nature of text, analyzing, understanding, organizing, and sorting through text data is hard and time-consuming, so most companies fail to use it to its full potential. This is where text classification with machine learning comes in. Using text classifiers, companies can automatically structure all manner of relevant text, from emails, legal documents, social media, Chabot's, surveys, and more in a fast and cost-effective way. This allows companies to save time analyzing text data, automate business processes, and make data-driven business decisions. [1]

1.5. The uses of text classification

- **Language detection**: the procedure of detecting the language of a given text (e.g., know if an incoming support ticket is written in English or Spanish for automatically routing tickets to the appropriate team).
- **Topic Labeling classification**: the task of identifying the theme or topic of a piece of text (e.g., know if a product review is about Ease of Use, Customer Support, or Pricing when analyzing customer feedback).

- **Sentiment Analysis**: the process of understanding if a given text is talking positively or negatively about a given subject (e.g., for brand monitoring purposes).
- **Intent classification**: is another great use case for text classification that analyzes text to understand the reason behind feedback.[2][3]

1.6. Objectives

Our objectives are to try different techniques in text classification to know which one is better and has the best efficiency on research papers datasets. We divided our objectives to these points:

- Building single label text classification models.
- Building multi label text classification models.
- Trying different techniques for classification models and differentiate between the different techniques based on (Accuracy, Performance, Memory handling).
- Building web application using react libraries as a front-end and Flask & Node JS as a back-end, to let people use our trained models.
- Posting our research on the web app to be public for developers and who is interested in text classification techniques.

1.7. Layout

Chapter 2: Background (more details about text classification, algorithms and techniques)
Chapter 3: Related work (analysis about the current text classification projects)
Chapter 4: Proposed text classification (project workflow, architecture and algorithms)
Chapter 5: Project specifications (methodology, requirements, use-cases, diagrams)

- Conclusion
- References

2. Background

In this chapter, we first review the methodologies for text classification. Concretely, we illustrate the **Preprocessing-data**, **NLP** of text classification, **algorithms** and **technologies**. Preprocessing-data includes text representation. For text representation, we review how to extract features from text, how to clean text and in NLP we review most known techniques for Text-NLP**.** Lastly, we review some algorithms and technologies used for classification implementation.

2.1. Preprocessing Data

In natural language processing, text preprocessing is the practice of cleaning and preparing text data.

- **Noise Removal:** noise removal is a text preprocessing task devoted to stripping text of formatting.
- **Tokenization:** tokenization is the text preprocessing task of breaking up text into smaller components of text (known as tokens).
- **Text Normalization:** normalization encompasses many text preprocessing tasks including stemming, lemmatization, upper or lowercasing, and stop-words removal.
- **Stemming:** stemming is the text preprocessing normalization task concerned with bluntly removing word affixes (prefixes and suffixes).
- **Lemmatization:** lemmatization is the text preprocessing normalization task concerned with bringing words down to their root forms.
- **Stop-word Removal:** stop-word removal is the process of removing words from a string that don't provide any information about the tone of a statement.[4]

2.2. Nature Language Processing

Natural Language Processing (**NLP**) is a field of Artificial Intelligence (AI) that makes human language intelligible to machines. NLP combines the power of linguistics and computer science to study the rules and structure of language, and create intelligent systems (run on machine learning and NLP algorithms) capable of understanding, analyzing, and extracting meaning from text and speech. There are many approaches to NLP text classification, which fall into three types of systems:

- **Rule-based systems:** In the rule-based approach, texts are separated into an organized group using a set of handicraft linguistic rules. Those handicraft linguistic rules contain users to define a list of words that are characterized by groups. For example, words like Donald Trump and Boris Johnson would be categorized into politics. People like LeBron James and Ronaldo would be categorized into sports**.**
- **Machine learning-based systems:** Machine-based classifier learns to make a classification based on past observation from the data sets. User data is pre labeled as train and test data. It collects the classification strategy from the previous inputs and learns continuously. Machine-based classifier usage a **bag of a word** for feature extension. It's preferred to use **tf–idf algorithm** to normalize the number of repeated words. As shown in figure 2.

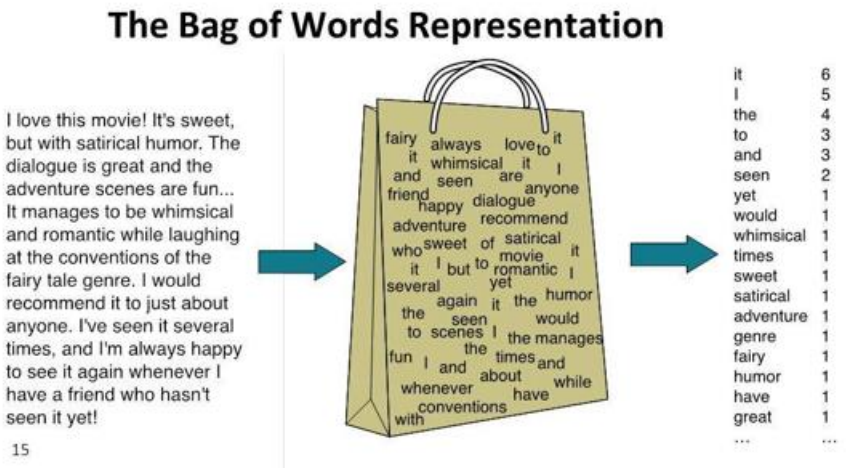

Figure 2 - Feature Selection (Bag of Words Representation)

- **Hybrid systems:** Hybrid approach usage combines a rule-based and machine Based approach. Hybrid based approach usage of the rule-based system to create a tag and use machine learning to train the system and create a rule. Then the machine-based rule list is compared with the rule-based rule list. If something does not match on the tags, humans improve the list manually. It is the best method to implement text classification.[5][6]

2.3. Algorithms

We separated our problem into two categories:

- **Single label text classification:** Is the task of classifying the elements of a set into two groups on the basis of a classification rule. Some of the most popular machine learning algorithms for creating text classification models include the **Naive Bayes** family of algorithms, **support vector machines (SVM),** and **deep learning**.

- **Multi label text classification:** is a generalization of multiclass classification, which is the single-label problem of categorizing instances into precisely one of more than two classes; in the multi-label problem there is no constraint on how many of the classes the instance can be assigned to. Some of the most popular techniques **One Vs Rest** to use with any of the mentioned algorithms above**.** [7]

3. Related work

For our research we will discuss some researches that talking about the idea and showing the differences.

3.1. Resources

Resource 1: [8]

Problem: Binary classification on **The 20 Newsgroups data set.**

Solution: Using scikit-learn and NLTK to load the data and for the NLP process using sklearn.feature_extraction library to apply Count Vectorization and TF-IDF Transformer to the data. Building the model using Naive Bayes algorithm with accuracy of 69% and using SVM with accuracy of 68%.

Advantages:

- Using different algorithms from the sklearn library.
- Explaining all steps and reporting information on each experiment.

Disadvantages:

- The data set is small and only covers a little number of classes.
- Not removing all unimportant data like headers and footers.
- Not using the new trends of text classification like Neural Networks algorithms.

Resource 2: [9]

Problem: Discussing best pre trained models for text Classification

Solution: Using pre trained models like (XLNet, ERNIE, T5…) on different datasets.

Advantages:

- The datasets meet industry-accepted standards.
- The pre trained models have already been vetted on the quality aspect.
- Reporting a summary on each model using different datasets.

Disadvantages: Using the most new trends of ML and deep learning which needs a lot of studying to understand.

3.2. The main differences between related works and our Project

- All related work working on small datasets while we using the Arxiv data (3 GB).
- Our training on many number of labels to cover a wider area of subjects.
- Letting users to try our models on our website.
- Our training for Multilayer classification for better performance not only single label on the related work.

4. Proposed text classification

The steps for our model and how to prepare the data.

4.1. Workflow

4.1.1. Preprocessing data

First phase is cleaning data from any unimportant words and characters in many steps as shown in figure 3.

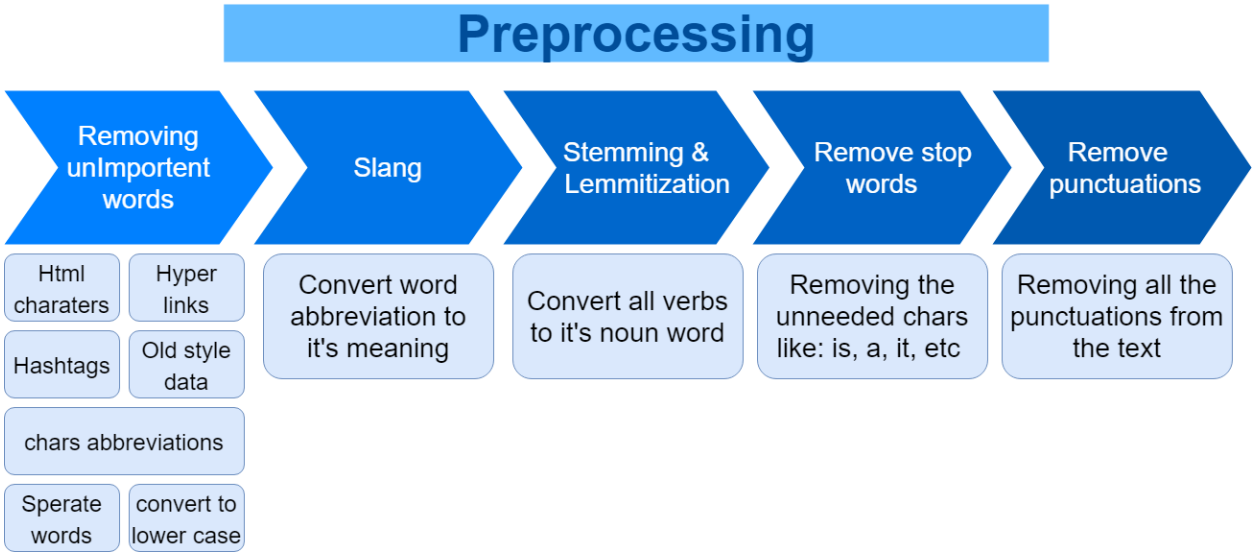

Figure 3 - Preprocessing data

4.1.2. Natural Language Processing (NLP)

Second phase is to convert textual data into numerical data to be used later.

4.1.2.1. Text Vectorization

This is the first step to proceed NLP by selecting all words from files by "Bag of words technique" and give each file a vector of numbers as shown in Fig 4.

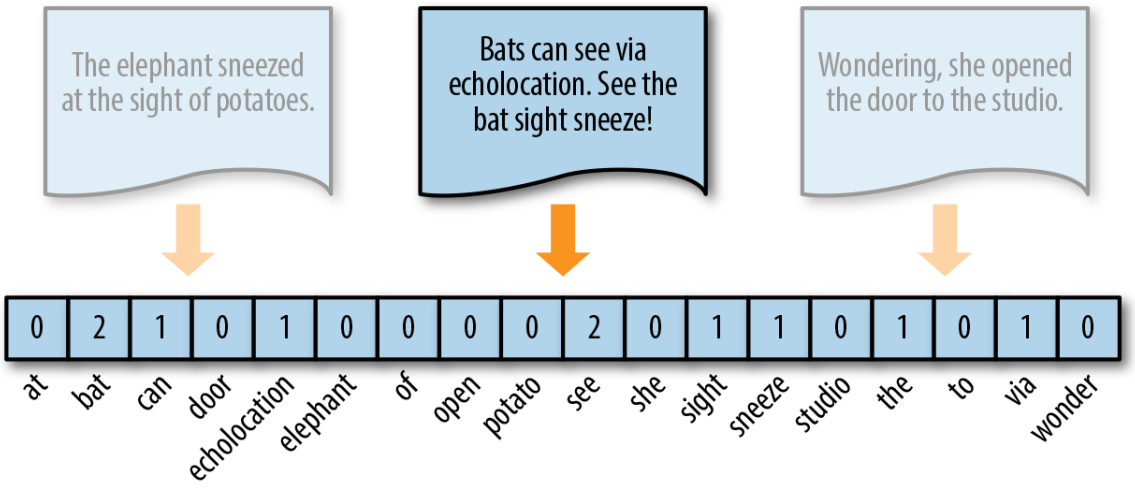

Figure 4 - Text vectorization

4.1.2.2. TF-IDF Normalization

This is the second step of NLP by normalizing previous mentioned vector to improve performance of learning phase as shown in table 1.

Table 1 - TF-IDF Normalization

Word	TF		IDF	TF*IDF	
	A	B		A	B
The	1/7	1/7	log(2/2) = 0	0	0
Car	1/7	0	log(2/1) = 0.3	0.043	0
Truck	0	1/7	log(2/1) = 0.3	0	0.043
Is	1/7	1/7	log(2/2) = 0	0	0
Driven	1/7	1/7	log(2/2) = 0	0	0
On	1/7	1/7	log(2/2) = 0	0	0
The	1/7	1/7	log(2/2) = 0	0	0
Road	1/7	0	log(2/1) = 0.3	0.043	0
Highway	0	1/7	log(2/1) = 0.3	0	0.043

4.1.3. Training and Prediction

Last phase is to:

- Train our model by using extracted vectors from last 2 phases with their label to generate classifier model.
- Use generated classifier model to predict new untrained data.

As shown in figure 5.

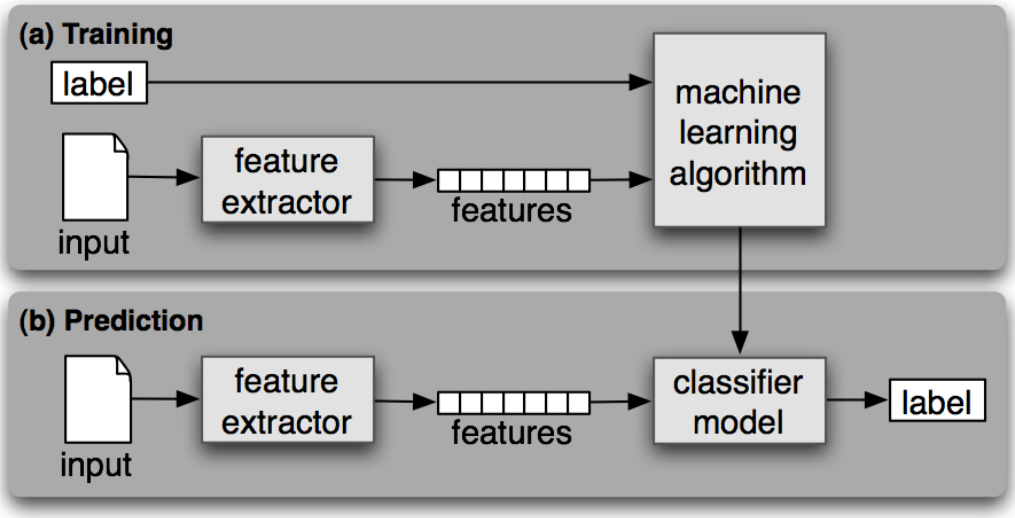

Figure 5 - Text classification processing

5. Project Specifications

5.1. Methodology

- **Agile Methodology**

  Why **Agile** (**Extreme Programming):**

  - To empower out team to manage our project easily.
  - Flexible in accepting changes for our future work like adding new algorithms and more features.
  - Parts of the often system can be deployed sooner.
  - Many tough problems can be addressed early in the project.

- **Technologies**

  - Machine Learning (Python – Sklearn – numpy – TensorFlow – Keras - Pickle).
  - Natural Language Processing (Python – NLTK).
  - Front-End (Web standard W3C – REACT).
  - Back-End (Python – Node JS – Flask).
  - Testing (Jest - Pandas).

5.2. Domain

- Artificial Intelligence
- Web software
- Scientific applications

5.3. Stakeholders

- Users (people who wanted to use this app to classify their document).
- Interested third parties (people who have interest in the Text classification field and want to use our app on their projects/researches).

5.4. System Architecture

As shown in figure 6 our system architecture is all about using our service (ML model) to show the layers of our application.

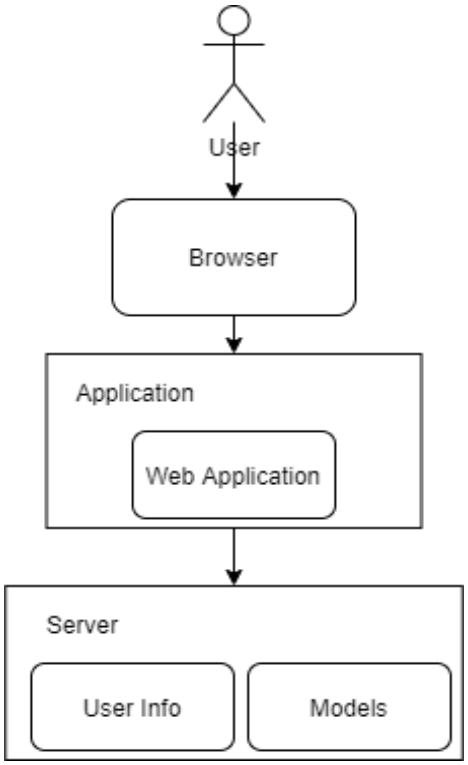

Figure 6 - System Architecture diagram

5.5. Functional and non-functional requirements

5.3.1. Functional Requirement

- Users can classify their document by uploading it as a text file or writing it directly.
- System takes the file and parses its text to sentences, paragraphs and words.
- System uses a classifier or multi-classifiers to classify the document.
- System output class or multi-classes of the document.

- Reporting our research on Text-Classification on our website.

5.3.2. Non-Functional Requirements

Performance	Calculation time and response time should be as little as possible, because one of the software's features is timesaving. The classifier takes a maximum response time of 5 seconds to predict the input data.
Usability	The system should be easy to use. The user should reach the summarized text with one or two button press if possible. The user doesn't need time to learn and train to use out web application. The system also should be user friendly.
Reliability	The classifier is developed with machine learning, feature engineering and deep learning techniques. So, in this step there is no certain reliable percentage that is measurable. The web application should handle and work under a lot of requests and large data input without failure.
Scalability	The system is designed to be scalable to add and remove algorithms and features.

5.4. Use-Case Diagram

As shown in figure 7, we have one use case that user uses the system to classify a document by uploading it either text input or a file and choose an appropriate classification algorithm.

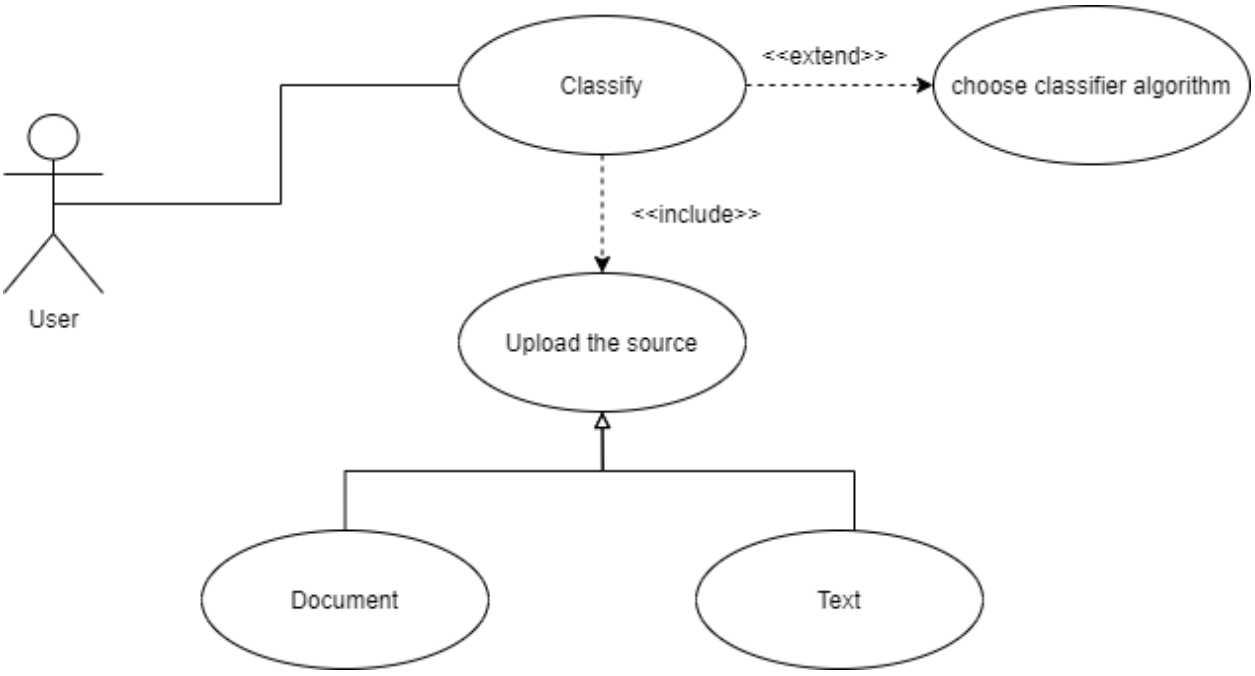

Figure 7 - Use-Case Diagram

5.5. Use-Case Description

Table 2 - Classification Use-Case Description

Use Case ID:	1
Use Case Name:	Upload and classify a Document
Actors:	User

Pre-conditions:	Document Should be text based *******	
Post-conditions:	The document is ready to be classified	
	User Action	**System Action**
	1- User uploads a documents or type a text.	
	2- User request to classify the document or the input text.	
		3- System validate user input and Preprocess user input.
		4- System classifies the document or the user input.
	User Action	**System Action**
	1- User uploads a non-text input.	
		2- Response to user invalid input format.

5.6. Class diagram

As shown in Figure 8 our class diagram for the interface model with different algorithms to use.

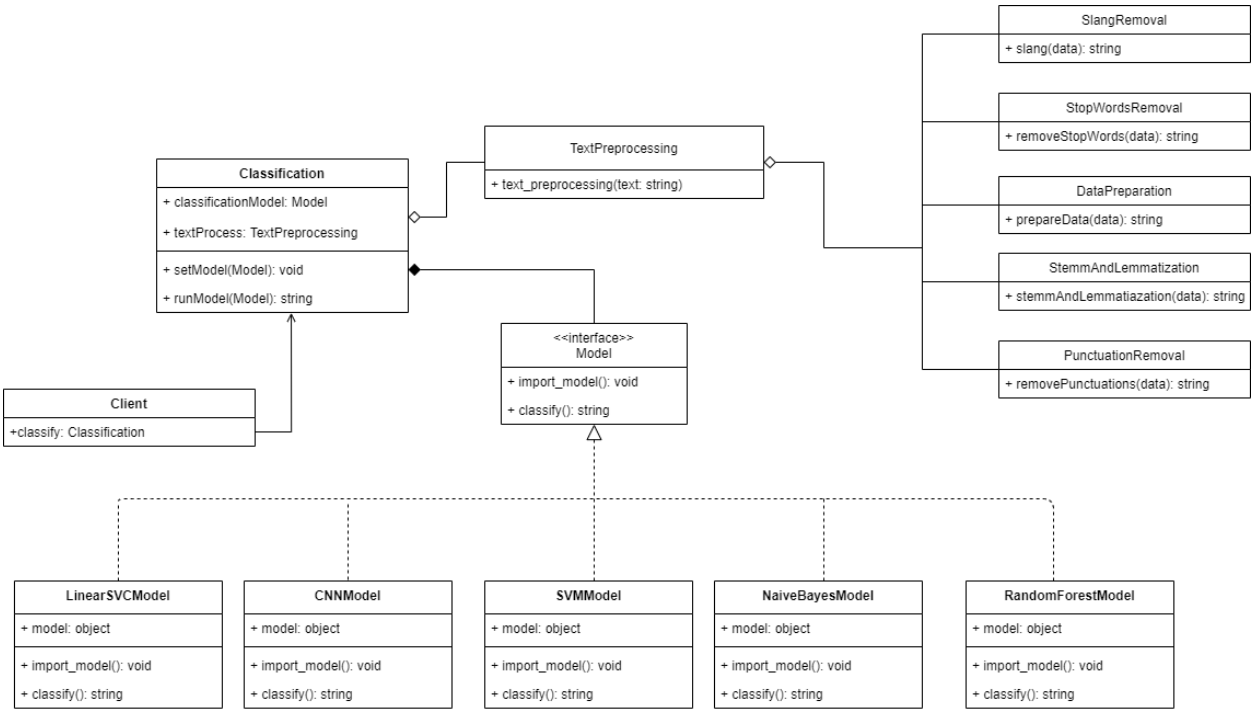

Figure 8 - Class Diagram

5.7. Sequence diagram

As shown in Figure 9 the user sends a request with the text to Classification class the TextPreprocessing Class works on the text the classify the word.

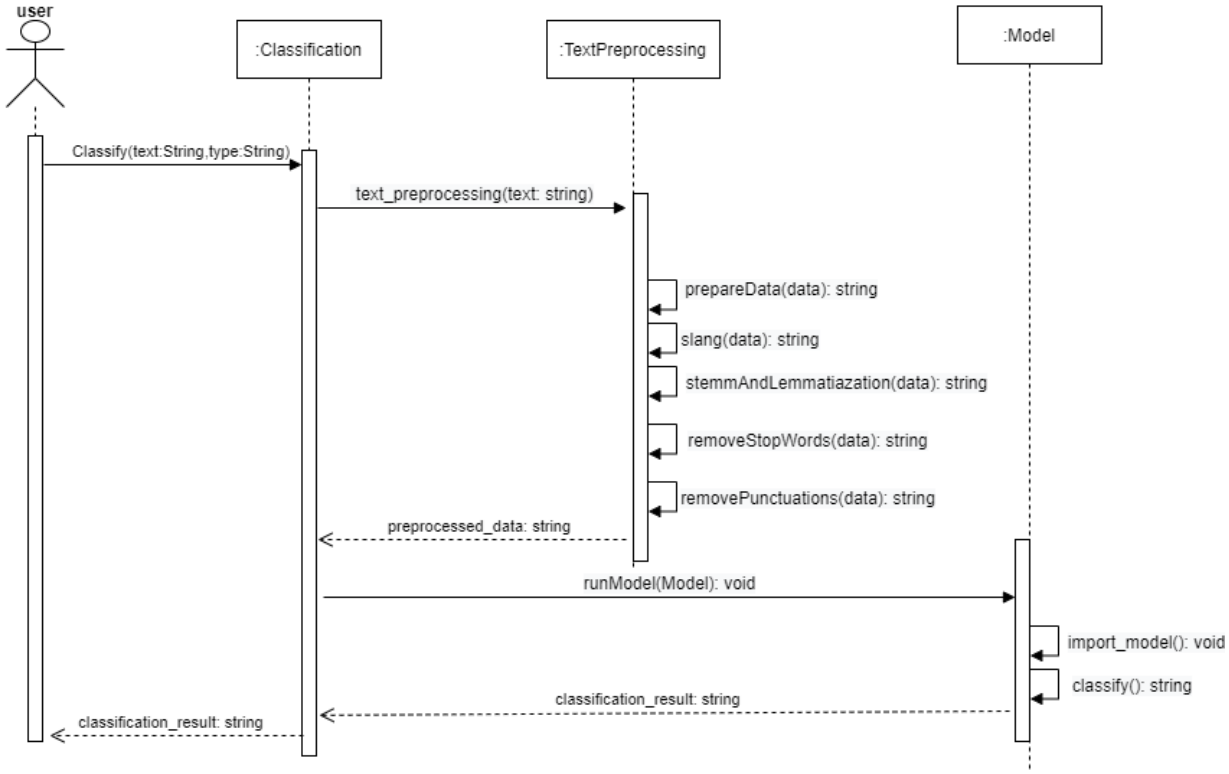

Figure 9 - Sequence Diagram

Work plan

Table 3 - Work Plan

Task	Task title	Description	Task status
1	Problem Searching	Search about text classification, techniques, algorithms and tools.	Done
2	Finding Datasets	Finding all available datasets and choose appropriate datasets.	Done
3	Start System Analysis documentation	Write full documentation about our project in details.	In Progress
4	Design Analysis	Design UML, Sequence Diagram and Use cases.	In Progress
5	User Interface	Design UI using React	In Progress
6	Back-End	Implementing Back-End Using Node JS and Flask	In Progress
7	Implementing Pre-Trained models	Use Pre-Trained models on our work	Future Work
8	Publishing our research on our website	Publishing our research for researchers on text/document classification field	Future work
9	Testing	Testing the models	Future work

Conclusion

Text classification is a mature area of research by the increase of information flow available. It has seen large attention especially due to the high growth rate of Internet and the importance of Internet search engines and generic classification of content on the Web. Process of text classification is well researched, but still many improvements can be made both to the feature preparation and to the classification engine itself to optimize the classification performance for a specific application. Research describing what adjustments should be made in specific situations is common, but a more generic framework is lacking. Effects of specific adjustments are also not well researched outside the original area of application. Due to these reasons, design of text classification systems is still more of an art than exact science.

References

[1] Monkeylearn.com. 2021. *Text Classification How text classification work*. [online] Available at: https://monkeylearn.com/text-classification/ .

[2] Monkeylearn.com. 2021. *Text Classification*. [online] Available at: https://monkeylearn.com/text-classification/ .

[3] Monkeylearn.com. 2021. *Text Classification Applications*. [online] Available at: https://monkeylearn.com/text-classification/ .

[4] Monkeylearn.com. 2021. *Text Classification*. [online] Available at: https://monkeylearn.com/text-classification/ .

[5] Monkeylearn.com. 2021. *Text Classification*. [online] Available at: https://monkeylearn.com/text-classification/ .

[6] Example, U., 2021. *Text Classification in Natural Language Processing*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/12/understanding-text-classification-in-nlp-with-movie-review-example-example/ .

[7] Example, U., 2021. *Text Classification in Natural Language Processing*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/12/understanding-text-classification-in-nlp-with-movie-review-example-example/ .

[8] Shaikh, J., 2017. Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK... [Online] Medium. Available at: https://towardsdatascience.com/machine-learning-nlp-text-classification-using-scikit-learn-python-and-nltk-c52b92a7c73a .

[9] HUILGOL, P., 2021. Pre trained Models For Text Classification | Deep Learning Models. [Online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/03/6-pretrained-models-text-classification/ .