BABEŞ BOLYAI UNIVERSITY, CLUJ NAPOCA, ROMÂNIA
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
SPECIALIZATION COMPUTER SCIENCE

Emotion Detection in Romanian Texts

– Diploma Thesis –

Supervisor
Lect. PhD Lupea Mihaiela

Author
Maxim Georgiana-Elena

2021

UNIVERSITATEA BABEŞ BOLYAI, CLUJ NAPOCA, ROMÂNIA
FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

Detectarea emoţiilor în texte în limba română

– Lucrare de licență –

Conducător științific
Lect. Dr. Lupea Mihaiela

Autor
Maxim Georgiana-Elena

2021

Abstract

The aim of this paper is the research of automatic emotion detection in Romanian texts, the creation of a Romanian corpus annotated with emotion labels through a web application, as well as the development of a tool for computing automatic emotion detection of texts and analyzing the performance of different machine learning algorithms.

Automatic detection of emotions is a significant task of the Natural Language Processing field, with extensive applications in e-learning, data mining, human-computer interaction, psychology or marketing. Although many tools and approaches have been developed for the English language, not as many advancements have been made for emotion detection of Romanian language texts.

This paper introduces the foundations for a Romanian corpus of emotion annotated texts, collected as a result of the development of web application, specifically designed with this purpose. This tool delivers a set of non-annotated short texts for each user, who will provide specific scores for a list of emotions.

A text emotion detection algorithm working with the previously defined corpus, employing binary and multi-label classification based on Naive-Bayes, Logistic Regression and Support Vector Machines is described, with a comparative analysis of the performance in terms of classifier choice. Given a Romanian short text and a chosen classifier, the algorithm will output the predominant sentiment polarity of the text, the list of emotions that are detected, as well as their probability percentages. These results are displayed using a user-friendly interface, represented by a web application.

The chosen technologies involved in the development of the corpus creation tool are Fastify, the Node.js web framework for the backend, with a Vue.js, JavaScript interface for the frontend server. The emotion detection tool backend works with Python, using Flask framework and Vue.js was utilized for the frontend, as well.

The paper describes its content in five chapters. Chapter 1 provides the motivation and purpose of the emotion detection application, as well as a background on related work. Chapter 2 gives details regarding the theory behind emotions and supervised machine learning classification. Chapter 3 describes the proposed approach. Chapter 4 introduces implementation details, architecture and technologies. Chapter 5 summarizes the outcome of the application, comparing the results of the classifiers. The final chapter outlines the conclusions and proposes future development ideas.

Contents

Chapter 1

Introduction

1.1 Motivation

Effective emotion recognition represents an important, if not vital factor in the communication between humans. In order to achieve an exhaustive human-computer interaction, the machine part would need to recognize the emotions expressed by humans, while also being able to express emotions through various aspects such as words, intonation, colors, and so on. A more natural interaction can emerge as a result of intelligent systems using emotion recognition, and therefore adapting their responses and behavioral patterns according to the signals sent by humans. Emotion recognition can be applied in several fields, such as education, medical diagnosis, patient care, video gaming, recruiting or retail.

Therefore, an intelligent learning system could help children learn while adapting the tasks difficulty to their emotion responses. Another software application may help doctors diagnose several diseases, such as depression or dementia.

However, emotion recognition is not an easy task, even for a human. Emotions are expressed differently among different people and can be expressed in numerous ways such as speech, facial expressions, text and other biosignals. Moreover, the accuracy of emotion recognition is still a big problem, especially in a real world setting.

Recognizing emotion from text is a significant topic, as the Internet usage continues to grow undoubtedly, more and more people are expressing their opinions and thoughts in the form of text and most news articles exist online. Emotions expressed through text may be recognized through the meaning and emotional valance of the words that appear in the analyzed text. In addition to this, understanding the context of the presented information plays a crucial role in determining these emotions.

The detection of emotions based on text represents a meaningful task in the Natural Language Processing field. Solving this task can contribute to advancements in several fields, such as e-learning, human-computer interaction, marketing and psychology. Considerable resources (corpora and lexicons), as well as notable studies of different approaches (keyword-based, rule-based, deep learning and hybrid) have been developed for the English language, whereas the Romanian language shows a lack in this direction.
The Romanian Emotion Lexicon (RoEmoLex) [27] has lately seen substantial improvement, as emotions present in Plutchiks Wheel of Emotion [37] appear as corresponding tags of Romanian words. On the other hand, datasets of Romanian language text for training classical learning or deep learning models are quite hard to find. In consequence, a corresponding texts dataset will be constructed.

1.2 Applications

The detection of emotions from textual data has real life applications in different domains. The medical field is representative of these domains, in which one important application involves emotion detection in suicide notes. Other fields are represented by social media and multimedia.

Sentiment analysis, as well as emotion detection are powerful tools to increase sales of different products, as most commercial websites have dedicated spaces for user reviews. Automatic analysis on these reviews contribute to finding out the weak spots of a product, for future improvement, as well as the features that users find great. Movie reviews can also be analysed from dedicated websites and so, rankings may be created.

Finally, the human-computer interaction can benefit greatly from an accurate recognition of emotions from texts, as well as other sources of input, such as visual or auditory. The role of emotion in human cognition and consciousness argues that emotion recognition in computer systems could not only increase human assisting, but also computers decision making ability. In this direction, Picard's work stands as a great foundation [38].

1.3 Objectives

The main purpose of this work is represented by research conducted on existing emotion detection algorithms, applying supervised machine learning algorithms for emotion detection for Romanian language texts, with a user-friendly interface represented by a web application, as well as creating the foundations of a Romanian texts corpus with annotated emotions, which could be further developed and used in other applications.

1.4 Related Work

The computer science field which specializes in text data processing, namely the Natural Language Processing field has its origins in the years 1950, starting with the Russian to English text automatic translation Georgetown Experiment [22]. A significant amount of NLP tasks, such as machine translation, document summarization, part of speech tagging, named-entity recognition, sentiment analysis have been successfully solved up to the present day. These discovered techniques have been integrated in many state-of-the-art applications.

As stated above, the affective computing field started its development in 1997 with Picard's work underlining the importance of automatic emotion detection in the interaction between humans and computers.

One such task is the emotion recognition and sentiment analysis of textual data. Although great advancements have been made in the development of various algorithms in recent years, to be able to detect and classify emotions from text is a complex task to solve. Many factors such as context, language specific features and emotions complexity contribute to the difficulty of obtaining a near perfect result.

In the context of NLP Emotion Detection, several techniques have been used: keyword-based, rule-based, classical learning, deep learning, as well as hybrid approaches [3].

The **keyword spotting approach** focuses on finding the keywords that may place the text in a specific emotion category [11]. Such a proposed model [28] was developed for a chat system. Starting from emotion keywords. Incorporating their synonyms, which were determined using WordNet and WordNet-Affect, weights were assigned in order to determine a sentence level score.

A **rule-based approach** focuses on determining rules based on different concepts, including linguistic, computational and statistics. Prominent work in this area was conduced by Lee et al.[26] for the Chinese language texts.

The **classical learning algorithms** involve machine learning and revolve around finding important features, useful for the training of a Support Machine Vector. Alm et al.[2] 's model represents one of the first models developed for emotion recognition. The work done in this paper is a part of this category of algorithms.

Deep learning methods include the use of LSTM (Long short-term memory) models or CNN (Convolutional neural network) models and word embeddings for textual data representation.

Hybrid models combine other approaches to get better results, but at the same time inherit the flaws of each approach. For example, a CNN-DCNN Autoencoder [25] that includes linguistic features with a joint reconstruction loss has proven to improve performance. Also, a combined network based on a SENN model, using CNN for emotions and BiLSTM for semantics [6] has proved to outperform other models.

An important work of the emotion detection task is its application in suicide notes [45] [15], having as potential impact a decrease in the suicide rate.

The emotion detection task has also been studied in social media, a space where more and more people express their opinions regarding products, news, concepts and their daily life emotions, where analyses on emotions have been conducted [20]. Moreover, emotions present in multimedia tagging have been studied [44]. Detection of insults in conversations have also been studied in this direction [1].

Needless to say, text representation has specific language features. For the Romanian language, a very important resource is represented by **RoEmoLex** [27], a lexicon that contains a significant amount of Romanian words, tagged with eight emotions and two valences. Having arrived at its third version, this lexicon has seen substantial improvement and consists a reliable resource for the Romanian language. An entrance in this lexicon is presented under this format:

	word	part_of_speech	Pozitivitate	Negativitate	Furie	Anticipare	Dezgust	Frica	Bucurie	Tristete	Surpriza	Incredere	wn_synset_id
0	se izbi	Verb	0	1	0	0	0	1	0	1	1	0	ENG30-01561819-v

Figure 1.1: RoEmoLex Entry [27]

1.5 Original contributions

The research conducted for this paper focuses on the existing solutions for detecting the basic emotions present in a piece of text and finding a suitable approach for this task for the Romanian language texts.

The contributions of this paper are represented by :

- The creation of a consistent corpus of Romanian language sentences from diverse sources such as blogs, articles or even social media posts, labeled with emotions and sentiment polarity. A database with such labels necessary for machine learning algorithms that solve the NLP tasks of emotion detection and sentiment analysis is rather absent for the Romanian language texts, although many such datasets exists for the English language. In this direction, a publicly available website has been developed.

- The comparative study of the performance of different supervised machine learning algorithms, using the formerly created Romanian language corpus and specific features processed with the use of RoEmoLex.

- The development of an emotion detection tool, represented by a user-friendly website. This way, any user can navigate through and discover the existing emotions in a Romanian language text they give as an input.

1.6 Paper structure

The paper introduces its content in five chapters. Chapter 1 provides a short preface into the motivation and purpose of the emotion detection application. Chapter 2 gives insight into the scientific problem, the concept of emotions and their type, the field of Natural Language Processing, as well as machine learning classification algorithms involved in solving this task. Chapter 3 describes in detail the creation of the emotion corpus, as well as the detection machine learning algorithms for the proposed application. Chapter 4 introduces implementation details, architecture and technologies. Chapter 5 summarizes the outcome of the application, comparing the results of the different machine learning classifiers. The final chapter outlines the conclusions and proposes future development ideas.

Chapter 2

Theoretical Concepts

In order to touch the NLP task of emotion detection, there are a few critical theoretical aspects that need to be taken into consideration, starting from the definition of emotions and continuing with the machine learning concepts that are being used as part of the tool that detects these emotions from textual data, that are incorporated in the proposed solution.

2.1 Emotion Models

The first questions to be addressed for emotion detection are what emotions are and how many emotions there exist. One of psychology field's main interest is emotions, how they are formed and the way they impact our daily life. There exist 3 main approaches to categorize emotions: categorical, dimensional and appraisal-based approach [3].

2.1.1 Categorical Models

This approach revolves around the idea that emotions can be divided into a finite number of basic emotions, that are universally perceived [21]. For this approach, the model that is most frequently approached in emotion detection is Paul Ekman's model [16], which includes six basic emotions: **happiness**, **sadness**, **surprise**, and **disgust**, **anger**, **fear**.

On the first hand, based on the study of facial expressions characteristics, Ekman concluded the existence of six basic emotions , namely disgust, fear, sadness, anger, surprise and happiness, to which he later added contempt, embarrassment, pride, shame, satisfaction, excitement and amusement.

On the other hand, Plutchik's theory [37] introduces a list of basic emotions, whose combinations outcome are other, more complex emotions. The way he illustrates this theory is associating emotions with colors in a color wheel (2.1).

As seen below, the 8 basic emotions in the wheel are **happiness**, **anger**, **trust**, **surprise**, opposed to **sadness**, **fear**, **disgust**, and **anticipation**. Furthermore, the combination of the neighbouring emotions on the wheel form other emotions : aggressiveness is composed of anger and anticipation, optimism of anticipation and joy, love of joy and trust, submission of trust and fear, awe of fear and surprise, disapproval of sadness and surprise, remorse of disgust and sadness, contempt of anger and disgust. It is clearly shown in the color wheel how the intensity of the colors correlates with the intensity of the emotions.

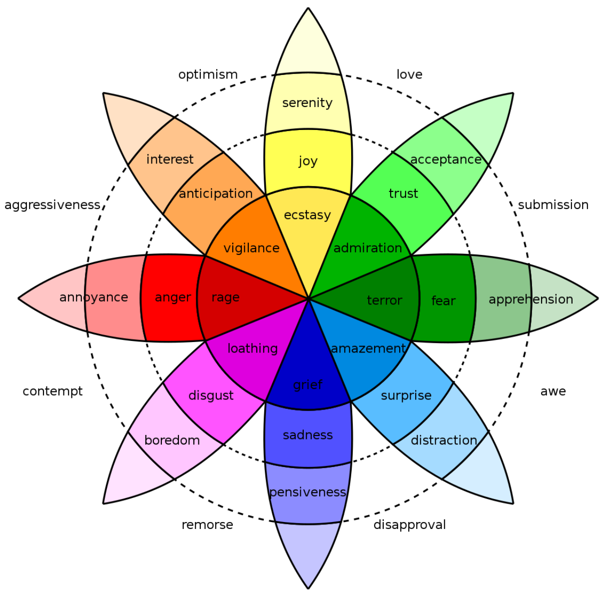

Figure 2.1: Plutchik's Wheel of Emotions [30]

2.1.2 Dimensional Models

This approach to categorize emotions stems from the concept of emotions in relation to each other, represented as a system.[21] They are not independent, nor existing in a limited number. The system in which emotions are represented is three-dimensional, those dimension being determined by valence, arousal and power.

The scale which determines the positivity or negativity of an emotion is represented by valence. The arousal dimension measures the excitement or apathy of the emotion, while power specifies the intensity to which an emotion is expressed.

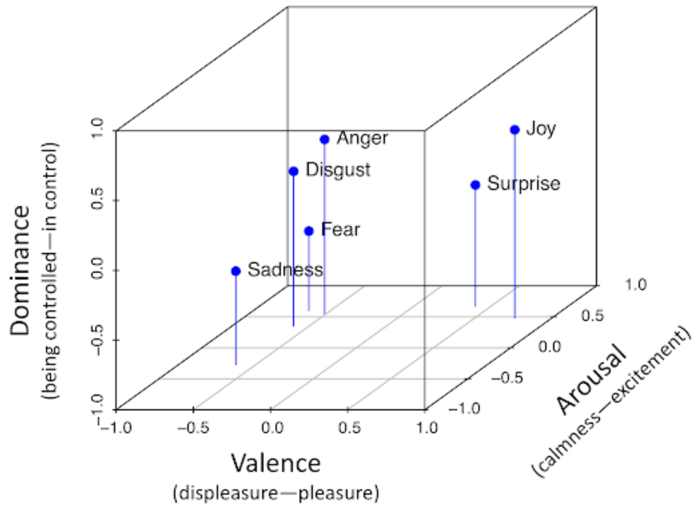

Figure 2.2: VAD Model [7]

2.1.3 Appraisal-based Models

These models have as foundation the appraisal theory. [14] Starting from the theory that emotions are highly complex, developed as an accumulation of experiences that each and every person lives, they are perceived differently by everyone, based on personal factors that include motivations and ambitions.

2.2 Classification

Classification, from the point of view of machine learning models, involves problems that deal with predictive modeling. This means that for an example input, its class label or the probability of membership of that class is predicted.

An important part of classification is represented by the required dataset used for training, which should be characteristic to the problem and have enough data input for all class labels involved.

The main types of classification tasks that this paper focuses on are represented by:

- **Binary Classification** - which involves two output classes

- **Multi-Class Classification** - which involves more than two classes and each input sample belongs to exactly one of them
- **Multi-Label Classification** - which involves two or more classes and each input sample may belong to one or more classes

2.2.1 Naive Bayes Classifier

Naive Bayes methods are known to have a good performance for tasks such as document classification or spam filtering. They are based on Bayes Theorem and work on the assumption that any feature taken into consideration is independent. This assumption is often wrong in real-life settings, from which the name "naive" comes from.

The advantages of these classifiers include the fact that they require a relatively small amount of data for training, they are fast and they are not sensitive to features that are not so relevant.

The Bayes Theorem is depicted as follows:

$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$

, where H is some hypothesis that states that X belong to a specific class, X is a predictor, some evidence specified by measure on the set of attributes.

$P(H|X)$ is the probability that H holds, given X.

$P(H)$ is the probability of of H, not taking into account X.

$P(X|H)$ is the probability of X, conditioned by the hypothesis H. [8]

Types of Naive Bayes classifiers include Multinomial, Bernoulli and Gaussian Naive Bayes Classifiers.

2.2.2 Support Vector Machine Classifier

Support Vector Machines are used for regression, classification and outliers detection.They have the goal to determine the hyperplane (decision boundaries location) which outputs the optimal separation of classes, being based on statistical learning.

These algorithms center around maximizing the side of a hyperplane which separate two classes, also known as “margin“. This has been proven to lower the upper bound on the expected generalisation error. After being defined for the linearly separable case, it uses kernel functions for nonlinear mapping in order to convert the initial data used for training into a higher dimension, searching for the linear optimal separating hyperplane. [8]

The advantages of these supervised learning methods include their effectiveness in spaces with high dimensions or in which the dimensions are bigger than the samples number. They are efficient from a memory point of view, due to support vectors use in the decision function.

Some disadvantages include the fact that estimates are calculated indirectly, using five-fold validation and the over-fitting risk in case there are much more features than samples.

2.2.3 Logistic Regression Classifier

Logistic regression is a probabilistic statistical method, commonly used in binary classification, whose outcome on a sample is the probability of positivity or negativity, probability which depends on a linear measure of the specified sample. [18]. The Logistic Regression model uses the sigmoid function:

$f(x) = \frac{1}{a+e^{-(x)}}$

Some properties include the fact that estimation is computed using maximum likelihood and that the fitness is computed through Concordance, KS-Statistics.

2.2.4 One-vs-Rest Classifier

The One-vs-Rest or One-vs-All strategy revolves around splitting a multi-label classification. N binary classifiers with one class are trained at a time, the rest of the classes are left out. For multi-label classification, the problem is treated as mutually exclusive on the existing classes, creating one classifier for each class. Some problems may arise in case of a big number of classes. One of its main advantages is its computational efficiency. This approach is highly utilized for multi-class classification and multi-label classification.

Chapter 3

Application

This chapter focuses on describing the application in two steps: the data collection of the annotated Romanian texts, the processing steps performed on the collected data in order to obtain datasets used by the classification algorithms, as well as details regarding these used machine learning classifiers.

3.1 Proposed Approach

As depicted in figure 3.1, the first step of the proposed approach consist of creating the annotated emotions corpus, meaning gathering data from the deployed website, specially created for this use. This data is stored in a dedicated database and processed, creating training and testing datasets for the chosen classifiers. These will then predict the emotion class a new given input text is part of.

Once the database has collected annotated texts with emotion scores from 0 to 5, the following several processing steps are being performed:

1. Computing the scores for the eight basic emotions
2. Creating a dataset containing labels 1 for the predominant(highest score) sentiment polarity (Positivity/Negativity) emotion and 0 for the other one
3. Creating a dataset containing the existing emotion(with scores $>$ 0) labeled as 1, the others as 0

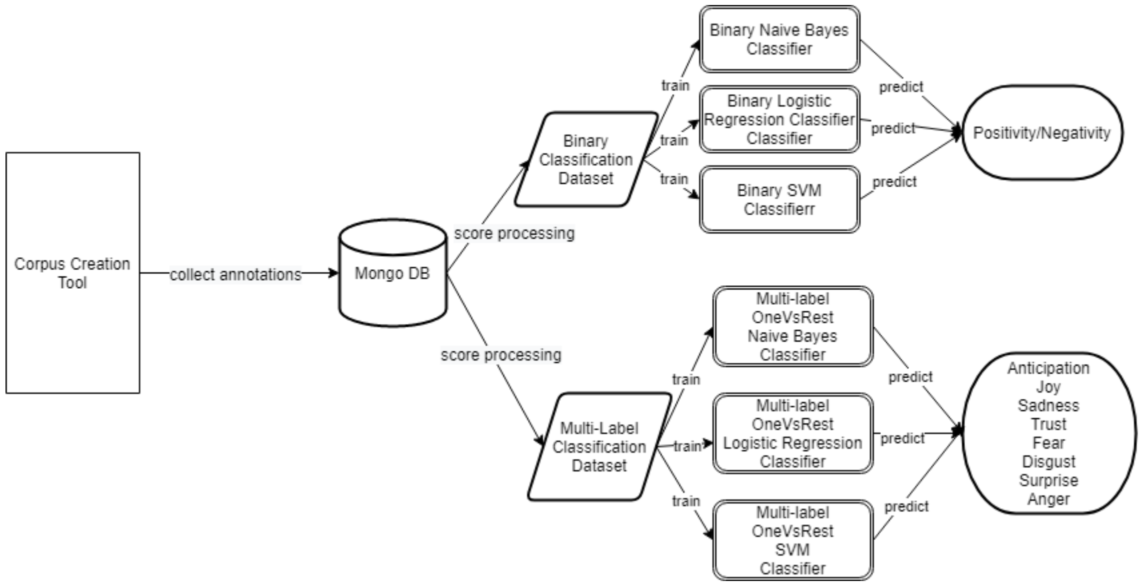

Figure 3.1: Proposed approach flow

Having these three datasets, we can consider three classification types:

1. *Binary Classification* - for the binary predominant sentiment polarity emotions considered as class (Positivity, Negativity)

2. *Multi-Label Classification* - for the emotions existing in the text, the eight basic emotions considered as classes (Anticipation, Joy, Sadness, Trust, Fear, Disgust, Surprise and Anger)

For each type of classification, training will be performed on this dataset with three types of classifiers(Naive-Bayes, Support Vector and Logistic Regression). Depending on the best accuracy results, for each one, the best performing model will be saved and later used for predicting the results for each classification.

The saved models are used in the backend of the Emotion Detection Website, in order to predict, given an input text:

- The predominant polarity sentiment, meaning whether the text information is Positive or Negative

- All the possibly existing emotions in the text

The website will give a user-friendly interface, making it easy for the user to see the achieved results.

3.2 Corpus Creation

In order to train any type of model to predict the emotion classes of an input text, a robust dataset should be used. The dataset for the emotion classification task should contain entries with an input text in the chosen language and its corresponding emotion annotations, meaning that each class represented by an emotion should appear with a 0-1 flag: 1 represents the existence of the emotion in the text and 0 otherwise.

Several datasets of considerable size and acuuracy have been created for the English language. Taking into consideration the lack of such a dataset for the Romanian language, I have attempted to develop such a resource to use in my algorithm.

3.2.1 Sources

The texts that are used in constructing a emotion classification dataset can be extracted from various sources, such as social media posts(tweets, facebook posts), news, articles on a specified subject, blogs, books, etc. The created dataset combines text sequencies that revolve around different topics from a variaty of Romanian online blogs:

- tudorchirila.blogspot.com society
- ciutacu.ro television
- bookblog.ro book reviews
- groparu.ro humor
- petreanu.ro social
- codulluioreste.ro turism
- alexunu.blogspot.com journal
- serbanhuidu.ro politics

In the inital iteration, 750 pieces of texts have been extracted from these blogs. Each piece one-two sentence phrases that may express the chosen emotions or not.

3.2.2 Emotions Labels

In order to annotate the extracted texts, several emotion labels have been taken into consideration. First of all, the sentiment polarity emotions have been considered as labels: Positivity (Pozitivitate) and Negativity (Negativitate). Secondly, the 8 primary emotions represented in Plutchiks wheel of emotions have been added: Anticipation (Anticipare), Joy (Bucurie), Trust (Încredere), Fear (Frică), Surprise (Surpriză), Sadness (Tristețe), Disgust (Dezgust) and Anger (Furie). Finally, as some of the primary emotions, such as anticipation, are harder to be detected, some secondary emotions have been added to the labels list: Love (Iubire), Optimism (Optimism), Pessimism (Pesimism), Regret (Regret), Shame (Rușine), Guilt (Vinovăție).

3.2.3 Annotations

The way each text is designed to be annotated with an emotion label is by assigning a score from 0 to 5. The scores are perceived this way:

0	not at all	The emotion is not expressed at all in the text.
1	very little	The emotion is expressed in the text, but with a very low intensity.
2	little	The emotion is expressed in the text, but with a low intensity.
3	average	The emotion is expressed in the text with an average intensity.
4	a lot	The emotion is expressed in the text moderately intense.
5	very much	The emotion is expressed in the text very intense.

The goal is to have 2 annotators for each piece of text. Emotion annotation is a rather difficult and imprecise task, given the subjectivity of how different people perceive emotions in a text. Based on the individual scores of an annotated unit by 2 different annotators and taking into account the impact of the secondary emotion labels scores on the primary emotions they are composed of, a final score is being calculated in the following way:

- Scores of the primary emotions labels for a single unit

  Secondary emotions are composed of different primary emotions. Therefore, Love is a combination of Joy and Trust, Optimism of Anticipation and Joy, Pessimism of Anticipation and Sadness, Regret of Disgust and Sadness, Guilt is believed to be a combination of Joy and Fear and finally, Shame is considered to be tied to Disgust and Fear. Scores have been added to the primary emotions in the following way:

$score_{Anticipation} = max(score_{Anticipation}, score_{Optimism}, score_{Pessimism})$

$score_{Joy} = max(score_{Joy}, score_{Optimism}, score_{Love}, score_{Guilt})$

$score_{Disgust} = max(score_{Disgust}, score_{Shame}, score_{Regret})$

$score_{Fear} = max(score_{Fear}, score_{Guilt})$

$score_{Trust} = max(score_{Trust}, score_{Love})$

$score_{Sadness} = max(score_{Sadness}, score_{Regret}, score_{Pessimism})$

- Final score for an emotion label after 2 annotations

  After the first calculation of the separate annotations of each annotator and taking these scores into consideration, the final score for an emotion has been computed as 0 if either one of the annotators has given a 0 score for the emotion label or the arithmetic mean of the separate scores for each emotion label, otherwise.

3.3 Emotion Classification

The proposed approach for the classification of emotions can be split into two types of classification:

- The classification of sentiment polarity emotions - Negativity, Positivity
- The classification of the eight primary emotions - Anticipation, Joy, Trust, Fear, Surprise, Sadness, Disgust and Anger

Given a short input text, containing one/two phrases, the goal is to determine, for the first classification, whether the text is positive or negative, and for the second classification, to determine the list of emotions that the text is represented by, as well as the probability of these predictions.

For both types of classification, there are several similar aspects to be taken into consideration, including a preprocessing of the text data, the representation of obtained textual data in vector terms, as well as the extraction of representative features.

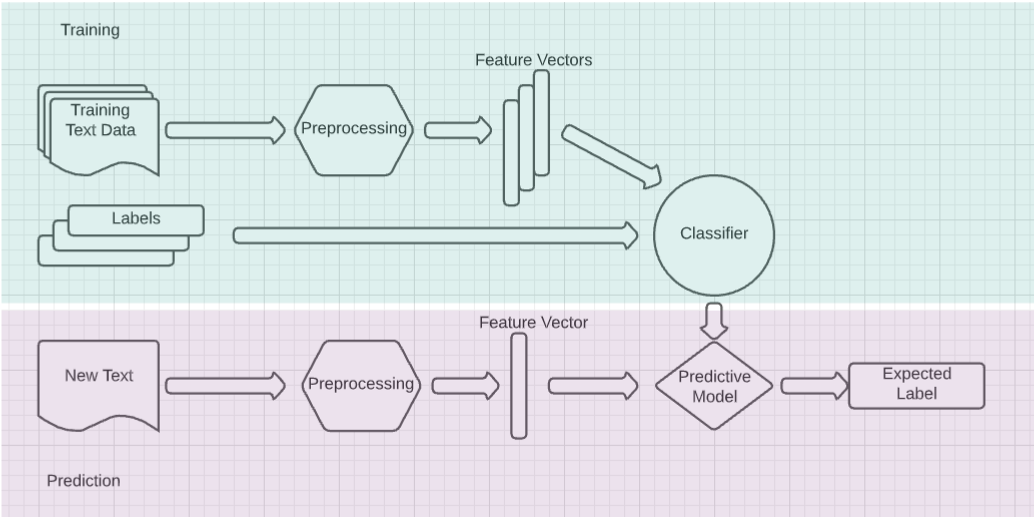

Figure 3.2: Classification flow

3.3.1 Text Preprocessing

Text preprocessing techniques include stop-words removal, punctuation marks removal, removal of numeric terms, lemmatization or stemming.

The proposed preprocessing of a text, for this approach, focuses on text lemmatization, since punctuation marks and stop-words are going to be used in the feature extraction, which is being further discussed.

The lemmatization of a text revolves around the replacement of the words contained in a sentence, by their lemma. By applying lemmatization to an inflected word, its root form is generated. Both stemming and lemmatization obtain the root, the difference being that the stem may not be an actual word, as opposed to the lemma.

In addition, a significant part of the Romanian language words contain diacritic characters, which are not always correctly placed in text-written data. To solve the discrepancy of the diacritics containing and non-diacritics containing words, an additional preprocessing is represented by the replacement of diacritics with their corresponding letter: ă <-> a, â <-> a, î <-> i, ș <-> s, ț <-> t.

Therefore, the preprocessig of text is represented by text lemmatization, followed by replacing of the diacritics characters.

3.3.2 Text Representation

As machine learning classifiers do not work with textual data, but rather with numerical data, a text to vectors of numbers conversion needs to be made.

Bag of Words is a widely used technique for modeling text data.Using this approach, the textual data is converted into a fixed-length vector depicting the occurrence of words existing in a document. This method is best applied with a previous text processing step.

Another approach for the text representation is Tf-Idf, Term Frequency - Inverse Document Frequency. While the Bag of Words approach takes into account the number of occurrences of particular words, Tf-Idf implements a rescaling of the words frequency by their occurence in all documents.

Since the Bag of Words method has its limitations, the proposed approach is to use Bag of Words representation, to which a Tf-Idf Transformer is applied.

3.3.3 Feature Extraction

Having a direct impact on text classification accuracy, feature extraction is a crucial step in applying a machine learning algorithm. The first, most important feature is the previously described vector representation of textual data.

In direct relation to the emotion classification task, the following features have also been taken into consideration:

- The number of **amplifiers**, words which amplify the force of words in a phrase, such as "foarte", "extrem", "mult", "super"
- The number of **downtoners**, the opposite of amplifiers, they reduce the force of a statement, such as "deloc", "puțin"
- The number of **negative markers**
- For each emotion taken into consideration, the number of emotion words extracted from the RoEmoLex lexicon

The appearance of negation words plays an important part in the overall meaning of a sentence. Placing a negation word before a positive adjective, for example, turns the overall polarity of the adjective into negativity. Therefore, another text processing is performed during the extraction of emotion features in the counting of emotion words step. [24]

While parsing individual texts, when finding a negation word, the approach is to append the prefix "NU_" to all the following words, until the appearance of a punctuation mark. Therefore, when computing the number of words corresponding to a emotion class, if the word included in an emotion class is prefixed by "NU_", the algorithm counts it as corresponding to the opposite class. The opposite emotion classes are considered as following, according to Plutchikș wheel:

- Positivity <-> Negativity
- Joy <-> Sadness
- Trust <-> Disgust
- Fear <-> Anger
- Surprise <-> Anticipation

3.3.4 Classification

For the binary classification, two class labels are taken into consideration: **Positivity** and **Negativity**. On the other hand, for the multi-label classification, eight classes are taken into consideration: **Anticipation**, **Joy**, **Sadness**, **Trust**, **Fear**, **Disgust**, **Surprise** and **Anger**.

Training

For the binary classification, the emotion dataset is trained on three types of classifiers: **Logistic Regression**, **Naive Bayes** and **SVM**. The multi-label classification dataset uses the One-vs-Rest classifier, which divides the problem in multiple binary problems, that are also solved using all three types of classifiers.

The features used in the classifications differ only in terms of the extracted terms from RoEmoLex which are counted, the binary classification makes use of positive and negative connotations of words, while the multi-label classification takes into account the other eight primary emotions in terms of what the RoEmoLex words express.

Testing

The dataset is split in proportions of 80% for training and 20% for testing, in a random order. The data is shuffled before this stage of splitting for training and testing. This approach is needed in order to achieve a model which has the best performance on repeated training-testing steps.

For the binary approach, a classification reports is computed, which contains the precision, recall and F1 score for each of the two classes - class 0 (Negativity) and class 1 (Positivity). Also, the macro average, which is the average of the unweighted mean per label and the weighted average, which is the average of the support-weighted mean per label are included, besides the accuracy for F1-score.

Precision is the proportion of the true positive outcomes from the total(true and false) positive outcomes.

$precision = \frac{TP}{TP+FP}$.

Recall is calculated as the proportion of true positive outcomes from the sum of the true positives and false negatives.

$recall = \frac{TP}{TP+FN}$.

The **F1-Score** is calculated as :

$f1 - score = \frac{2*(precision*recall)}{(precision+recall)}$.

For the multi-label approach, the **accuracy score** is calculated for each class, meaning the fraction of correctly classified samples.

$accuracy = \frac{TP+TN}{TP+FN+FP+TN}$.

For both approaches, the results of the different classifiers are compared in terms of the above mentioned reports, with and without emotion-specific features.

Chapter 4

Implementation

This chapter describes in grater detail the technologies used for the two developed tool: the corpus creation framework, as well as the emotion detection one. Both of these frameworks are represented by web applications and therefore conform to the **REST**(Representational State Transfer) architecture.

4.1 Corpus Creation Framework

For the purpose of offering a friendly and easily accessible user interface to those willing to contribute to this corpus creation, the website dedicated for annotating phrases with emotions was deployed publicly.

For the backend, a **Fastify** [17] server was created that interacts with a **MongoDB** [31] where data is stored and appropriate units are returned to annotate for each user. The server was deployed as a lambda function at https://yiup480roe.execute-api.us-east-1.amazonaws.com/, using Amazon Web Services [4].

For the User Interface, a Vue.js [42] app was developed and deployed at ro-emotion-detection-ds.netlify.app using netlify [32].

4.1.1 Storage

The first aspect to consider for this dataset was the way to store data, as more inputs were added. The chosen storage is a MongoDB cluster database containing 3 collections: sentences, annotations and final_annotated_sentences.

The sentences collection holds information about the text units, each having a unique id number. The annotations collection is composed of individual annotations, each element having as fields the emotion labels and the id of the annotated text. The final_annotated_sentences contains the computed scores of each annotated unit.

The MongoDB cluster is deployed as a Shared Tier Cluster, AWS / N. Virginia (us-east-1) as a Replica Set with 3 nodes. The network access is not restricted, meaning any IP Address can access the cluster.

4.1.2 Backend

The backend of the this framework's main purpose is to interact with the MongoDB Storage. Following this goal, a **Node.js** [34] backend was developed, using the Fastify web framework.

The following routes are exposed, routes that fulfill the purpose of the database interaction in order to perform the needed operations with sentences and their annotations:

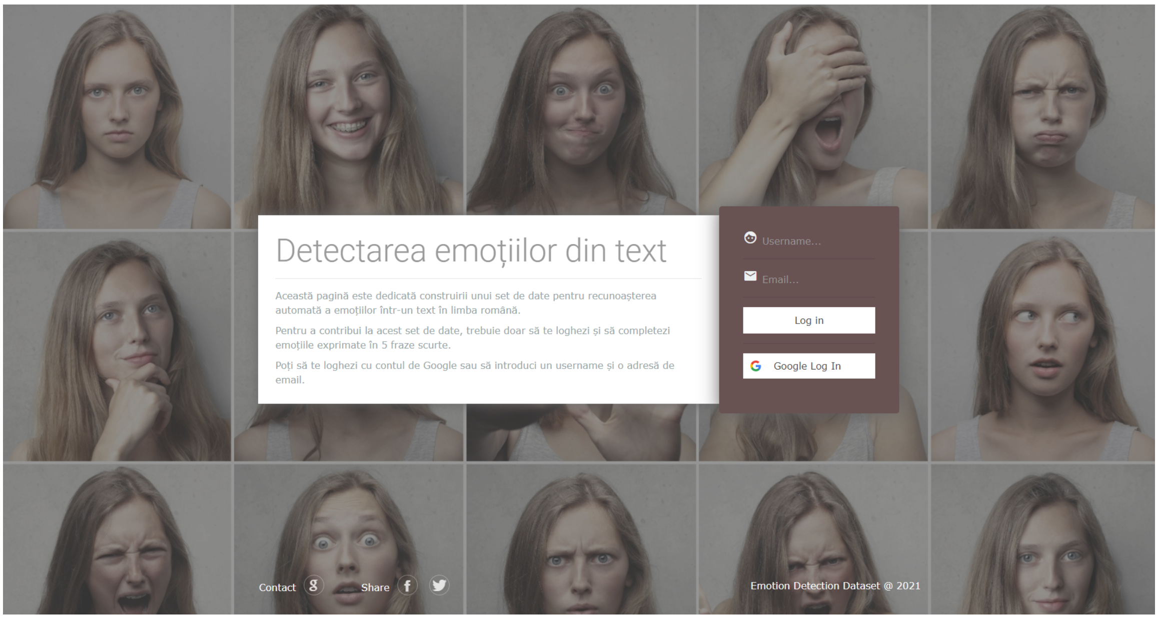

Figure 4.1: Login page

- **/annotations**

  GET - returns a list of all existing annotations in the database

  POST - adds an annotations to the database

- **/sentences**

  GET - returns a list of all existing sentences in the database

  POST - adds a new sentence to the database

- **/ed**

  GET - returns a list of 5 sentences that do not have two annotations and that the user has not previously annotated

For the validation of the request data, **JSON Schema**[23] was used, implemented with the use of the **ajv** package. As a brief description of this schema, it specifies the properties that should be included in each request. As for the database interaction, the **mongodb** Node.js package was used.

4.1.3 Frontend

The frontend of this tool was created using the Vue.js JavaScript Framework. It includes a login page, seen in Figure 4.1 and an annotations page, Figure 4.2. The used background images' source is presented at [13]. In order to use, the framework, the user must either sign in with a google account, with the hep of the npm package vue-google-auth [43] which performs a redirect to the google login, or provide a user name and a password. For the time being, no accounting creating approach was created, as for now, the goal is only to identify the user by a email address, with the goal to not give sentences that are already annotated two times to a user and not give the same sentences to the same user.

The annotations page presents a short descriptions of how annotation scores should be given and has three main sections: the section where the user chooses emotion scores for each sentence, the section where scores meaning is presented and a section with an annotation example.

4.2 Emotion Detection Tool

The emotion detection tool consists of two main parts, the backend classification algorithms and the user interface. The classification application, where Python [39] language with its specific libraries was used in the flow of training classification models, getting performance

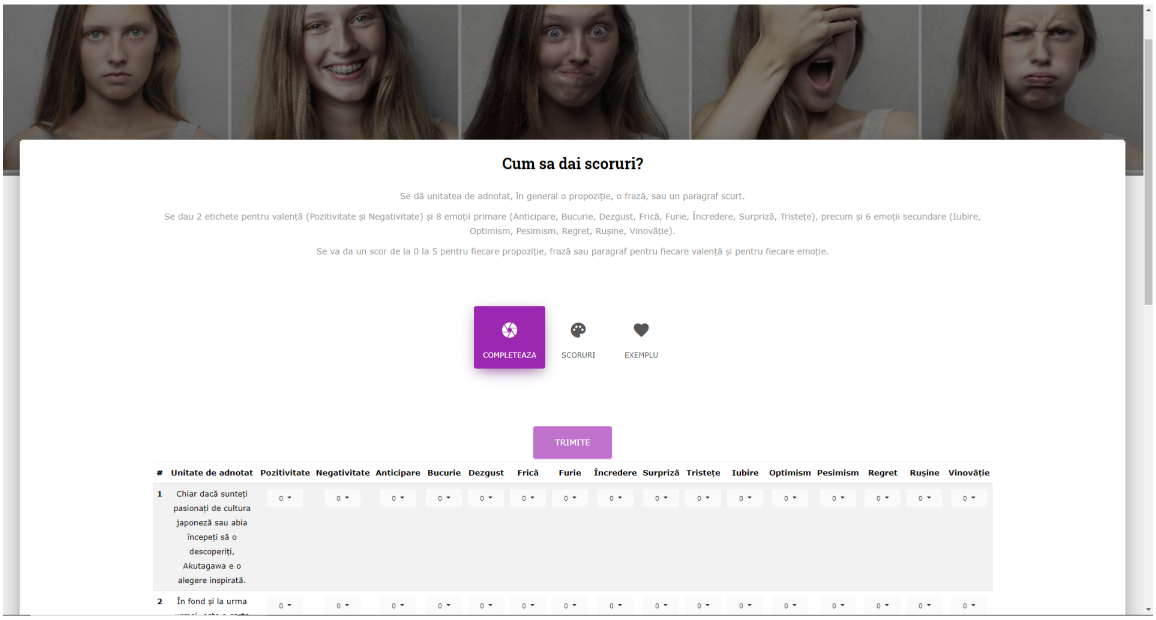

Figure 4.2: Emotion annotation page

metrics for data predictions and predicting probabilities for new text data. As for the server part, the Python Fastify[17] framework was used to start a server locally. The frontend user interface was designed using Vue.js, similar to the frontend of the corpus creation tool.

Class Diagram

In order to structure the functionalities of the emotion classification tool, several classes have been created, as it can be seen in Figure 4.3, each one with its specific purpose. Text Processors are represented by NLPCubeProcessor and RoEmoLexProcessor classes. NLPCubeProcessor processes text data that is later introduced in the classifiers, while RoEmoLexProcessor loads the RoEmoLex lexicon and determines the weights for each emotion label, given a new text. The Features class maintains a list of features that are given to the classifiers, these features are represented by function that are different for binary and multi-label classifiers. The Classifier class has prediction and training functions, as well as the function to initialize the classifiers with their appropriate features. The Classification class work as a controller of the classifiers, but most importantly works with the dataset texts and their preprocessing. BinaryClassification and MultiLabelClassification inherit the properties of processing the input data and splitting it for tests and training, but require different implementations for the work with classifiers.

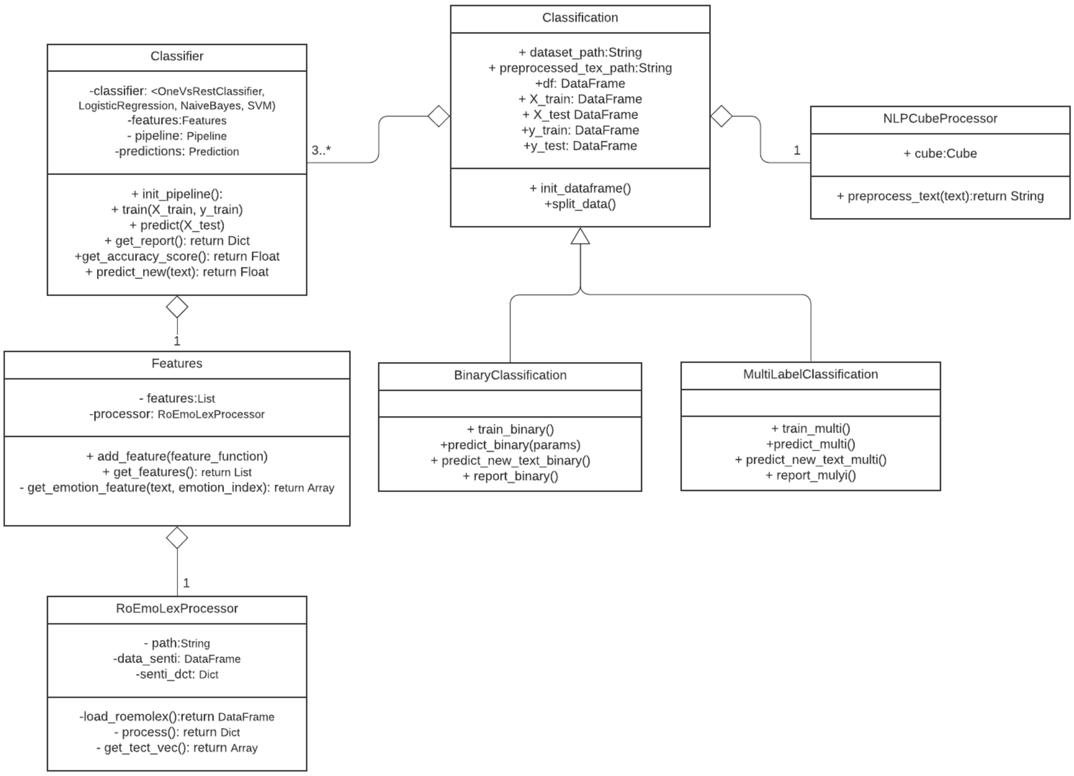

Figure 4.3: Class Diagram

4.2.1 Tools

The lemmatization of sentences is being done using NLP-Cube [33], a Natural Language Processing Framework. Starting with sentence segmentation, the lemma for each word is extracted in the preprocessing step of the algorithms, using the nlpcube Python library for the Romanian language.

The machine learning classification for binary and multi-label approaches, as well as the train-test data split and report metrics are performed using the scikit-learn[40] Python tool for machine learning. Namely the classifier classes that are used are OneVsRestClassifier, MultinomialNB, LogisticRegression and SVC with linear kernel. CountVectorizer and TfIdfTransformer classes are included for text data processing, which are included in the features of the classifiers.

Other useful Python libraries used in the proposed implementation are numpy[35], used for working with arrays, especially in the processing of the RoEmoLex lexicon, as well as pandas[36] for the reading of the datasets presented as .csv files and of RoEmoLex, a .xlsx file. The library pickle is also used for storing the trained models for later use of new predictions.

4.2.2 Backend

The server part of the application consists of a **Fastify** server, which is not yet deployed. It runs on localhost and the main path /**predict** works with http GET requests, receiving a short text in the querystring and return the predictions for the given text as a json.

4.2.3 Frontend

The user interface of this tool, like the one developed for the corpus creation, uses Vue.js, having as a main page the emotion detection one, presenting a text box for the input data and three buttons for the the type of classification the user wants to analyze. Similar to the corpus creation interface, Bootstrap[10], the open source tool-kit for frontends is used, along with Material design [29].

As it can be seen in Figure 4.4, where background image [41] was used, this is what the user interface looks like before the input analysis, while Figure 4.5 presents the probability of the text being positive or negative and the emotions that are characteristic for the given text.

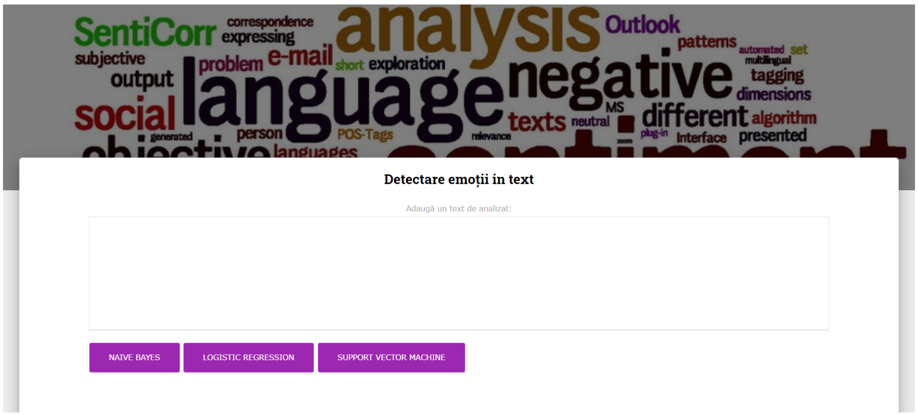

Figure 4.4: Main page before analyzing input text

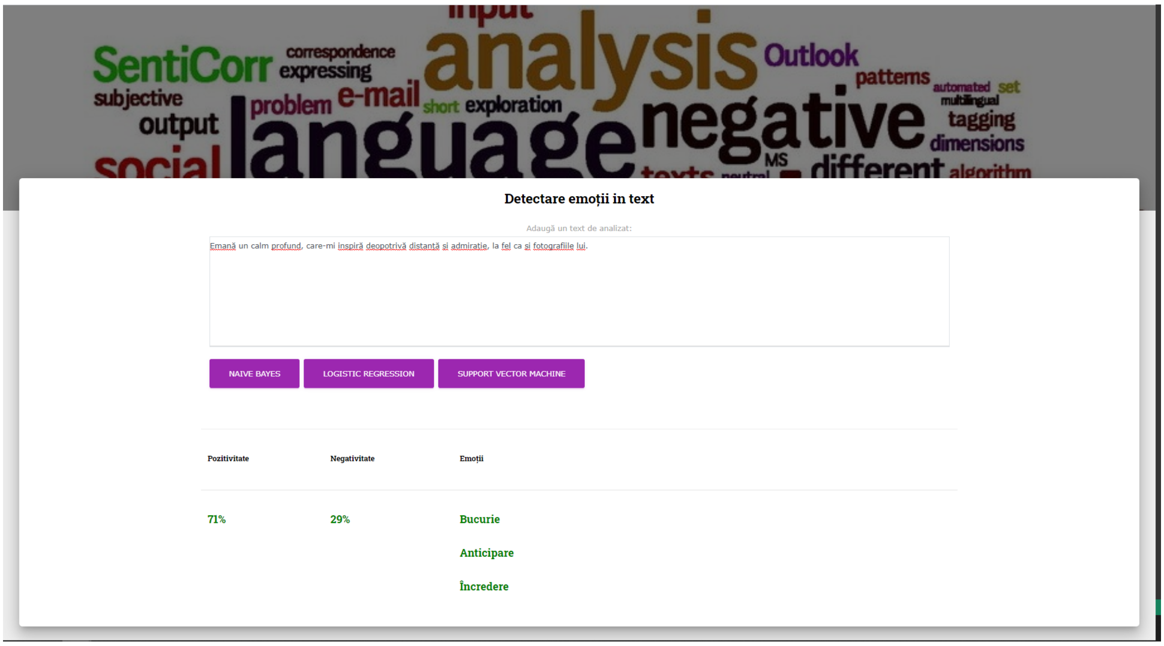

Figure 4.5: Main page with text emotion analysis

Chapter 5

Experiments and Results

The purpose of this chapter is to present the data distributions of the collected text annotations over the specific emotion labels, to compare the testing results of different algorithms on both binary and multi-label classification, with and without the addition of emotion specific features. Additionally, prediction results are shown, as displayed by the emotion detection user interface.

5.1 Dataset distribution

After the score computing stage, the dataset used by the classification algorithms was analyzed.

- Binary Dataset

  The binary dataset distribution is as represented in Figure 5.1, with 218 texts labeled as positive and 240 texts labeled as negative. As it can be seen, a relatively equal distribution of the two label classes was obtained. This dataset is later processed and used in the training and testing of the binary classifiers. This testing will be analyzed in this chapter, comparing the different classifiers accuracy and the results for them working with and without features (except for the vector equivalents of the textual data, which is considered a feature).

Category	Number of Texts
Positivity	218
Negativity	240

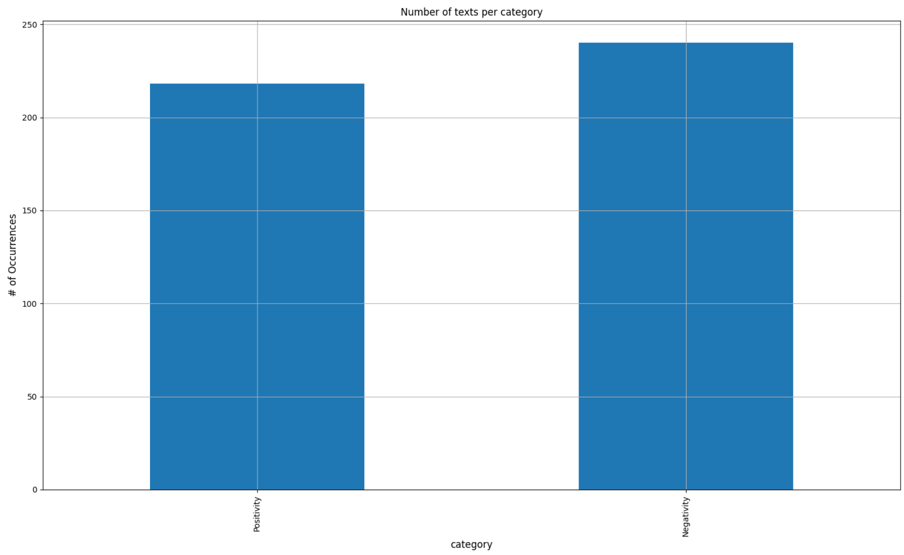

Figure 5.1: Binary dataset distribution

- Multi-Class Dataset

  A dataset containing the predominant emotions, meaning the emotion class with the maximum scores out of all the eight emotion classes has been created, for a possible multi-class approach, each sample text input has exactly one emotion label, as in Figure 5.2. It is clearly visible that the distribution is not equal at all, with sadness being the emotion that may be considered predominant in most sentences.

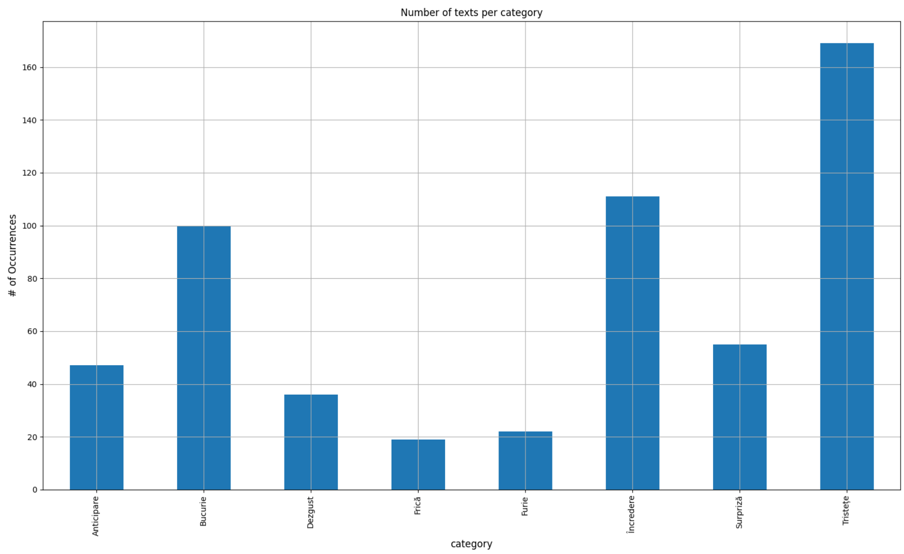

Figure 5.2: Multi-class dataset distribution

Category	Number of Texts
Anticipare	69
Bucurie	100
Dezgust	36
Frică	19
Furie	22
Încredere	111
Surpriză	55
Tristețe	169

- Multi-Label Dataset For the multi-label classification approach, all the emotion scores higher than zero were taken into consideration Figure 5.3. More than one emotion label was considered for each text, the final results clearly showing that anticipation appears in the most sentences, most probably as a result of the chosen computation of the scores, followed by joy and sadness.

Category	Number of Texts
Anticipare	424
Bucurie	319
Dezgust	215
Frică	139
Furie	114
Încredere	239
Surpriză	136
Tristețe	263

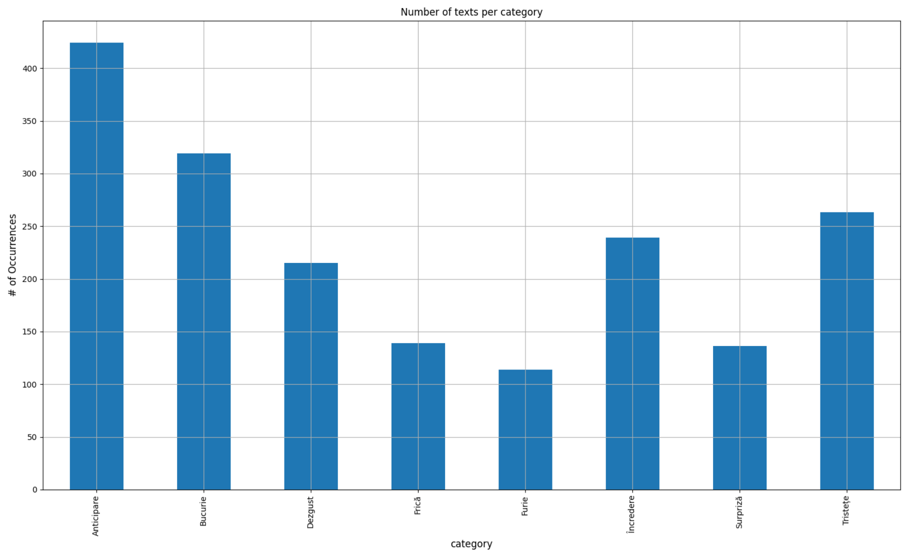

Figure 5.3: Multi-label dataset distribution

5.2 Classifiers Results

- Binary Classifiers

The first experiments on the binary classification were performed with only the Bag-of-Words with the Tf-Idf Transformer as features, with a Logistic Regression Classifier. After adding the emotion features, meaning the emotion words count, the negative marker processing, amplifiers and downtoners count, the algorithm was giving a higher performance on the precision, recall and f1-score reports, as it can be seen below.

Initial results:

	Precision	Recall	F1-Score
0	0.57	0.56	0.56
1	0.58	0.60	0.59
accuracy			0.58
macro avg	0.58	0.58	0.58
weighted avg	0.58	0.58	0.58

Final results:

	Precision	Recall	F1-Score
0	0.82	0.68	0.74
1	0.65	0.79	0.71
accuracy			0.73
macro avg	0.73	0.74	0.73
weighted avg	0.75	0.73	0.73

Finally, the accuracy of the other two types of classifiers, Naive Bayes and SVM, were computed after training with features, having received similar results. The SVM appears to have performed slightly better, but only with a difference of 1%.

Logistic Regression	Naive Bayes	SVM
0.73	0.73	0.74

- Multi-Label Classifiers

Following the approach of using the same features as for the binary classifiers, with a slight difference in the count of emotion words (the eight emotion labels were taken into consideration, not positivity and negativity), the OneVsRest Classifier was tested with the same three classifiers as for the binary approach for the multiple binary classifications the OneVsRest Classifier has divided the multi-label problem into. The following results were obtained:

	Logistic Regression	Naive Bayes	SVM
Anticipare	0.70	0.71	0.71
Bucurie	0.66	0.56	0.62
Dezgust	0.58	0.64	0.62
Frică	0.74	0.73	0.73
Furie	0.76	0.80	0.78
Încredere	0.53	0.55	0.53
Surpriză	0.70	0.70	0.70
Tristețe	0.63	0.56	0.63

As it can be seen, the anger class appears to have given the best results, but that may not be the reality, since as it was stated above, this class had the least amount of data samples for training and testing. A more balanced dataset should be able to give more accurate results.

5.3 New input analysis

For the prediction part, an example of a predicted new input text is seen in Figure 4.5. The sentence "Emană un calm profund, care-mi inspiră deopotrivă distanță și admirație, la fel ca și fotografiile lui."[9] is seen to be positive with a probability of 71%, meaning that the binary class it belong to is Positivity, which is corresponding with the meaning of the sentence. Then, the emotion labels that were predicted were Bucurie, Anticipare and Încredere. They appear in the descending order of their probabilities.

Another example is for the sentence "Cand o ușă se închide, o alta se deschide; dar deseori ne uităm atât de mult la ușa închisă că nu o mai vedem pe cea care s-a deschis pentru noi."(Hellen Adams Keller) [12]. The higher probability appears for Negativity, 88% and the predicted emotions are Anticipare, Frică, Tristețe and Dezgust. This sentence appears to be in contrast with the other example, appearing to perform rather good predictions, from a subjective point of view. Of course, some predictions are not as accurate and the algorithm does not perform as good on very short sentences, for example.

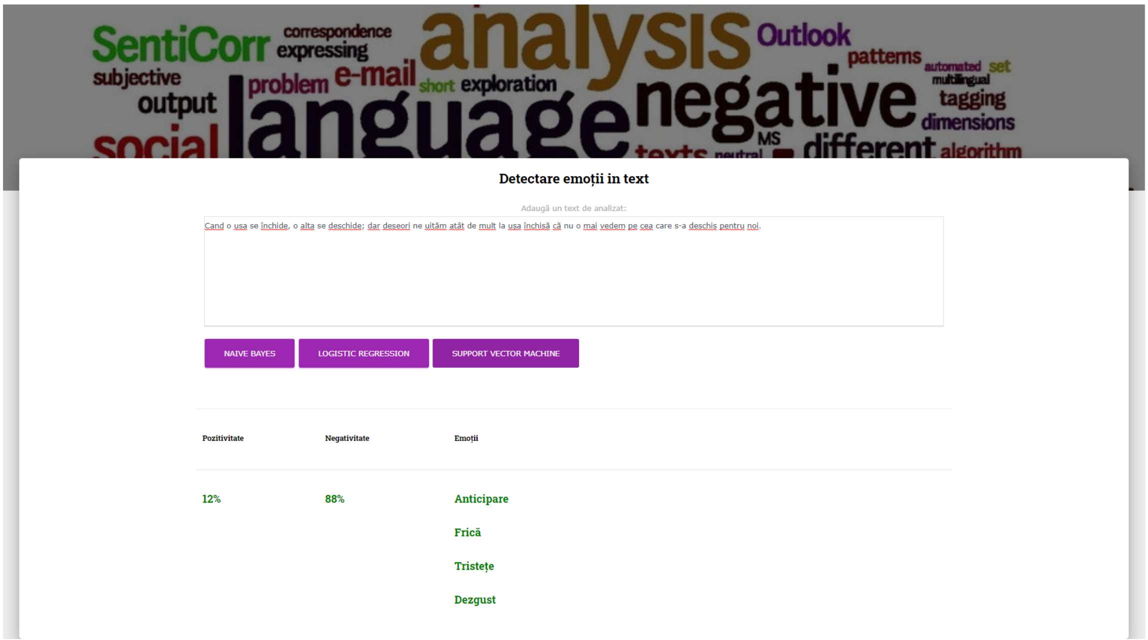

Figure 5.4: UI Text Input Results

Chapter 6

Conclusions and future work

Although many approaches and resources for the Natural Language Processing Task of Emotion Detection have been studied and developed for the English language, the Romanian language showed a lack in this direction.

The aim of this paper was to study the automatic text emotion detection, develop a foundation for a corpus containing Romanian texts annotated with emotion labels, creating a tool represented by a public website in this direction, making a comparative study of the performance of different supervised machine learning classifiers trained on the previously specified corpus and an emotion detection tool that predicts emotion classes of a Romanian text based on these classifiers, with a user-friendly web interface.

Although the basis of a solid application for emotion detection of Romanian texts has been developed, there is much room left for further improvements.

Starting with the created corpus, it must be noted that emotion annotations are highly subjective and, moreover, not analyzed by specialists of this domain. The developed tool for the creation of this corpus may be utilized by specialists in order to obtain a more accurate set of data.

The corpus creation website, although fulfilling its purpose of gathering text annotations, would benefit from development on the security side, taking into consideration that it is publicly available. Moreover, the management of its users could be better tracked using proper password administering.

The algorithms for both the binary and multi-label classification should be further evaluated in terms of the best use of hyperparameters and features. In terms of features, rather than a count of emotion words, weights computed based on the scope of amplifiers and downtoners surrounding these emotion words may perform better in terms of context. Moreover, the performance measurements may be better analysed in order to determine the best operating supervised machine learning models.

Regarding the emotion detection algorithms, once the developed corpus becomes more robust, several deep learning approaches, such as using a Bi-Long Short Term Memory models, which has been proven to outperform the supervised learning models should be taken into consideration. Even hybrid approaches, using a deep-learning model with linguistic features must be taken into account, making further use of the RoEmoLex lexicon.

As for the web application for emotion detection, the next step would also be deployment. In the context of an increasing use of the corpus creation tool and having a classification model with an increased performance, the following phase is the automation of the model training once a substantial number of new data is collected, thus optimizing its performance.

Bibliography

[1] Allouch, M., Azaria, A., Azoulay, R., Ben-Izchak, E., Zwilling, M., Zachor, DA., "Automatic detection of insulting sentences in conversation", *2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE)*, 2018, 1-4.

[2] Alm, C.O., Roth, D., Sproat, R., "Emotions from text: machine learning for text-based emotion prediction", *Proceedings of the conference on human language technology and empirical methods in natural language processing*, 2005, 579-586.

[3] Alswaidan, N., Menai, M.E.B., "A survey of state-of-the-art approaches for emotion recognition in text", *Knowledge and Information Systems*, 2020, 1-51.

[4] *AWS Documentation*, https://docs.aws.amazon.com/index.html.

[5] *Axios Documentation*, https://www.npmjs.com/package/axios.

[6] Batbaatar, E., Li, M., Ryu, K.H., "Semantic-Emotion Neural Network for Emotion Recognition From Text", *IEEE Access*, 7 (2019), 111866-111878.

[7] Bălan, O., Moise, G., Petrescu, L., Moldoveanu, A., Leordeanu, M., Moldoveanu, F., "Emotion Classification Based on Biophysical Signals and Machine Learning Techniques", *Symmetry*, 1 (2020).

[8] Bhavsar, H., Ganatra, A., "A comparative study of training algorithms for supervised machine learning", *International Journal of Soft Computing and Engineering (IJSCE)*, 4 (2012), 2231-2307.

[9] *Blogul lui Dragoș Bucurenci*, https://bucurenci.ro/tag/timp/

[10] *Bootstrap*, https://getbootstrap.com/.

[11] Canales, L., Martínez-Barco, P., “Emotion Detection from text: A Survey“, *Proceedings of the workshop on natural language processing in the 5th information systems research working days (JISIC)*, 2014, 37-43.

[12] *CitateCelebre.Net*, https://www.citatecelebre.net/

[13] *Collage Photo of Woman*, https://www.pexels.com/photo/collage-photo-of-woman-3812743/

[14] Dalgleish, T., Power, M. eds., “Handbook of cognition and emotion“, *John Wiley & Sons*, 2000.

[15] Desmet, B., Hoste, VR., “Emotion detection in suicide notes.“, *Expert Systems with Applications*, 16 (2013), 6351-6358.

[16] Ekman, P., “Basic emotions. Handbook of cognition and emotion“, *Wiley*, New York, 1999, 301-320.

[17] *Fastify Documentation*, https://www.fastify.io/docs/latest/.

[18] Feng, J., Xu, H., Mannor, S., Yan, S., “Robust logistic regression and classification“, *Advances in neural information processing systems*, 27 (2014), 253-261.

[19] *Flask Documentation*, https://flask.palletsprojects.com/en/2.0.x/.

[20] Gaind, B., Syal, V., Padgalwar, S., “Emotion detection and analysis on social media“, *arXiv preprint arXiv:1901.08458*, 2019.

[21] Gunes H, Pantic, M., “Automatic, dimensional and continuous emotion recognition“, *International Journal of Synthetic Emotions (IJSE)*, 1 (2010), 68-69.

[22] Hutchins, W.J., Dostert, L., Garvin, P., “The Georgetown-I.B.M. experiment“, *Locke and Booth*, 1955, 124-135.

[23] *JSON Schema*, https://json-schema.org/.

[24] Jurafsky, D., Martin, J. H., “Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition“, *Upper Saddle River, N.J: Prentice Hall*, 2000.

[25] Khosla, S., "EmotionX-AR: CNN-DCNN autoencoder based Emotion Classifier", *Proceedings of the sixth international workshop on natural language processing for social media*, 2018, 37-44.

[26] Lee, S.Y.M., Chen, Y., Huang, C.R., "A text-driven rule-based system for emotion cause detection", *Proceedings of the NAACLHLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, 2010, 45-53.

[27] Lupea, M., Briciu, A., "Studying emotions in Romanian words using Formal Concept Analysis", *Computer Speech & Language*, 57 (2019), 128-145.

[28] Ma, C., Prendinger, H., Ishizuka, M., "Emotion estimation and reasoning based on affective textual interaction", *International conference on affective computing and intelligent interaction*, 2005, 622-628.

[29] *Material Kit.*, https://demos.creative-tim.com/material-kit/index.html.

[30] Mohammad, S.M., Turney, P.D., "Crowdsourcing a word-emotion association lexicon", *Computational Intelligence*, 3 (2013) , 436-465.

[31] *MongoDB*, https://www.mongodb.com/.

[32] *Netlify*, https://www.netlify.com/.

[33] *NLP-Cube*, https://github.com/adobe/NLP-Cube.

[34] *Node.js*, https://nodejs.org/en/.

[35] *NumPy*, https://numpy.org/.

[36] *pandas - Python Data Analysis*, https://pandas.pydata.org/.

[37] Plutchik, R., Kellerman, H., "Theories of emotion", *Academic Press*, 2013.

[38] Picard, R. W., "Affective Computing", *The MIT Press*, 1997.

[39] *Python Documentation*, https://docs.python.org/3.8/.

[40] *scikit-learn Machine Learning in Python*, https://scikit-learn.org/stable/.

[41] Paul Houghton, *Sentiment Analysis 3 ways in Alteryx*, https://www.theinformationlab.co.uk/2018/12/21/sentiment-analysis-3-ways-in-alteryx/

[42] *VueJS*, https://vuejs.org/.

[43] *vue-google-auth - Handling Google sign-in and sign-out for Vue.js applications*, https://www.npmjs.com/package/vue-google-auth.

[44] Wang, S., Peng, G., Zheng, Z., Xu, Z., "Capturing Emotion Distribution for Multimedia Emotion Tagging", *IEEE Transactions on Affective Computing*, 2019, 1-11.

[45] Yang, H., Willis, A., De Roeck, A., Nuseibeh, B., "A Hybrid Model for Automatic Emotion Recognition in Suicide Notes", *Biomedical Informatics Insights*, 5 (2012), BII-S8948.