“BABEŞ-BOLYAI” UNIVERSITY

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
COMPUTER SCIENCE SPECIALIZATION

Diploma Thesis

A LEXICON-BASED APPROACH FOR SENTIMENT ANALYSIS OF ROMANIAN TEXTS

Supervisor

Lect. PhD. **Mihaiela Lupea**

Author

Ioana-Raluca Gabor

Cluj-Napoca

2020

UNIVERSITATEA "BABEŞ-BOLYAI"

FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ
SPECIALIZAREA INFORMATICĂ

Lucrare de diplomă

ANALIZA BAZATĂ PE LEXICON A SENTIMENTELOR EXTRASE DINTR-UN TEXT ÎN LIMBA ROMÂNĂ

Coordonator științific

Lect. Dr. **Mihaiela Lupea**

Autor

Ioana-Raluca Gabor

Cluj-Napoca

2020

Abstract

The exponential growth of the available online information provides computer scientists with many new challenges and opportunities. A recent trend is to analyze human sentiments, opinions and orientation about facts, brands, products, etc. This can be achieved by exploiting the available techniques that the domain of Sentiment Analysis has to offer.

In this paper, I propose a lexicon-based approach for sentiment classification of short pieces of text, written in Romanian. Many of such studies have been conducted for other languages, especially those belonging to the Germanic linguistic family, but quite few for Romanian. The novelty of this thesis implies the aggregation of three main approaches: a consistent study of the Romanian language specificity and characteristics in the context of Sentiment Analysis, the construction of a complex opinion lexicon for Romanian language, consisting of more than 4700 opinion words, 202 expressions and 97 modifiers, and the development of a tool for sentiment detection and classification of a wide range of Romanian texts.

This work is the result of my own activity. I have neither given nor received unauthorized assistance on this work.

Keywords: Sentiment Analysis, opinion lexicon, subjectivity, polarity, modifiers

Table of Contents

1. Introduction

The importance of expressing opinions has grown exponentially in the last few years, once with the development of online websites, social platforms and an increased interest in online purchases of various products. Thus, the activities conducted in the domains of Natural Language Processing and Sentiment Analysis (also known as Opinion Mining) have gained more popularity and received much more attention than ever before, such that text processing and opinion extraction became valuable instruments in obtaining a general view on the people's preferences.

1.1. Problem statement and motivation

In a modern, highly industrialized marketing-driven society, it comes as a necessity to analyze and process people's preferences regarding products, services or political views. The task to identify and classify an opinion extracted from textual information is nonetheless challenging, as the complexity and subjectivity of any language overburdens the elaboration of a perfectly accurate sentiment analysis solution.

In the context of a sentiment analysis performed on Romanian written documents, little work has been done in comparison to other languages, such as English. Thus, it may be of great significance to design and elaborate a solution, able to perform opinion mining activities, along with the development of a complex opinion lexicon, for Romanian language.

1.2. Purpose of this work

The main goal of the "A Lexicon-based Approach for Sentiment Analysis of Romanian Texts" bachelor thesis implies the achievement of three important objectives:

- A complex study of the Romanian language characteristics and behavior in the context of sentiment analysis applied on written texts;
- Elaboration of an opinion lexicon, which represents the fundamental phase in the development of the presented solution;
- Development of a sentiment analysis tool, able to identify, extract and classify the polarity of a written document;

Moreover, it is highly significant to mention that the novelty of the proposed solution lays in the elaboration of a Romanian opinion lexicon, enriched by a complex collection of expressions, modifiers and an additional domain-based lexicon, consisting of movie reviews related vocabulary and symbols.

1.3. Structure of the thesis

This paper has a complex structure and is consists of 6 main chapters, each one concerned with an important phase in the development of the presented solutions.

Chapter 2, entitled “Theoretical background”, outlines the most relevant and essential theoretical aspects, which represent the fundamental notions on which the presented application is constructed.

The elaboration of an opinion lexicon for Romanian language is thoroughly presented in Chapter 3, and each of the constituent sub-chapters is responsible for outlining important aspects regarding the lexicon construction.

Chapters 4 and 5 are concerned with presenting more technical aspects of the development of a tool in Python programming language, able to perform a sentiment analysis task on a wide range of Romanian texts. The most important ideas are organized on sub-chapters, each section presenting relevant information regarding tool implementation.

The last chapter, Chapter 6, outlines resulting conclusions and observations, regarding the elaboration of the proposed solution, along with obtained results and future improvement suggestions.

2. Theoretical background

This chapter aims to introduce fundamental notions from the domain of Sentiment Analysis and describe essential aspects of the linguistic domain, found in tight coupling with the implementation and consideration of the elaborated solutions.

2.1. Sentiment Analysis fundamentals

Textual information in the world can be broadly categorized in *facts* and *opinions*. A ***fact*** relies on an objective observation or reasoning with regards to entities, products or events, while an ***opinion*** represents a human subjective expression or feeling. Sentiment Analysis is a computational study of opinions, sentiments and emotions expressed through written text. Semantic orientation treats sentiment analysis as a text classification problem, and the classification criteria consists in the value of the inferred ***polarity*** [1]. In other words, the main goal is to establish if the analyzed text expresses a predominantly *positive, negative* or *neutral* opinion.

The sentiment analysis problem can be approached either in *supervised* or *unsupervised* manner. The supervised approaches rely on machine learning algorithms functionality to map the input to expected target, by trying to learn the best approximation of a function that accurately describes the polarity of the text at hand. The strength of this approach relies in the ability to learn from vast amounts of data and successfully acquire knowledge that would help classify new texts. On the other hand the unsupervised approach handles with sentiment detection by applying a rule based mechanism enhanced by an extended knowledge based with complex sentiment computation methods [2], [3]. The most unsupervised methods present in literature are ***lexicon-based***, ***dictionary-based*** or ***corpus-based*** algorithms , [4]. The solution proposed by this thesis belongs to the **unsupervised** algorithms category, and the two presented sentiment analysis methods rely on the complexity and correctness of the developed ***opinion lexicon*** (a collection of words which posses a positivity or negativity valence).

Another aspect worthy to be taken into consideration is the level at which one performs a sentiment analysis on a text. Mainly, sentiment analysis has been investigated at three levels:

- *Document-level* – The document level sentiment analysis classifies the entire document opinion as positive, negative or neutral;
- *Sentence-level (or phrase-level)* – The polarity is extracted by processing each phrase of a document. This is an approach often encountered when analyzing reviews and comments provided by customers or users;
- *Entity and aspect level* – Aspect level is the opinion mining and summarization based on feature; the classification is concerned with identifying and extracting features from the source data and it is mostly applied when the main interest is to examine the expressed sentiments about a desired aspect/feature in a review [5].

In the following sections of this thesis, the relevance of these notions, along with application examples, will be extensively presented and enhanced.

2.2. Preprocessing methods

Prior to applying any of the proposed solutions, it is imperiously necessary to preprocess the texts, in order to reduce them to a more approachable state. Generally, in the domain of Sentiment Analysis, it is common for the preprocessing techniques to impose numerous steps which need to be performed, before actually applying any sentiment extraction algorithm [2], [6]. Some of the preprocessing methods involve tasks such as:

- elimination of unimportant or disturbing elements, such as misspelled words or extra punctuation characters;
- substituting slang vocabulary with its formal correspondent terms, or treating an erroneous content, from the grammatical point of view;
- stemming (a process reducing all the derivative words to the common radix form);
- stopwords removal (stopwords include conjunctions, pronouns and articles);
- lowercasing
- lemmatization (reducing a term to its corresponding dictionary form).

The preprocessing techniques utilized in this application consist of solely applying lowercasing and lemmatization, since the removal of punctuation would cause inability to process emoticons or exclamation marks (whose importance is detailed in the sub-chapter 2.5) and the elimination of stopwords would affect the structure of utilized language patterns in the microphrase-based algorithm.

2.3. Related work

Sentiment Analysis and Natural Language Processing are domains that escalated in importance in the past few decades, and, as a result, more studies have been carried out and numerous papers have been published in this interval of time. Further on, the following paragraphs will describe and present some of the studies which were formerly conducted by Sentiment Analysis experts, and whose objects of study are relevant to this paper.

One unsupervised lexicon-based approach was presented by the authors [7] which focused on accomplishing two main tasks: contextual **polarity disambiguation** and message **polarity classification**. The sentiment extraction of the tweets was performed at phrase-level, and it involved negation detection and intensification treatment.

Other works involved a comparison of state-of-the-art resources for lexical-based sentiment analysis, such as the one presented in paper [8]. The main goal of the proposed study was examining the performance of SentiWordNet, WordNet-Affect, MPQA and SenticNet, the best accuracy being achieved by SentiWordNet and MPQA. The main approach presented in this

paper implied a micro-phrase extraction and processing in order to obtain the overall opinion polarity of the analyzed texts.

A very interesting and complex approach of sentiment analysis performed on movie reviews was presented by the authors of [9]. The main goal of the study was to perform a ranking of movies from IMDB database, classifying the reviews by characterizations ranging from "highly disliked" (or the equivalent mathematical score of 0), to "highly liked" (labeled with the maximum value, 4). The most challenging aspect in analyzing the selected movie reviews was treating the informal register vocabulary, which did not present a very clear structuring, correct spelling or grammatical rules. The method used to process the reviews in order to obtain a correct classification of the texts, consisted of the *n-gram* approach. The n-gram method implied selecting a continuous sequence of n items from a given sample of text or speech, and then analyzing each extracted sequence. The flow of the presented strategy implied a preprocessing activity, then parts of speech tagging, followed by a feature extraction phase, a feature reduction phase, and then, the final task performed implied sentiment classification and movie review ranking, according to the previously mentioned labeling from 0 to 4.

In this thesis, the structure of the proposed solution for Romanian language mirrored the ones presented in the previously related studies for foreign languages but it also presents elements of novelty due to personal contribution, such as development of a **Romanian opinion lexicon**, studying the Romanian language particularities and behavior and adapting the existing methods in order to function accurately for Romanian language.

2.4. Romanian language structure and particularities

Romanian is a part of the Eastern Romance sub-branch of Romance languages, a linguistic group that evolved from several dialects of Vulgar Latin. Romanian shares many characteristics with its' distant relatives, such as other Latin-originated languages (Italian, French or Spanish), but it also presents some particularities, such as verbal tenses, phrasal order of the words, the structure of active and passive diathesis, etc [15]. In the following subchapters, furthermore of such particularities will be presented, especially those details which are relevant to the application proposed by this thesis.

2.4.1. Diacritics

Probably one of the most noticeable characteristics of the Romanian language is the alphabet. The Romanian alphabet is a modified version of the classical Latin alphabet, consisting of thirty-one letters, five of which were altered from their Latin originals for the phonetic requirements of the language (Ă, Â, Î, Ș, and Ț, along with their corresponding lowercase versions). In casual written conversations, these modified letters are usually omitted, and substituted by their corresponding Latin characters. Despite the fact that, in

the absence of diacritics, the message is usually understood by interlocutors, in the context of a natural language processing, the presence of these special characters is indispensable [10]. Whether or not the diacritics are used can thoroughly change the final results, the precision of algorithms or the general interpretation of the analyzed text. This is caused by the functionality mechanism of the preprocessing methods, such as lemmatization, which reduces every word to its most basic form (dictionary form). A concrete example illustrating the significance of these diacritics could be the words "***urât***" and "***urat***". The first word is the translation of the English word "ugly" (which denotes a negative feature), while the second term denotes a very positive concept, which can be equivalent to "wished well to others" or even "blessed". It can be observed that a single sign has drastically shifted the entire meaning and polarity, thus substantially influencing the analysis of the phrase. Additional details regarding these preprocessing concepts and their purpose will be extensively described and explained in the succeeding chapters.

2.4.2. Negation

In linguistics and grammar, ***affirmation*** and ***negation*** are the ways by which grammar encodes negative and positive polarity in phrases, clauses or other utterances. In the majority of the cases, affirmation expresses a true statement or validity of a basic assertion, whereas a negative form denotes falsity. The grammatical category associated with affirmative and negative meaning is defined as ***polarity***.

Negation represents a universal category, inherent to human reasoning. Every language presents specific syntactical structures which are used to express negation. In Romanian language there are numerous ways to form negation, and treating this aspect in a natural language processing algorithm is unquestionably challenging, due to negation being achievable through different negation words, complex expressions or ambiguous rephrases [11].

Most Romance languages, and also the ones belonging to the Slavic group, negate the sentence by means of a negative marker (designated by Linguistics experts as "***morpheme***") in pre-verbal position, and Romanian language is no exception to this rule. In Romanian finite sentences, negation takes the form of a free morpheme *nu* ("not"), which obligatorily precedes the finite verb, and in some cases, few words may intervene between the verb and the negative marker [11]. These words are denoted as *clitic-like* elements (auxiliaries, pronominal clitics, adverbial intensifiers such as *mai – "*more, anymore*"*, or *prea – "*very, too", etc.), which in Romanian unavoidably precede the verb, or other parts of speech, but follow negation (Examples: "nu prea frumoasă" – "not very beautiful", "nu mai vreau" – "I don't want (it) anymore", or, simply, "nu pot" – "I can't")

Negative polarity items designate weak or disadvantageous features. In order to shift the meaning of an adjective, adverb, verb or noun, some negative contexts can also be marked by the negation affixes such as "*ne-"* (*nedorit* – "not wanted", *nedreptate* –"unjustice"), *"dez-"/"de-"* (*dezamăgi* –"to disappoint", *destabilizat* – "unstable", *dezgust* – "disgust")

,*"in-"* (*incorect* – "incorrect", *inuman* – "unhuman") or *"a-"* (*anormal* – "abnormal", *acalmie* – "the state of not being calm").

Another aspect worthy to be taken in consideration is blind negation, which expresses a negative feeling, but in a more subtle manner. Blind negation is usually omitted in natural language processing, but it may provide substantial information regarding the true polarity of the analyzed sentence, generally occurring in feedback reviews or surveys. For instance, in the context of movie reviews, the phrase *"Calitatea efectelor speciale trebuie îmbunătățită"*, meaning "The quality of the special effects must be improved", suggests that a certain feature, such as the quality of image processing, is not sufficiently satisfying, thus it needs to be changed, and it is equivalent to simply negating it, by stating "the quality is not good enough". Additional information regarding negation will be presented in the following chapters.

2.4.3. Language patterns

One of the possible approaches of determining the polarity of a phrase is to analyze the constituent subphrases, concentrating on identifying and processing their ***language pattern***.

A *language pattern* represents a sequence of parts of speech which can be distinguished in numerous phrases or documents written in that specific language. For example, one of the most frequently encountered patterns in Romanian is ***'NOUN, ADJ'*** (noun followed immediately by an adjective, as opposed to English language, where the adjective always precedes the determined noun), or, similarly, ***'VERB, ADV'*** (verb followed by adverb).

Another important aspect, worthy of being taken into consideration, is that various forms of negation can be extracted by simply analyzing the language pattern of a phrase (for instance the sequence ***'PART, 'VERB, ADV'*** can often indicate the specific negation form obtained from the terms ***'nu'*** and ***'deloc'***, which are placed before and after the verb the negation refers to).

Additionally, when extracting a microphrase pattern from a phrase, ***conjunction*** (part of speech which connects words, phrases, or clauses, equivalent to the word "and") may play a significant role, as it usually connects two identical morphological units which posses the same value of positivity or negativity. For example, in the case of the phrase "*un oraș frumos și prosper*" (meaning "a beautiful and prosperous city"), the conjunction "*și"* connects two positive adjectives, "*frumos"* ("beautiful") and "*prosper*" ("prosperous"), and the noun "*oraș*" is determined by both of the two adjectives.

Therefore, in a specific pattern such as previously mentioned, ***'NOUN, ADJ'***, one must also verify whether the last adjective is in a conjunction relationship with another adjective, which also determines the initial noun and, as a result, the final pattern will be, as follows, ***'NOUN, ADJ, CCONJ,ADJ'***. The importance of this adjective conjunction will be furtherly detailed in the chapter concerned with the presentation of the microphrase-based

algorithm, and more information regarding extraction and classification of these patterns will be provided as well.

2.5. Alternative forms of sentiment expression

The algorithms proposed by this thesis are applied on opinionated Romanian texts, among which 127 are movie reviews. These reviews were collected from a public source and they present many characteristics specific to the online communication style. When inferring the polarity of a text, one must be aware that, sometimes, it is not words which voice the opinion best, but punctuation and use of emoticons or emojis.

Emoticons are textual portrayals of a writer's emotional attitude or facial expressions in the form of icons. Initially, emoticons were created solely from ASCII characters, but once with the advancement of technology, they transcended from static to dynamic emoticons, currently known as emojis. In the last decade, these graphical icons have considerably increased in popularity, often joining the traditional text-based messages, offering a more personal and vivid perspective of the phrases. There exists a wide range of variety of these emoticons, but in the context of Sentiment Analysis, the most relevant are the ones which express a positive feeling (such as :), :D, ;), :P, etc.) or a negative one (such as :(, :/, ;[, etc.).

Punctuation also plays an important role in analyzing the sentiments from a given text, but not as significant as emoticons do. In particular cases, in expressive pieces of text, such as in poetry, punctuation is found in tight coupling with the message transmitted by the writer, highlighting the emotional meaning of the phrases. For example, three consecutive periods ("…") may indicate a state of melancholy, exclamative mark ("!"), may suggest excitement or, on the contrary, pain, anger, fury, while the question mark ("?") may express doubt, uncertainty or even confusion.

Without any doubt, there are many other additional means of expressing sentiments and feelings in the online environment, such as short videos, gifs, memes, but in this thesis, the principal focus will be on extracting, classifying and analyzing words, linguistic structures, punctuation and emoticons.

3. Development of an opinion lexicon for Romanian language

As stated in the previous introductory paragraphs, the development of the lexicon represents a key-phase in the elaboration of the presented work. Typically, lexicon-based approaches for sentiment classification are based on the insight that the polarity of a written piece of text can be computed based solely on a specially designed opinion lexicon. Undoubtedly, this simplistic approach is prone to failure because the complexity of the natural language, along with the presence of expressions, punctuation, abbreviations or negations, makes it extremely challenging to determine the real polarity expressed by the phrases. Further on, the presented subchapters will detail each important step in the development of an opinion lexicon for Romanian language.

3.1. Words and sentiments

In Romanian language, and also in other languages, each linguistic item, whether it is a word, an expression, or a syntactical structure, may carry a certain emotional significance (except, of course, the objective words which denote factual aspects, or terms belonging to a specialized vocabulary, such as in Science, Law, History, Mathematics and so on). Having this aspect in view, it is imperative to mention that the analysis of sentiments extracted from a text (document-level) is thus reduced to analysis of a single unit (word or expression). In Romanian, one could identify numerous words (or phrases) which express subjectivity in a certain way. The majority of these words are adjectives as part of speech. The semantic role of an adjective is to define a feature, trait, characteristic of its referent, usually a noun. Usually, in Romance languages, the adjectives are varying in form and sound according to the gender and plurality of the referenced noun. For instance, the word "beautiful" can present the following forms: *frumos* (for masculine noun gender), *frumoasă* (for feminine noun gender), while *frumoși*, *frumoase* denote the quality of beauty for plural nouns, or "they" pronoun. Adjectives can be divided in two separate categories, by the polarity it expresses: positive adjectives and negative adjectives. For example, positivity could be extracted (interpreted) from the following adjectives: *bun* ("good"), *harnic* ("hardworking"), *divin* ("divine"), *iubitor* ("loving/careful"), *drăguț* ("nice/pretty"), *curajos* ("brave"), *maiestuos* ("majestic"). On the other hand, negative meaning can reside in words such as: *rău* ("bad"), *laș* ("coward"), *diabolic* ("evil"), *necinstit* ("dishonest"), *prost* ("stupid/dumb").

Secondly, the next largest group of words which may have emotional significance is mostly consisting of nouns and verbs. Linguistically, the morphological purpose of nouns is to denote and name objects, beings, states, activities etc. In the context of sentiment expression through written words, nouns bring a significant contribution, as one may perceive positive meaning in nouns such as *bunătate* ("kindness"), *dreptate* ("justice"), *iubire* ("love"), *libertate* ("freedom"), *entuziasm* ("excitement"), and negative meaning in words similar to *răutate* ("evilness"), *avariție* ("avarice"), *ură* ("hate"), *moarte* ("death"). In the case of verbs, their functional role is to convey an action or a state of being. The positive polarity may be extracted

from verbs which express an action of possessing a positive feeling or performing a positive activity, among which the following are enumerated: *a iubi* ("to love"), *a respecta* ("to respect"), *a spera* ("to hope"), *a învinge* ("to win"), *a zâmbi* ("to smile"). As for the negative verbs, some examples are the the following: *a pierde* ("to lose"), *a eșua* ("to fail"), *a răni* ("to harm"), *a distruge* ("to destroy"), *a ucide* ("to kill"), *a detesta* ("to hate").

Probably the least significant from the numerical point of view is the set of sentiments determined by analyzing adverbs. Adverbs represent those parts of speech whose linguistic function is to characterize a verb, adjective, determiner, clause, preposition, or sentence. In Romanian, adverbs may also be replaced by phrasal structures, which are their correspondent in meaning (this is also encountered in the case of nouns, adjectives or verbs, but, predominantly, it represents a distinctive mark in the case of adverbs). Adverbs such as *bine* ("well"), *rău* ("badly"), *cu greu* ("hardly"), *abia* ("barely"), *clar* ("clearly"), *exagerat* ("exaggeratedly"), *politicos* ("politely"), *de-a dreptul* ("really, very") are just few adverbial structures from which emotions can be sensed, and those sentiments can either emphasize a certain quality of an action, or provide a new positive or negative meaning (further information about this situation will be laboriously presented in the consecutive subchapter). As a supplementary observation, some of the adjectives can also function morphologically as adverbs. In the previous example, this is the situation of the words *rău*, *clar*, *exagerat* and *politicos*. This equivalence in meaning and form, but not in linguistic role, is interesting to analyze in the case of providing positivity scores, when, even if it is analyzed the same word, the numerical scores might result in very different values. This particular case, and many others related, will be thoroughly described in the following sections.

3.2. Intensifiers and downtoners

When deciding to mathematically quantify the polarity of a written text at the phrase-level, it is of great significance to not disregard the presence and purpose of ***intensifiers***, and ***downtoners***. An intensifier is a vocabulary unit belonging to the linguistic terminology, which denotes a ***modifier***. A modifier does not offer any substantial contribution to the propositional meaning, but it rather serves as an enhancer, which provides additional emotional context to the word it modifies. Strictly speaking, if a term expresses a positive or negative meaning, then the presence of such an intensifier will have as effect the increasing of the term's polarity, thus resulting into a "more positive", or "more negative" overall meaning of the phrase. For example, the word *frumos* in the Romanian language meaning "beautiful", possessing a certain positive score, increases the intensity by, for instance, 50% when an intensifier (such as *foarte*) is applied on the word ("foarte frumos").

On the other hand, a downtoner is a modifier, usually in the form of a degree adverb, which decreases the effect of a modified item. When applied on a word or expression which denotes a positive or negative meaning, the obtained result is a "less positive" or "less negative" one. For exemplifying, suppose the word *frumos* is determined by a downtoner such as *puțin* ("puțin

frumos"). In this case, the intensity of the word drops by 25%, resulting into a term with reduced positivity.

Having in view the impact that these modifiers may have in a sentiment analysis process, the construction of the lexicon would not have been sufficient without considering including them in the lexicon documents. Thus, an important constituent phase of the development of the Romanian opinion lexicon was creating a complex collection of intensifiers and downtoners, along with individually determined scores (percentages), representing "how much" do these modifiers influence the polarity of the terms they refer to. In the fifth subchapter, entitled 'Lexicon development iterations', a sample of both intensifiers and downtoners will be presented, along with additional numerous technical details regarding the methodology applied in order to build all the information incorporated in this lexicon.

3.3. Treating expressions

In both formal and informal vocabulary of any language, there might exist several specific textual units, also known as expressions, whose purpose is to either denote an action, feeling, state, either to rephrase an already existing term. It is very difficult to collect all the currently used expressions in a language, due to the fact that human language is constantly changing, new expressions may frequently appear (some in slang language as well), while others are stopped being used or no longer appropriate.

In the context of developing an opinion lexicon for Romanian language, treating expressions in a written text is a phase which should be, by all means, considered. The main reason for this is that, from the syntactical point of view, expressions can replace other parts of speech, such as adjectives or nouns which may express a positive or negative concept or feature, and, moreover, expressions may as well present a certain polarity in meaning. In a sentiment analysis of a text, omitting expressions might lead to an unrealistic and faulty results, and it is important to mention that even though the expression can consist of objective terms (which do not express a certain polarity), it is possible that the final composed structure may indeed provide marks of affection.

The lexicon constructed and designed for the presented application consists in an elaborate collection of files and tables, each concerning a certain category of words, among which some are positive and negative expressions, belonging to the general Romanian vocabulary. In the following paragraphs, there will be presented a small-sized sample of the aggregated expressions, along with the corresponding polarity, meaning, translation, and other additional relevant information.

- *bătaie de joc* – "mockery", is an expression with a negative polarity; morphologically, it behaves as a noun, and it is equivalent in meaning with the term which combines all the component words*, batjocură;*

- *a face de oaie* – "to fail"; this is a negative expression, belonging to the informal register; it usually occurs in slang vocabulary, in online texts and short natural conversations; the expression *da greș* is a synonym structure, identical in meaning and polarity; the *a face de oaie* expression is a classical case when a negative action is expressed by objective, affectionless terms: *a face* ("to do"), *de* (preposition), *oaie* ("sheep");
- *de bun augur* – "favorable"; this expression clearly transmits a positive connotation, it usually belongs to the formal register, but it is also encountered in informal vocabulary;
- *a muri de râs* – "to laugh hard"; the particularity of this expression is that, despite its' positivity, it is formed by both positive and negative words: the verb *a muri* ("to die") and the noun *râs* ("laughter"); it is interesting that, depending on the context, or whether a word is part of a lexical structure, such as an expression or a saying, it can shift its' polarity, and denote an entirely different notion; as an observation, in a naïve approach of a sentiment analysis of a text, resuming to analyze word by word polarity, these special situations are completely ignored, thus resulting into faulty interpretation of the polarity at a document-level;
- *(de) nota zece*; this a frequently encountered example of a phrase whose meaning cannot be determined unless it is present in a specific context; thus, the variations of this expression (*notă zece, de notă zece, de nota zece plus*) have to be carefully analyzed, according to the context. The word *zece* denotes the value ten (10), and on a scale from one to ten, it obviously represents the maximum value, which indicates that this expression is commonly used to suggest superiority, greatness (for example, *el este un om de nota zece* – "he is an extraordinary/superior/educated man", and it definitely suggests a positive meaning); considering a grading situation at school, this expression may simply infer an objective aspect, or even a negative case, when the maximum value is larger than ten;

The expressions studied previously represent an insignificant part of the constructed lexicon, and supplementary information will be delivered in the following subchapter, as well in subchapter 3.6, in which quantifications of the polarity, expressed by numerical scores, are provided for these textual items.

3.4. Special vocabulary

In the context of sentiment analysis of a text belonging to a particular category of written documents, one should consider including in the development of a lexicon a set of specific terms, whose purpose is relevant exclusively to that category. The solution proposed by this thesis analyzes a large sample of Romanian written texts, among which more than one hundred are movie reviews. As expected, the dataset of movie reviews collected for this application contains several linguistic particularities, such as the usage of slang vocabulary, abbreviations, emoticons (as previously presented in the subchapter 2.5) and some terms which belong to the domain of movie reviews.

One of the most often encountered situations is usage of slang specific adjectives such as *'ok', 'cool', 'fain', 'top'*, '*blockbuster'*, which are influenced by English language, and some of them can be still be used in other category of texts written in an informal register. Other lexical

particularities observed in these reviews is the usage of abbreviations (‘*bvo*’ is sometimes used instead of the original word ‘*bravo*’) and few forms of rating (‘*nota zece’* being used to express a positive assessment of the movie, while ‘*nota 2* or ‘*nota 3*’ suggest a poor quality of the movie, or a disappointing plot or cast). It is true, however, that these words do not significantly impact the final results of the analysis, but in some situations, the usage of special vocabulary can be decisive in the analysis of seemingly neutral texts, thus contributing to a better accuracy.

3.5. Usage of RoWordNet API

RoWordNet is a complex knowledge-base lexicon, storing various information about words, designed to process information related to Romanian language vocabulary, mirroring Princeton WordNet [12]. In the context of the presented application, the benefits obtained by utilizing this lexicon are numerous, as RoWordNet is able to provide valuable lexical data, such as *definition* of a word, the corresponding part of speech (*pos*), a set of synonyms for a specific word (*synset*) and *sentiwn*, a three-valued list indicating the scores of SentiWN PNO (Positive, Negative, Objective) corresponding to a *synset*. The usage of RoWordNet API is depicted below, in the following Python language code samples.

```
import rowordnet as rwn

wn = rwn.RoWordNet()
word = 'bine'
synset_ids = wn.synsets(literal=word)
for synset_id in synset_ids:
    synset_object = wn(synset_id)
    print('Definition: ', synset_object.definition)
    print('Part of Speech: ', synset_object.pos)
    print('Literals: ', synset_object.literals)
    print('SentiWN: ', synset_object.sentiwn)
```

different *synsets*, each *synset* consists of a set of *literals* and it is associated to a list of scores and a definition, providing the context of the *synset*. The resulting parts of speech are represented as a single character, and the values of the polarity scores are real numbers, ranging from 0 to 1, and the sum of all the values of a *sentiwn* corresponding to a *synset* is always 1. Several concrete results are presented below:

```
Definition:  în condiții financiare confortabile
Part of Speech:  r
Literals:  ['bine']
SentiWN:  [0.125, 0.25, 0.625]
```

```
Definition:  Starea de a fi mulțumit, fericit, sănătos, prosper.
Part of Speech:  n
Literals:  ['bunăstare', 'prosperitate', 'bine']
SentiWN:  [0.75, 0.0, 0.25]
```

3.6. Lexicon development

The main focus of this subchapter is to provide relevant information regarding the process of the development of a Romanian opinion lexicon. The process of creation and improvement of the lexicon represented a complex and meticulous activity, performed in 9 different iterations, which are presented in detail in the following paragraphs. The process of creating a lexicon is detailed in [14] where the author describes standard steps in the creation of a lexicon in any language.

Iteration 1 - The initial stage of development. In the earliest phase of the development, the construction of the lexicon commenced by translating a list of more than 2000 opinion English words into Romanian, placing each translated term into a file corresponding to its' polarity (positive, negative) and part of speech (nouns, adjectives, verbs, adverbs), thus resulting into a collection of eight lexicon files. The first major step required to manually verify and improve the collection of files, analyzing line by line to correct inadequate translations or spelling errors, and verify that the words in each file are in the appropriate lemmatized form (dictionary form). The final requirement which needed to be accomplished was setting the diacritics for all the words, in order to be able to perform the succeeding operations.

Iteration 2 - Revising and improving the initial lexicon. During the analysis and processing of the selected dataset, multiple words were still omitted and thus, the situation imposed an additional revision of the words, and correction of possible erroneous content. At this stage of the development, two additional files were created, one for the positive expressions, and the other one for negative expressions, each operation performed manually. Another lexicon improvement was creating a small list of special vocabulary for movie reviews, including abbreviations and specific slang words, belonging to the informal register.

Iteration 3 - First automatic refinement of the lexicon. This step involved processing the obtained lexicon files and concatenation of all the words obtained into a Python dictionary object. After the last modifications regarding content were finally performed, all the data was filtered, all the duplicates removed and the data set was then sorted alphabetically. The results were collected in a Python dictionary object, resembling the initial files structured based on polarity and parts of speech.

Iteration 4 - Usage of RoWordNet API – obtaining score lists. The previously obtained lexicon was suitable enough to be utilized in a counting-based sentiment analysis approach, but for the microphrase-based algorithm, which consists of evaluating various kinds of written documents in Romanian, further improvement needed to be performed. This is the main reason for which the RoWordNet API was used, and the most important feature of RoWordNet is to provide polarity scores for various words, accompanied by a large set of synonyms. In this iteration, all the eight previously constructed separate lists of words were concatenated in one

single file, and then for each word, a search in RoWordNet was performed to obtain the corresponding scores (positive, negative and objective).

Iteration 5 - RoWordNet automatic enhancement and score computation. At this point, the list of words which represented the first version of the opinion Romanian lexicon, was not large enough to fulfill all the requirements of the microphrase-based algorithm, thus it was imperiously necessary to provide more items (the total number of words was 2837, consisting of both positive and negative terms). In this phase, the file which contained all the generated words was passed through and for each word entry, several steps were performed:

1. A polarity quantification was obtained based on the average values of positive, negative and objective list of scores, in the following manner:

```
function compute_score_for_word (positive_scores, negative_scores, objective_scores)
    avg_positive ← compute_avg(positive_scores)
    avg_negative ← compute_avg(negative_scores)
    avg_objective ← compute_avg(objective_scores)

    if avg_objective = MAX_VALUE_OBJECTIVE then
        compute_score_for_word ← 0
    else
        score ← avg_positive - avg_negative
        compute_score_for_word ← score
    endif
endfunction
```

   In this case, all the words which received a zero score, were considered to be neutral and then inserted in a separate file (where all the objective words were stored), the positive words obtained a score higher than 0, while the negative ones received a score lower than 0.

2. If the search word was not found in RoWordNet (the case for some words, such as: '*draconic'*,' *excelat'*, or '*uzurpator'* , for which the resulting score lists were empty), the word was saved in a separate file (where were stored all the words for which RoWordNet did not provide scores).
3. The list containing all the synonyms (synset) of the search word was analyzed and processed. Every synonym which appeared was saved in a separate file (where all the synonyms of all the sought words were stored). Among the terms which RoWordNet provided, some expressions were generated as well. The expressions were extracted from

the synsets and saved in an file whose purpose was to store all the detected expressions, along with the computed score (calculated based on the algorithm previously presented), for that specific expression.

4. After all of the words from the initial lexicon were processed, the same operations were again performed for the obtained list of synonyms, excepting the step concerning the extraction of new additional synonyms (this specific step was omitted to prevent an exponential growth of the lexicon, but it can be repeated as many times as necessary, in order to obtain a very complex lexicon; the currently obtained lexicon suffices in complexity and variety of words, but undoubtedly, it can be furtherly improved).

Iteration 6 - Revising RoWordNet objective words and unidentified words. Even though the RoWordNet provided scores for the majority of the words in the dataset, an additional revision was unquestionably required, since the words which did not have a generated list of scores were still relevant to the final version of the lexicon. Thus, each word which occurred in this situation was manually checked and provided a subjectivity score, considering the expressed meaning and the general interpretation of the term. Moreover, the neutral words were verified once more, since it may be possible for RoWordNet to provide faulty mathematical values. Thus, some of the objective words were provided scores, in a similar manner as in the case of unidentified words.

Iteration 7 - Collecting and refinement of all the results. After all the previous steps had been accomplished, the resulting words, along with the corresponding mathematical score values and parts of speech, were collected from all the files and concatenated in a single large list. The obtained list was furtherly refined, and the duplicate pairs of word and part of speech were eliminated. The final version of the Romanian opinion lexicon consists of a total of 4780 words, and a small sample, of the opinion lexicon is presented in Table 1. Additionally, relevant statistics based on the obtained results are depicted in Table 2.

Table 1 – Lexicon results sample

Word	Part of Speech	Score
abandon	NOUN	-0.125
abia	ADV	-0.083
binefăcător	ADJ	0.5
binefăcător	NOUN	0.525
blândețe	NOUN	0.562
calomnia	VERB	-0.362
daună	NOUN	-0.25
defect	NOUN	-0.161
defect	ADJ	-0.125
imoral	ADJ	-0.716

Table 2 – Lexicon results statistics

	ADJECTIVES	NOUNS	VERBS	ADVERBS
Total number:	1985	1790	755	250
Positive words	674	601	247	110
Negative words	1311	1189	508	140
Total number of words	**4780**			
Total number of positive words	1632			
Total number of negative words	3148			

Iteration 8 - Processing expressions. Starting from this stage of the development, all the succeeding operations were performed as a process of enrichment of the Romanian opinion lexicon. This specific iteration was exclusively concerned with providing a substantial collection of expressions and frequently encountered phrases, along with positivity and negativity scores. In the preceding phases, two distinct expression files were constructed. The first one was manually created in the initial development phase, while the other one was automatically generated from the results RoWordNet provided. In this iteration, the main objective was concerned with aggregating all the expressions obtained in one single source file. The expression statistics results (Table 3), accompanied by a concrete sample of data (Table 4) are depicted below.

Table 3 - Expressions statistics

Expressions statistics	
Total number of expressions	202
Total number of positive expressions	62
Total number of negative expressions	140

Table 4 - Expressions sample data

Expressions	Score
atac de panică	-0.125
bun de nimic	-0.75
da peste cap	-0.103
pur și simplu	0.13
poftă de viață	0.45
lipsă de griji	0.25
ajunge cuțitul la os	-0.33
ajunge la sapă de lemn	-0.55
apă de ploaie	-0.156

Iteration 9 - Constructing lists of intensifiers and downtoners. This final stage of the lexicon development involved an activity which was independent of all the previous presented phases. The main goal represented the creation of lists of modifiers and providing subunitary numerical values for each modifier, percentages representing the amount by which the score of the determined word increases or decreases (inspired by the ones [4] associated to English modifiers). In order to facilitate the further operations performed in the microphrase-based algorithm, two categories of modifiers were elaborated, *pre-modifiers* (the intensifier or downtoner belonging to this category usually precedes the determined word or expression) and *post-modifiers* (modifiers which are often located after the determined word or expression). The final set of modifiers consists of 21 post-modifiers and 76 pre-modifiers (the difference of number of items is not surprising, due to the particularity of the Romanian language to place all determiners before the determined item in a phrasal structure). In the tables below, Table 5 and Table 6, there are presented few examples of the resulting modifiers.

Table 5 – Post-modifiers

Modifier	Score	Type
cu totul	0.5	intensifier
pe deplin	0.75	intensifier
foc	0.5	intensifier
cu sfințenie	0.5	intensifier
de tot	0.5	intensifier
slab	-0.5	downtoner
drastic	0.75	intensifier
radical	0.5	intensifier

Table 6-Pre-modifiers

Modifier	Score	Type
ce	0.25	intensifier
cu adevărat	0.5	intensifier
atât de	0.75	intensifier
așa de	0.75	intensifier
neobișnuit de	0.75	intensifier
cu totul și cu totul	0.75	intensifier
puțin	-0.25	downtoner
cât de cât	-0.25	downtoner

4. Application - Sentiment Analysis in the Romanian Language

The main objective of this chapter is to present both practical and theoretical important aspects regarding the elaborated solution proposed by this thesis. As mentioned in the previous introductory paragraphs, the problem of sentiment extraction from Romanian written texts is approached in an unsupervised lexicon-based manner, providing two different methods (counting-based and microphrase-based algorithms) in order to achieve the desired results. The following subchapters will meticulously present each method proposed, along with concrete examples and the obtained results.

4.1. NLPCube text preprocessing methods

In order to obtain the lemmatized structure of the phrase, the text was processed using NLP Cube. NLPCube is an end-to-end Natural Language Processing opensource framework [13], which can perform splitting, tokenization, compound word expansion, lemmatization, tagging and parsing for multilingual text entries. The usage of NLP Cube is depicted in the snapshot below.

```
from cube.api import Cube

cube = Cube(verbose=True)
cube.load("ro")  # select the desired language

text = "Bucuresti este capitala tării noastre."
sentences = cube(text)
for sentence in sentences:
    for entry in sentence:
        print("Word:" + entry.word + " ;Lemma:" + entry.lemma +
              " ;Part of Speech:" + entry.upos)
```

In the context of the presented application are the parts of speech, and the lemmatized forms. As depicted in the snapshot below, the obtained results are provided in the form of simple strings, and the punctuation is also detected, as it can be noticed in the displayed results. As an additional observation, the accuracy of the provided results is strongly affected by the absence of diacritics (in case of Romanian language processing), thus it is undoubtedly necessary to verify the correctness of spelling for the phrase, before passing it as an input.

```
Word:București ;Lemma:București ;Part of Speech:PROPN
Word:este ;Lemma:fi ;Part of Speech:AUX
Word:capitala ;Lemma:capitală ;Part of Speech:NOUN
Word:țării ;Lemma:țară ;Part of Speech:NOUN
Word:noastre ;Lemma:meu ;Part of Speech:DET
Word:. ;Lemma:. ;Part of Speech:PUNCT
```

4.2. Internal representation of a text

The two proposed solutions, counting-based and microphrase-based algorithms, are applied on two different datasets. The counting-based approach processes texts in form of short movie reviews, while the second method focuses on sentiment extraction from a general Romanian written text. The two categories of selected documents for these algorithms present distinct features, and this is the main reason for which two separate internal data representations (classes) were elaborated, with the intention to hold data of the analyzed texts. In other words, the selected movie reviews have a different internal representation of the data, compared to the other, more general, texts. In the succeeding paragraphs, a meticulously presentation of the two text structures will be provided, along with an explicit, clear-cut graphical representation of those structures.

In the case of the selected movie reviews, each text had a previously established polarity and the afferent content. The reason for creating a data representation for movie reviews was to store the corresponding data before applying the sentiment analysis algorithm and the data resulting from the counting-based approach, in order to have a general view of the accuracy of the solution. In Figure 1 and Table 7 below, a more detailed representation for this text structure is provided.

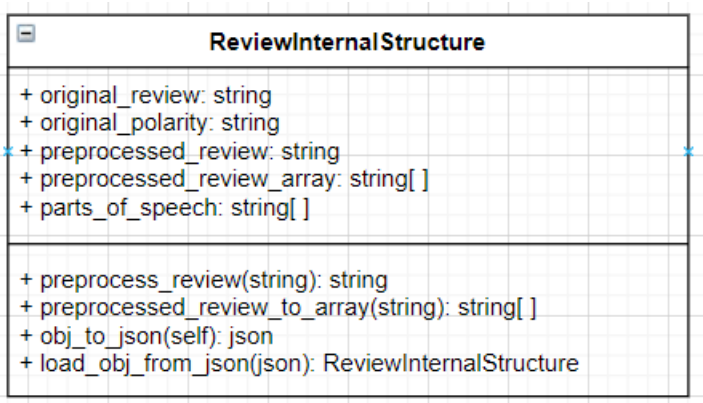


Figure 1 - Review internal representation

Table 7 - Review class details

Class field/method	Purpose/Description
original_review	Stores the initial, unaltered form of the review
original_polarity	The pre-determined polarity of the review
preprocessed_review	The text resulting from the preprocess algorithm
preprocessed_review_array	A tokenized representation of the preprocessed review
parts_of_speech	The afferent part_of_speech for every lemmatized item
preprocess_review(string)	Returns the corresponding preprocessed form of the review
preprocessed_review_to_array(string)	Coverts the obtained preprocessed text to an array of tokens
obj_to_json(self)	Coverts an ReviewInternalStructure object to json format
load_obj_from_json(json)	Updates fields according to values from json format

On the other hand, the representation corresponding to the general texts presents a slightly different structure. Since these texts represent the data on which the microphrase-based algorithm is applied on, they are in tight coupling with the information stored in microphrase objects. Thus, the internal structure of the microphrases will also be provided below, along with all the details associated to the general Romanian texts (see Figure 3, Figure 2, Table 8) .

Figure 2 - Text Internal Representation

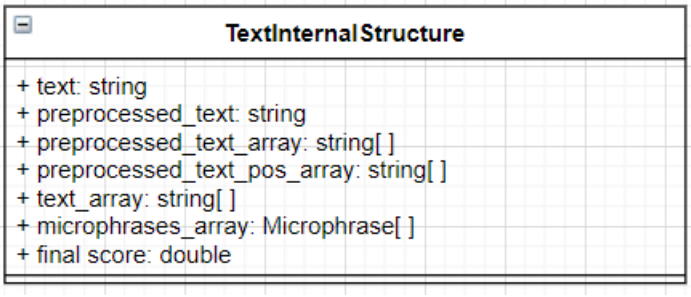


Figure 3 - Microphrase

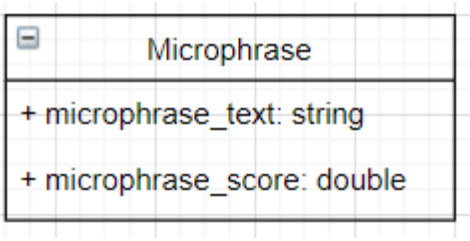


Table 8 - Text class details

Class field/method	Purpose/Description
text	Stores the initial, unaltered form of the text
text_array	The tokenized representation of the original text
preprocessed_text	The text resulting from the preprocess algorithm
preprocessed_text_array	A tokenized representation of the preprocessed text
preprocessed_text_pos_array	The afferent part of speech for every lemmatized item
microphrases_array	The list of all identified microphrases
final score	A double value representing the polarity score of the text

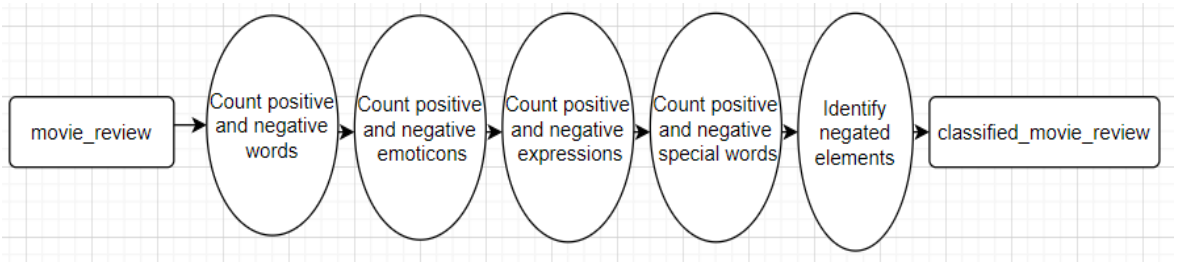
4.3. The Counting-based algortihm

The first solution algorithm proposed by this thesis is a sentiment detection approach based on counting the positive and negative lexicon extracted elements of a movie review. The functionality of the algorithm presents a pipeline structure, and it is considered to be a naïve approach, since it cannot provide very accurate results, should the analyzed phrases present complex syntactical structures. In order to improve the quality of the generated results, expressions and negation lexical forms have also been examined and included in the counting-based approach. The following two subchapters will provide additional details regarding this lexicon-based counting algorithm.

4.3.1. Description

As a general overview of the counting-based method, the sequence of steps approached in this naïve algorithm pipeline are depicted in Table 9 below, presenting the main tasks performed by this algorithm.

Table 9 - Counting-based pipeline



All the counting tasks (counting positive and negative words, expressions, emoticons), are very similar regarding the sequence of instructions. The fundamental idea of these intermediate steps is to perform a search for every movie review token (word, expression or emoticon) and to append it to the corresponding list of elements (positive and negative elements). In Figure 4, a pseudocode representation of the word counting algorithm is presented in a more detailed manner.

```
function count_words (review_preprocessed_array,review_pos_array,lexicon,positives,negatives)
        l ← review_preprocessed_array.length()
        for index ← 1,l do
                lemma ← review_preprocessed_array[index]
                part_of_speech ← review_pos_array[index]
                is_positive ← search_in_lexicon (lemma,part_of_speech,'positive')
                is_negative ← search_in_lexicon (lemma,part_of_speech,'negative')
                if is_positive is True then
                        positives ← positives.add(lemma)
                else
                    if is_negative is True then
                        negatives ← negatives.add(lemma)
                    endif
                endif
        endfor
        count_words ← positives, negatives
endfunction
```

Figure 4 - Count polarity words algorithm

Treating negation is an extremely important aspect to consider when improving the counting-based algorithm. Undoubtfully, this specific task must represent the final step in the pipeline algorithm, on the grounds that in order to detect the negated terms correctly, the lists of both positive and negative items should be complete.

The **negation detection algorithm** implies utilizing several other sub-algorithms, each responsible with performing a different search for negative structures. More concretely, the presence of negation can be identified if any of the following situations occur:

- The review contains "non words", meaning that a word can be negated if it is part of a compound term, having *non* as a prefix. For instance, the words *non-profit* or *non-violent*, both negate a term with a positive or negative meaning, thus resulting into a term with opposite polarity valence.
- A positive or negative word is preceded by the term *fără* (the equivalent for *without* in English). As a concrete example, if the word *sens* (meaning "reason" or "good judgement") is preceded by word *fără*, it results into an entirely different meaning ("illogical" or "unreasonable"), thus it must not be counted as a positive word.
- A positive or negative word is negated by the particle *nu*, which always precedes the negated term (the "distance" between the *nu* particle and the negated word may vary, as pronouns or determiners may interfere between verbs and negation words, for example and thus the accuracy of this algorithm is not perfect).

As previously presented, the negation detection algorithm is unquestionably challenging and complex, hence in the pseudocode representation below, the main important tasks are briefly sketched, solely to emphasize the general approach of this algorithm.

```
function negation_detection (review_obj, positives, negatives)
        non_words ← find_non_words(review_obj)
        for non_word in non_words do
                word ← non_word.eliminate_non_particle
                if word is positive then
                    negatives.add(non_word)
                else if word is negative then
                     positives.add(non_word)
                endif
        endfor


        for word in positives do

                is_without_word ← check_if_without_word(word, review_obj)
                is_negated ← check_if_negated(word, review_obj)
                if is_without_word is True or is_negated is True then
                        positives.remove(word)
                        negatives.add(word)
                endif
        endfor
        for word in negatives do

                is_without_word ← check_if_without_word(word, review_obj)
                is_negated ← check_if_negated(word, review_obj)
                if is_without_word is True or is_negated is True then
                        negatives.remove(word)
                        positives.add(word)
                endif
        endfor
        negation_detection ← positives, negatives
endfunction
```

In order to reach a conclusion and to obtain a final polarity for the review, two tasks must be performed: firstly, all positive and negative items must be separately accumulated and then, if the number of found positive items exceeds the number of identified negative items, the content of the review is classified as being a positive one (and the same approach is applied in the case of a predominantly negative sentiment expression of a review). Secondly, the computed polarity is then compared to the original polarity of the review to verify the accuracy of the counting algorithm.

4.3.2. Examples

The paragraphs below present in detail the functionality of this algorithm applied on two different movie reviews, along with additional explanations.

Review1: "Ice Age 2 este o animație reușită, și cu subiect fain. Chiar dacă nu mi-a plăcut coloana sonoră, în ansamblu a fost ok ;)"

Identified positive words: ['reușit', 'fain', 'ok']

Number of identified positive words: 3

Identified negative words: ['nu place']

Number of identified negative words: 1

Identified positive emoticons: [';)']

Number of identified positive emoticons: 1

Identified negative emoticons: [ ]

Number of identified negative emoticons: 0

Final score: **positive** (nr positive elements = 4 > nr of negative elements = 1)

Review2: **"**Un film îngrozitor ! Personajul principal este lispsit de farmec, iar subtitrările sunt greșite :("

Identified positive words: [ ]

Number of identified positive words: 0

Identified negative words: ['îngrozitor', 'lispsit de farmec', 'greșit']

Number of identified negative words: 3

Identified positive emoticons: [ ]

Number of identified positive emoticons: 0

Identified negative emoticons: [':(']

Number of identified negative emoticons: 1

Final score: **negative** (nr positive elements = 0 < nr of negative elements = 3)

4.3.3. Results

As mentioned previously, the accuracy of the counting-based lexicon approach is found in tight coupling with the similarity between original reviews polarity and algorithmically computed polarity. The algorithm was tested on a total of 127 Romanian movie reviews, and the obtained results were collected and processed in a form of a confusion matrix.

In the context of a statistical classification, a confusion matrix, also known as an error matrix, represents a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm and facilitates identification of confusion between classes. Most performance measures are computed from the confusion matrix. As far as the counting-based algorithm is concerned, the confusion matrix structure is based on the one depicted in Figure 5, and the obtained statistics results are presented in Table 10.

		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

Figure 5 - Confusion matrix structure

Table 10 - Counting-based algorithm results

		COMPUTED	
		Negative	Positive
ACTUAL	Negative	22	29
	Positive	6	70
Total number of movie reviews		127	
Precison		0.707071	
Recall		0.921053	
F1 Score		0.8	
Accuracy		0.724	

As an observation, it is important to specify that the number of the positive reviews (76) was larger compared to the negative ones (51), and the accuracy of sentiment extraction in the case of a negative review was strongly influenced by the precision of negation detection algorithm. The rate of **accuracy** is characterized by the ratio between the number of

correctly classified reviews and the total number of reviews, while the **recall** is defined as the ratio between the number of correctly classified positive reviews divided by the total number of positive reviews. In order to obtain the **precision** of the algorithm, the ratio between the total number of correctly classified positive reviews, divided by the total number of computed positive reviews. The **F1 Score** (or F-measure) represents a harmonic mean computation based on the values previously obtained for precision and recall results, and its purpose is to characterize a test's accuracy, providing a more realistic measure of the test's performance. The mathematical formulas used to compute the precision, recall, accuracy and F1 score are depicted below.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

4.4. The Microphrase-based algorithm

In the context of sentiment analysis at the document-level, one plausible approach would be to first perform a sentiment extraction at phrase-level, and then analyze all the obtained results in order to establish the general polarity of the examined document. Relying on the reasoning provided by this technique, the microphrase-based algorithm is designed in a similar manner, dealing with more complex language structures and patterns. In the succeeding subchapters, more details regarding the structuring of this algorithm will be thoroughly presented.

4.4.1. Description

The microphrase-based approach is structured as a pipeline algorithm, and it presents several similar features, as well as various additional improvements, compared to the counting-based method. The main difference is that this algorithm is tested on a more extensive range of Romanian texts, an this is the principal reason for which it relies on identifying, extraction and classifying of specific language patterns. Moreover, another particularity of this approach is the usage of intensifiers and downtoners, in order to provide a more nuanced variety of degrees of polarity (neutral, very positive, slightly negative etc.). As opposed to the previously presented naïve approach, the microphrase-based algorithm

represents a more challenging and laborious task, since it requires a careful analysis and detection of language patterns and phrases, which in natural language are extremely flexible and vary in meaning depending on the context. Figure 6 schematically presents the general approach of this algorithm.

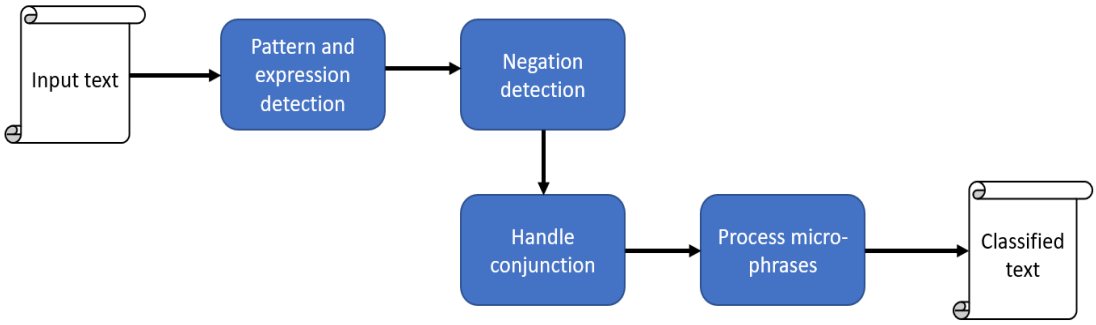


Figure 6 - Microphrase-based pipeline

The principal purpose of the microphrase-based algorithm is to extract, analyze and process the list of all the microphrases from a piece of text, which posses a polarity valence (in other words, the selected text fragments are relevant for sentiment analysis of the document). The approach consists of performing four important steps, each one presented in detail below:

- ***Expressions and patterns detection***. This specific phase involves searching for Romanian expressions or patterns in a text;
    - The sought expressions are the ones obtained during the development of the Romanian opinion lexicon, and the search involves looking for the actual textual form of the expressions;
    - In the case of patterns, specific sequences of parts of speech tags are sought in the list of the text parts of speech, corresponding to all of the words of the analyzed text. For example, the sought patterns of parts of speech may be elementary forms, specific for the Romanian language, such as *NOUN,ADJ* or *VERB,ADV*, or parts of speech corresponding to the collection of modifiers (for instance *'ADJ', 'ADP'* – specific for modifiers such as *extraordinar de*, *ciudat de* or *'DET', 'ADV'* – specific for modifiers such as *mult mai*);
    - An important aspect, worthy of mentioning is that the modifier itself, without a determined item is of no polarity value; thus, after locating an identifier, the immediate step is to include the determined term as well in the microphrase; (as mentioned in the previous sections, there are two types

of modifiers: *pre-* and *post-* modifiers; depending on the type, the determined word will be adequately located in the phrase).

Once a pattern or an expression is located, the text is then split by two in: text before the identified expression or pattern, and the text fragment residing after the found expression or pattern. This approach of text splitting will facilitate the succeeding operations; For simplicity of explanations of the following steps, the notations *pre-text* and *post-text* will be used.

- ***Negation detection***. Usually negation precedes the term it refers to, thus the search for negation particles (*nu, lipsit de, fără de*) is performed in the corresponding *pre-text*;
- ***Treating conjunction***. If an expression is identified, then the search for conjunction is performed in the corresponding *post-text*. If there is an identified conjunction term (the Romanian word "*și*"), then the phrasal element located on the right of the identified term will be part of the final microphrase.
- ***Polarity score computation***. Once all the elements of a microphrase were identified (conjunction term, negation, modifiers along with the determined item), the microphrase score needs to be computed. A sketched algorithmic description of the score computation algorithm is depicted below, including the most important steps performed.

```
function microphrase_compute_score(microphrase)
    microphrase_score ← microphrase.expression or
    microphrase_score ← apply_modifer_on_term(microphrase.modifier,
                                              microphrase.term)
    is_negation ← check_if_negation(microphrase)
    is_conjunction ← check_if_conjunction(microphrase)
    if is_conjunction is True then
        microphrase_score ← microphrase_score +
                            get_score(microphrase_conj_term)
    endif

    if is_negation is True then
        // shift polarity
        microphrase_score ← microphrase_score * (-1)
    endif
    microphrase_compute_score ← microphrase_score
endfunction
```

As opposed to the counting-based algorithm, this approach proves to be more challenging and complex. It is also important to mention that the final score of the text is computed based on all the previously obtained microphrases scores, using a normalized formula.

In the normalized formulation (see Equation 1), the microphrase-level scores are normalized by using the length of a single microphrase, in order to weigh differently the microphrases according to their length.

$$pol_{norm}(m_i) = \sum_{j=1}^{k} \frac{score(t_j)}{|m_i|}$$

Equation 1 - normalization formula

The final score obtained in this manner defines entirely the polarity of the analyzed text. Thus, if the value of the score is below 0, the text is classified as negative, while if above 0, it suggests a positive polarity.

4.4.2. Results

Since the microphrase-based algorithm can be applied on a general kind of text, the obtained score results represent an interpretation of the text content. The excerpt below depicts the behavior of this approach on a piece of text in Romanian, along with the identified microphrases and their scores, and a classification of the text

Text1: "Multe clădiri din România sunt lăsate la voia întâmplării și dărăpănate. Am trăit o mare dezamăgire când am vizitat Herculane."

Microphrases:

1. microphrase_text: "la voia întâmplării și dărăpănate";
   microphrase_score: -0.5
2. microphrase_text: "mare dezamăgire"
   microphrase_score: -1.125

$final\ score = \frac{\sum_{k=1}^{n} \text{microphrases(k).score}}{n}$, n – number of microphrases

Final score: (-0.5 – 1.125) / 2 = -0.8125

Classification: Strongly negative text

Text2: "El i-a oferit o mâna de ajutor și sprijinul lui nu a fost în zadar."

Microphrases:

1. microphrase_text: "o mâna de ajutor";
   microphrase_score: 0.432
2. microphrase_text: "nu în zadar"
   microphrase_score: -0.612 * (-1) = 0.612

$final\ score = \frac{\sum_{k=1}^{n} \text{microphrases(k).score}}{n}$, n – number of microphrases

Final score: (0.432 + 0.612) / 2 = 0.522

Classification: Strongly positive text

As an additional observation, according to the final score value, one can classify the content of the text by degrees of polarity or negativity. Since the range of the final scores is the interval [-1,1], a more specific classification can be performed according to the following subintervals:

- [-0.2, 0.2] – relatively *neutral* text content
- [-0.5, -0.2) – *slightly negative* polarity
- (0.2, 0.5] – *slightly positive* polarity
- [-1, -0.5) – *strongly negative* polarity
- (0.5, 1] – *strongly positive* polarity

5. Implementation details

The purpose of this chapter is to outline the more technical details regarding the application architectural structure, utilization and presentation.

5.1. Application architecture

The application is constructed based on a modular structure, each module fulfilling a different task. In the Figure 7 below, the Conceptual Diagram of the application is depicted.

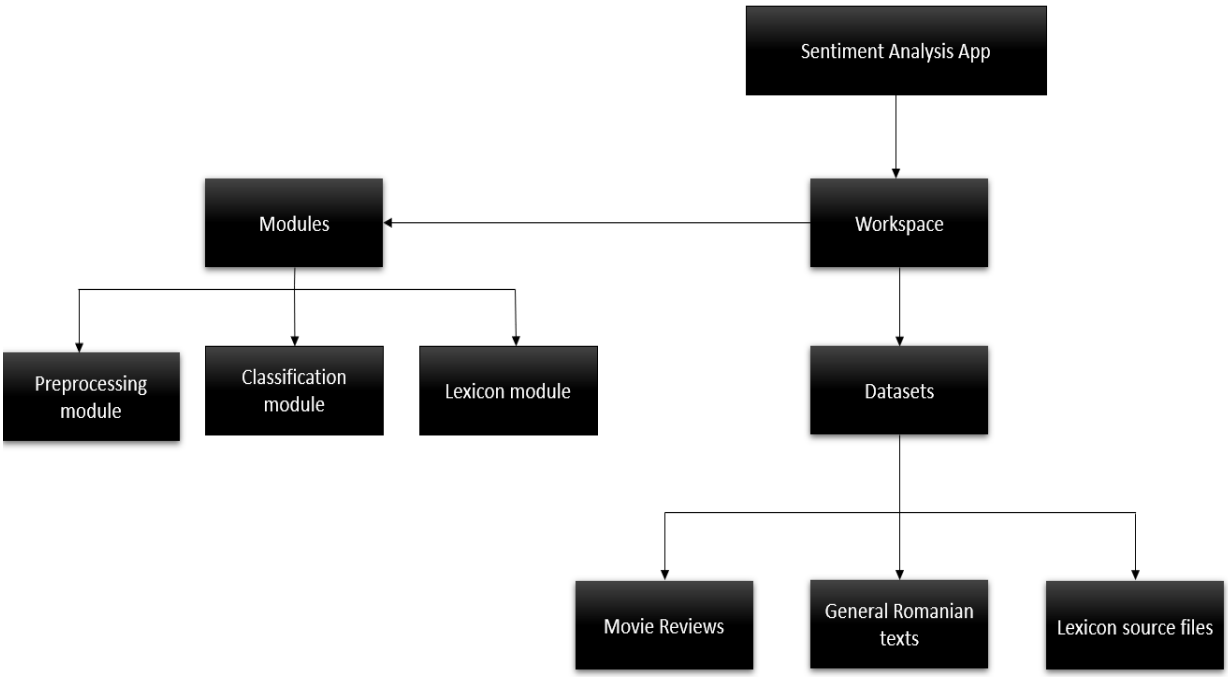


Figure 7 - Conceptual Diagram

Each application functionality is encapsulated in a package. Some of the packages are subdivided in other packages, responsible with performing a more specific task. The Figure 8 presents the general package architectural design of the application and in the following pages, each of the component package is explicitly described and analyzed.

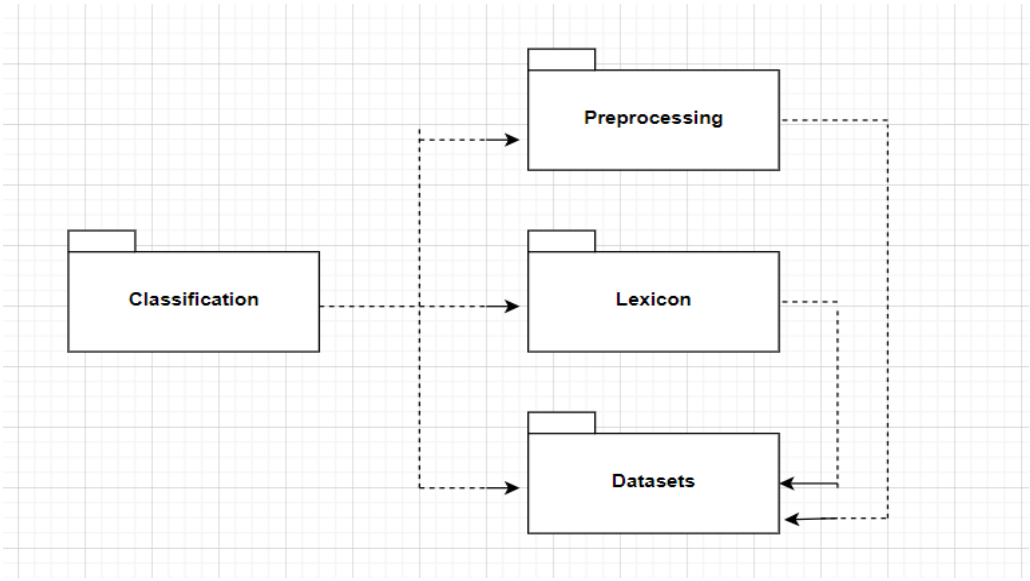
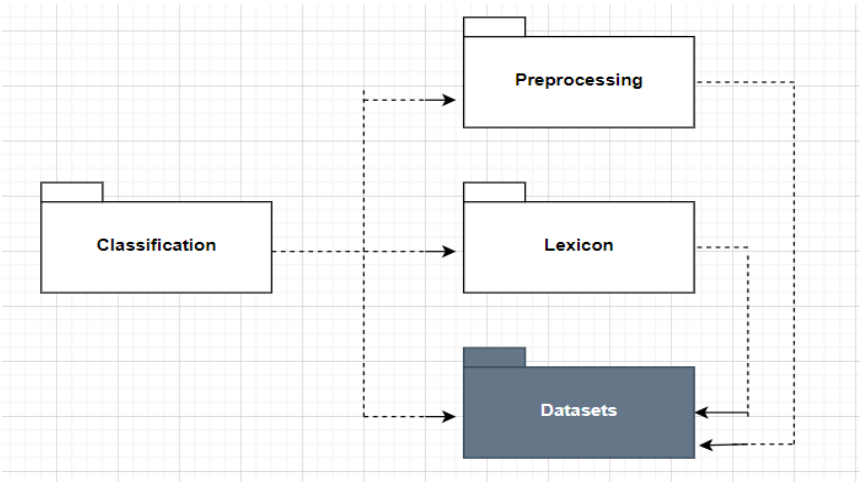
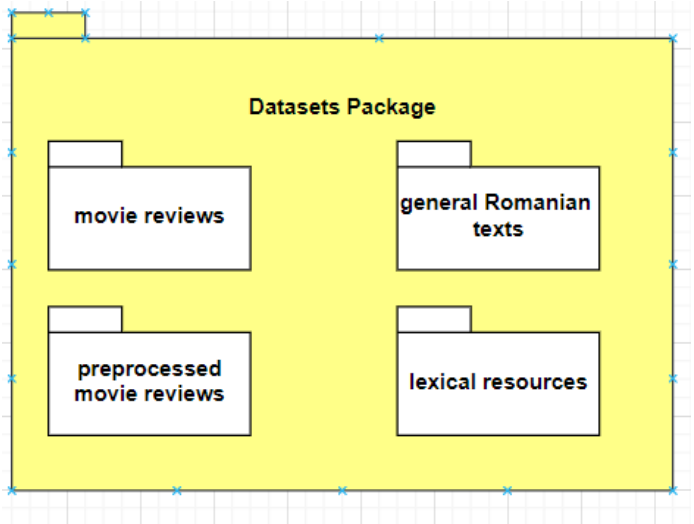


Figure 8 - Package diagram

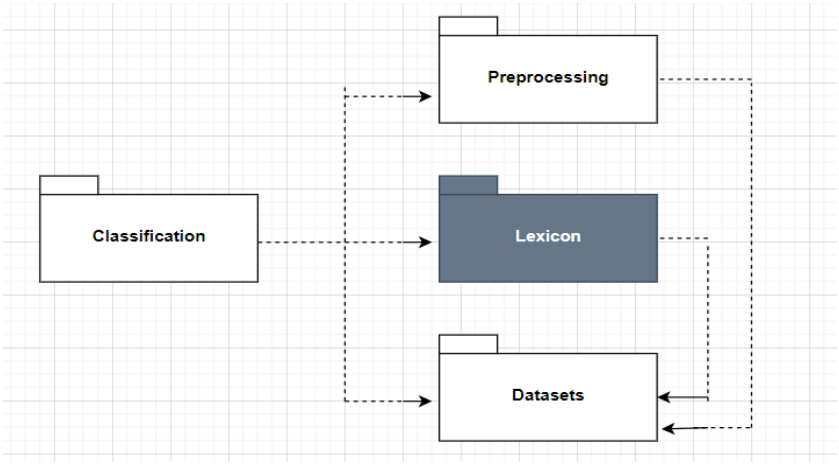
1. *Datasets Package*

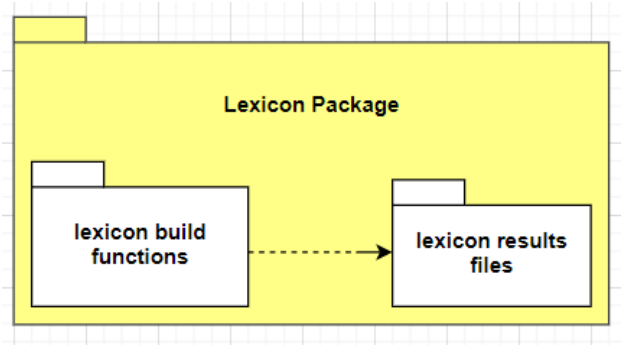




The Datasets Package includes four other sub-packages, each one storing a different kind of textual data: the movie reviews original texts, the reviews in a preprocessed form (provided by the preprocessing module), the lexical resources, containing all the necessary files in order to construct the lexicon, and a sub-package containing a collection of various Romanian texts.

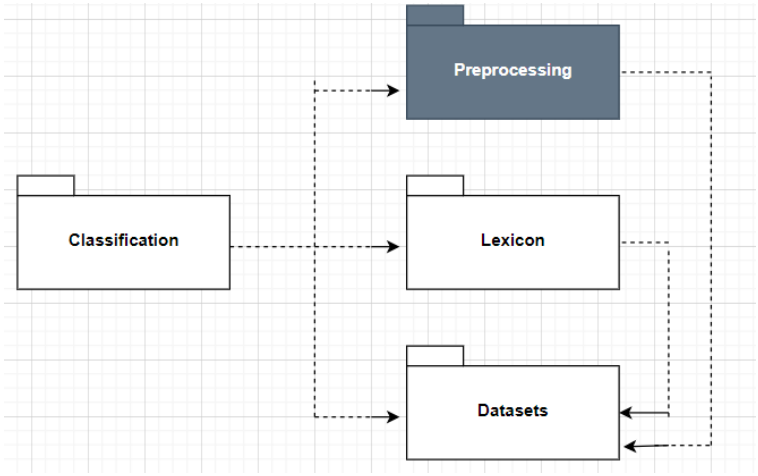
2. *Lexicon Package*

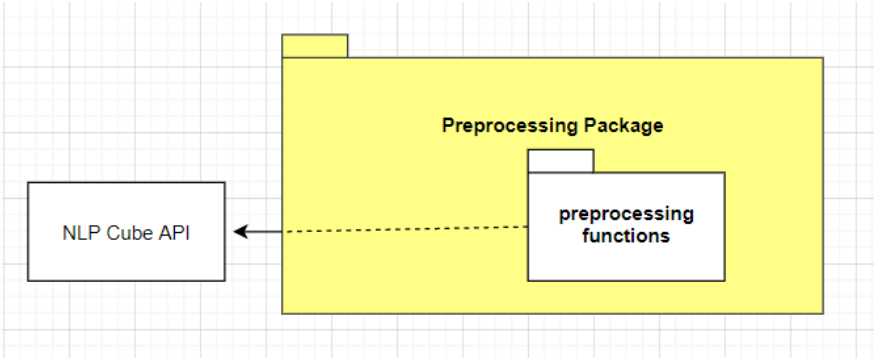




The lexicon package includes functionality sub-package, along with lexicon results sub-package. There is a dependency relation between the lexicon build functions sub-package and the lexical results files, as the lexicon functionality is found in tight coupling with the resulting collection of files.

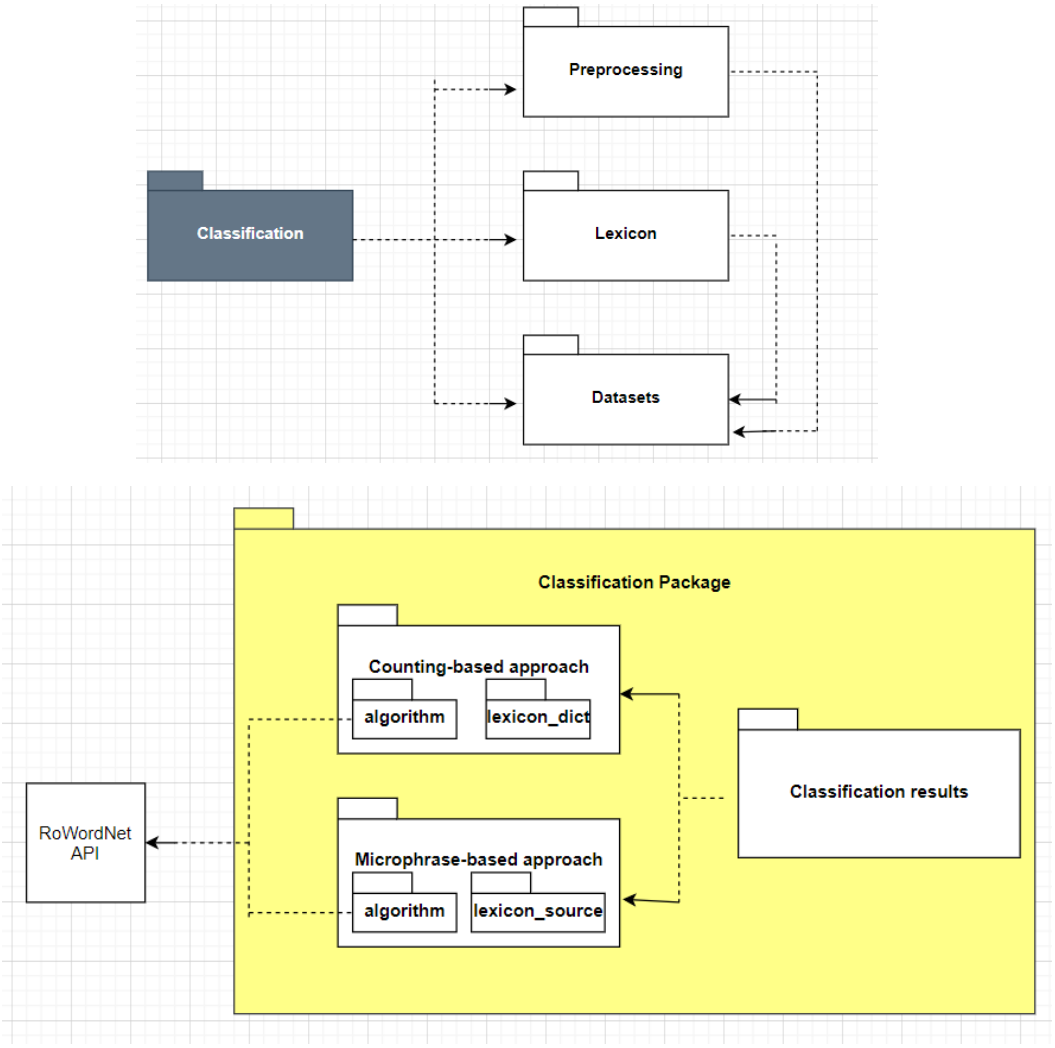
3. Preprocessing Package





The preprocessing package includes only one sub-package, containing all the preprocessing implemented functions. In order to perform the preprocessing properly, there is the need of an external dependency, by importing the NLP Cube API. The results of the preprocessed texts are stored in the Datasets Package

4. Classification Package



Undoubtedly, the Classification Package has the most complex structure, compared to the other application packages. One distinct sub-package is created for each of the two proposed algorithms: counting-based and microphrase-based. Their functionality is totally dependent on the RoWordNet API. After applying the algorithms on the selected datasets, the results are then collected in the sub-package "Classification results".

5.2. Utilization

The developed tool permits a user interaction with the functionality provided by the application. The elaborated graphical user interface allows the user to perform the following operations:

- Select the desired method (counting-based or microphrase-based algorithm);
- Entering the path of a source file containing a piece of text, which will be analyzed by the previously selected method;
- Visualizing the result provided by the selected method;

As depicted in Figure 9, the graphical user interface (GUI) displays the constituent components in a simple, yet organized manner. After filling in the file source path and selecting the desired method, by pressing the "Get Result" button, the sentiment analysis of the text is then illustrated suggestively in the Result text box, along with the classification result (stating clearly the polarity of the text).

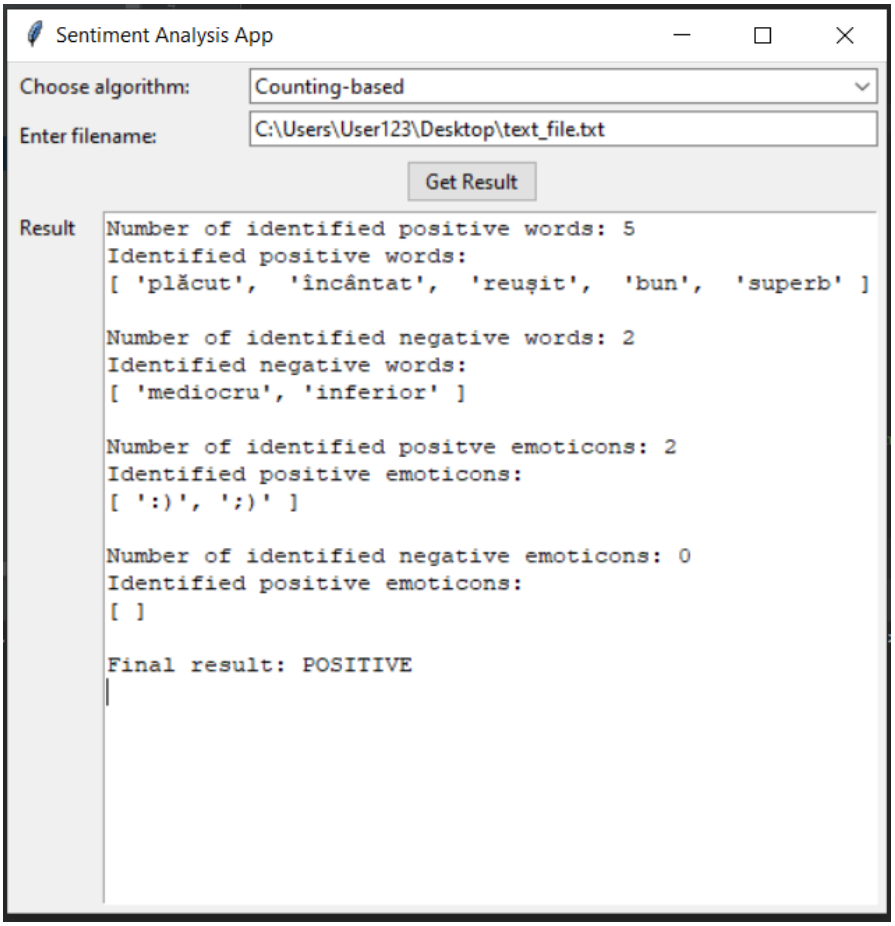


Figure 9 - Application User Interface

More details regarding the user interaction with the application are presented in the diagrams depicted in Figure 10 and in Figure 11 below. The purpose of the first diagram is to capture the functional requirements of a system, while the second one models high-level interactions between the active objects of the system.

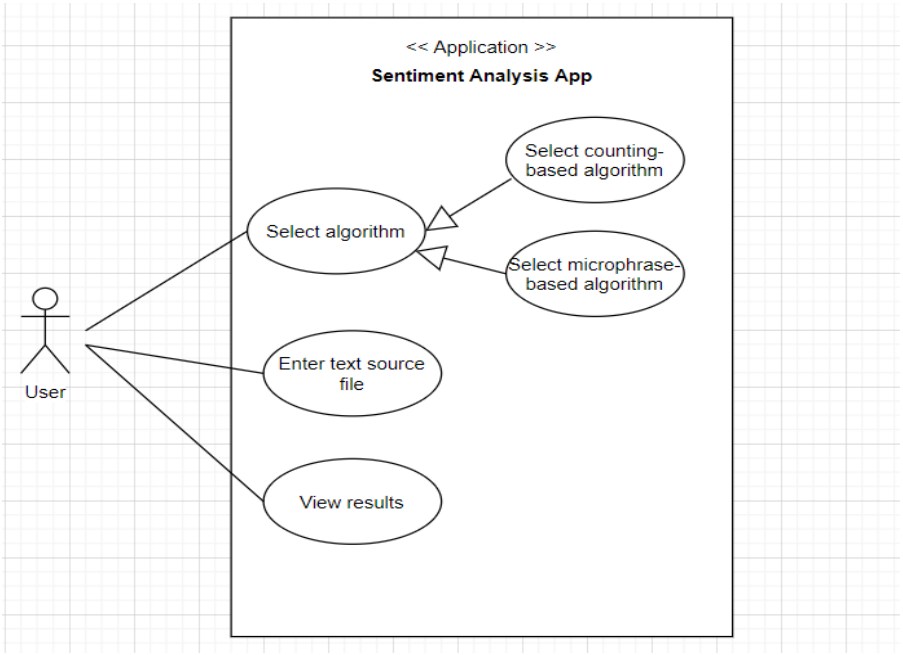


Figure 10 - Use Case Diagram

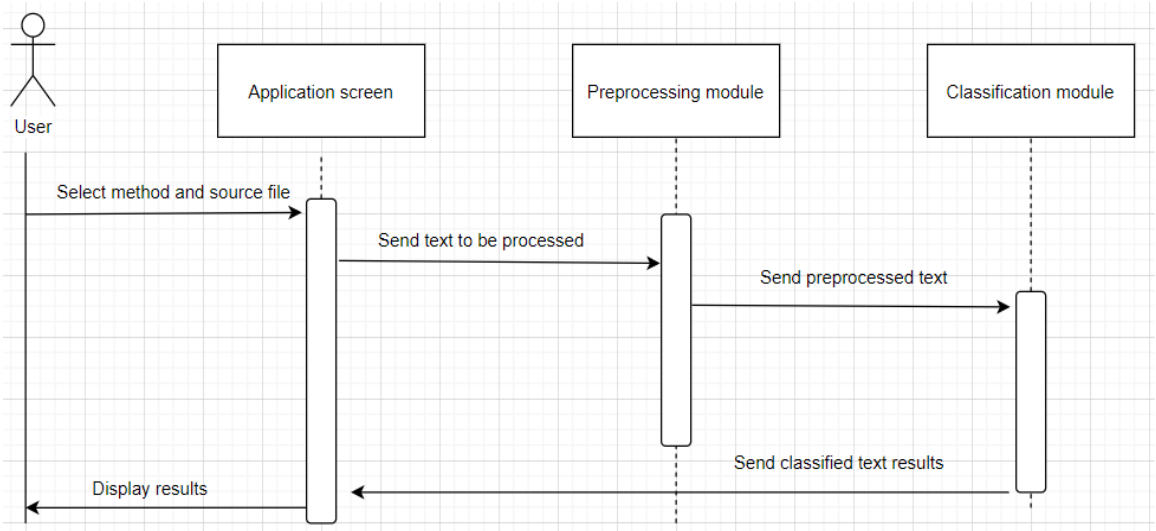


Figure 11 - Application Sequence Diagram

6. Conclusions and future work

The solution proposed by this bachelor thesis represents a complex lexicon-based approach of sentiment analysis of Romanian texts, by fulfilling the following objectives:

1. The conduction of a study with regards to the lexical and syntactical characteristics of the Romanian language, in the context of Sentiment Analysis domain. The research involved the studying of the Romanian particularities, behavior, vocabulary and grammatical rules.
2. The elaboration of an opinion lexicon for the Romanian language, consisting of 4780 opinion words. The obtained lexicon was furtherly enriched by a collection of 202 expressions and 97 modifiers. An interesting observation is that the opinion lexicon is predominantly composed of negative terms, indicating that Romanian might be a "pessimistic" kind of language.
3. The development of an application presenting two algorithms, each one differently approaching the sentiment analysis problem. The first method, a counting-based algorithm, was validated on a set of 127 movie reviews written in Romanian, and the resulting accuracy is around the value 72.4%. The second method involved a sentiment analysis strategy of computing the polarity of a document by analyzing the obtained positivity or negativity scores at phrase-level. This approach was tested on samples of various Romanian texts, and the results also provide a more specific classification, due to utilization of polarity degrees (neutral, slightly negative, very positive, etc.).

As a final thought, it is of great significance to mention that there are additional ways to improve the complexity and quality of the proposed solution. Firstly, the lexicon could be enlarged by performing several other iterations, enriching it not only by considering synonyms, but also by treating antonyms. Secondly, the accuracy of the algorithms could be improved by increasing the performance of negation detection and completing the set of analyzed language patterns.

Bibliography

[1] L. Bing, "Sentiment Analysis and Subjectivity," in *Handbook of Natural Language Processing*, 2010.

[2] S. A. Denny Matthew, "Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It," *Political Analysis,* vol. 26, pp. 1-22, March 2018.

[3] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, 2002.

[4] Khin Zezawar Aung, Nyein Nyein Myo. "Sentiment Analysis of Students' Comment Using Lexicon Based Approach," no. 978-1-5090-5507-4/17, 2017.

[5] Priyanka Patil, Pratibha Yalagi. "Sentiment Analysis Levels and Techniques: A Survey," *International Journal of Innovations in Engineering and Technology (IJIET),* vol. 6, no. 4, 2016.

[6] Mohan Vijayarani, "Preprocessing Techniques for Text Mining - An Overview," *International Journal of Computer Science & Communication Networks,* vol. 5, no. 1, pp. 7-16.

[7] Milagros Fernandez-Gavilane, Tamara Alvarez-Lopez, Jonathan Juncal-Martinez, Enrique Costa-Montenegro, Francisco Javier Gonzalez-Castagno. "GTI: An Unsupervised Approach for Sentiment Analysis in Twitter," *Proceedings of the 9th International Workshop on Semantic Evaluation,* pp. 533-538, 2015.

[8] Cataldo Musto, Giovanni Semeraro, Marco Polignano. "A comparison of Lexicon-based approaches," *CEUR Workshop Proceedings,* vol. 1314, pp. 59-68, 2014.

[9] Sahu Tirath, Ahuja Sanjeev. "Sentiment analysis of movie reviews: A study on feature selection & classification algorithms," in *International Conference on Microelectronics, Computing and Communications (MicroCom)*, Durgapur, 2016.

[10] Tufis Dan, Ceausu Alexandru. "DIAC+:A Proffesional Diacritics Recovery System," in *Language Resources and Evaluation Conference*, Marrakesch, 2008.

[11] Nicoleta Sava, "Romanian Negation - Slavic or Romance?," 2007.

[12] Stefan Daniel Dumitrescu, "ROWORDNETLIB – THE FIRST API FOR THE ROMANIAN WORDNET," *PROCEEDINGS OF THE ROMANIAN ACADEMY,* vol. 16, pp. 87-94, 2015.

[13] Boros Tiberiu, Dumitrescu Stefan, Burtica Ruxandra. "NLP-Cube: End-to-End Raw Text Processing With Neural Networks," 2018.

[14] Eynde Frank, Gibbon Dafydd., Lexicon Development for Speech and Language Processing, 2000.

[15] Laurentiu Theban, "The Syntactic Type of Romanian. The Basic Sentence.," *Revue Roumaine de linguistique, L(1),* vol. 27, pp. 179-194, 2006.