Russell conjugative debiased reading

by

Ana Maria Martinez Sidera

Submitted to the Department of Computer Science
in partial fulfillment of the requirements for the degree of

Master of Computer Science

at the

ILLINOIS INSTITUTE OF TECHNOLOGY

May 2018

Author . . . .
Department of Computer Science
May 18, 2018

Certified by . . . .
Shlomo Argamon
Professor of Computer Science
Thesis Supervisor

Abstract

In rhetoric, emotive or emotional conjugation mimics the form of a grammatical conjugation of an irregular verb to illustrate humans' tendency to describe their own behavior more charitably than the behavior of others. The use of subtly emotive language can bias interpretation of otherwise objective and accurate characterizations of people and events. Speechwriters and rhetoricians have used careful word choice to good effect. In this thesis, I built and deployed a prototype bias-revealing browser plugin. This web extension reveal hidden sources of emotive bias in news stories and web pages. The main goal of this project is to investigate to the extent to which Russell conjugations are used to bias rhetoric, to develop tools to make readers aware of such rhetorical tricks, and to investigate how such tools affect readers perceptions of bias and evaluation of information.

Thesis Supervisor: Shlomo Argamon
Title: Professor of Computer Science

Acknowledgments

First of all to my parents and my brother for their support, advice, understanding and help in difficult times. Thank you for teaching me an ideal of life grounded in daily work, in disinterested friendship, honesty and positive attitude.

To all my friends and colleagues. Thanks for all the support and for all the good moments that we have spent together among the four walls of this our university.

Contents

Chapter 1

Introduction

In this introduction we describe the project and define the main goals. In the final section, we also explain the structure of this report.

1.1 Background

The project originates to cover a need that today is not resolved, or even being partially resolved, have not evolve enough to has an extensive use among the natural language processing researchers. The field in which the project is framed is emotive conjugation or, how often it is called, Russell conjugation.

In terms of linguistics, psychology and rhetoric, Russell conjugation is an ambiguous construction that demonstrates how our rational minds are educated with a series of bias, instead of observing the basic meaning of words. For example, two words that are technically synonymous can have a completely different inference in our minds. Therefore, the sentimental perception we have of a text may vary depending on the words we use. It showed how easily my opinions could be manipulated without any need to falsify facts just change a word.

In order to understand the concept properly you have to appreciate that most words and phrases are actually defined not only by a single description, also rather by two or more distinct attributes:

- The factual content of the word or phrase.

- The emotional content of the construction.

However, frequently when we read or listen we forget the first description and we go directly to the most emotive meanings, mechanically and unconsciously. This is why many authors use this conjugation to influence you when you read or listen to a story in the press. The use of such emotional language to pre-emptively destroy one's opponents and prop-up one's heroes.

1.2 Description of the problem

When we tell a story we want to influence and transmit to our listeners what we want. When we listen we expect them to color their perceptions. Russell discussed this by putting three such presentations of a common underlying fact in the form in which a verb is typically conjugated:

- I am firm. [Positive empathy]

- You are obstinate. [Neutral to mildly negative empathy]

- He/She/It is pigheaded. [Very negative empathy]

In all three cases, the first description in the dictionary is to describe people who did not immediately change their minds. Placing this three verbs together we realize that most of our positive feelings are present in the firm person and the negative towards the pig-head person.

Speechwriters and rhetoricians have used careful word choice to good effect. The goal of this project is to investigate to the extent to which Russell conjugations are used to bias rhetoric, to develop tools to make readers aware of such rhetorical tricks, and to investigate how such tools affect readers perceptions of bias and evaluation of information.

The main goal of this project is to develop a web extension to reveal hidden sources of emotive conjugation of the web pages in general, and in particular of the news pages. The key idea is to identify adjectives, verbs, and nouns with positive or negative emotional content and present their Russell conjugations as alternative readings within the text. After the option is selected, the user will be able to see the different emotional conjugation. Moreover, the user will interact with the web page analyzing all the synonyms, one by one, all words together and another types of combinations that we will be detailing throughout this report.

For this reason, we need to make a sentiment classification of the words using the context of our page in order to adjust and categorize the dictionary synonyms. In the web page, we can visualize the words that have the same sentiment as the word used by the author. In the case of positive words, all synonyms will have a positive connotation in our web page.

1.2.1 Russell conjugation classifier

Sentiment analysis is used to classify people's attitudes towards a topic, document or event. In this research project we are going to have to use this technology to be able to find out the Russell conjugation of the people towards a text that in principle is objective. Little by little we will see how they can influence us in our feelings and emotions without us realizing it, being able to change our opinion about a topic.

Nowadays, sentiment classification has many applications, most notably in the consumer goods industry where it is commonly referred to as opinion mining. Manufacturers are keen to understand what consumers think about their products in order to gain valuable information about how to improve everything from design to advertising strategy.

One of the most widespread projects to start in the world of sentiment classification is the classification of comments on films on the IMDB page. Dividing the comments in two: train and test, and analyzing the use of the words and context that people use in the train we can get to make a quite effective model that classifies us the rest of the comments between positive and negative.

More examples include investigating the political climate, the sentiment toward any given party, election promise or can help agencies and PR firms gauge the successfulness of previously launched campaigns. As we have observed in the news the use of opinion mining is quite widespread today. As we can see with the recent news of the filtering of Facebook data for the electoral campaign:

"The data analysis company Cambridge Analytica illegally collected personal data from more than 50 million Facebook users to support Donald Trump's presidential campaign, as revealed on Saturday by The New York Times and London's Observer.

The company used the data collected without authorization in early 2014 to develop a computer program that predicted and influenced electoral choices. The source is Christopher Wylie, a former Cambridge Analytica employee who collaborated with a professor at the University of Cambridge to obtain the data to then present each elector with personalized political advertising."

As we can see the applications that have the classification of feelings are unlimited. The main problem is the collection of data for the classification, from conducting surveys to collecting data from social networks thanks to the APIs. Our project focuses on the classification of people's feelings towards certain words and in real meaning in their daily use. Thanks to this we can find out the inference that these words have and the impact it has on people.

1.2.2 Web extension

Extensions are small software programs that adjust or improve the browsing experience. They are built on web technologies such as HTML, JavaScript, and CSS. Web extensions make the experience more personal and focused on the use of each person. We can find all kinds of web extension on the internet.

An extension must achieve a single or multiple purpose that is defined. A single extension can include multiple components and a range of functionality, as long as everything contributes towards a common purpose.

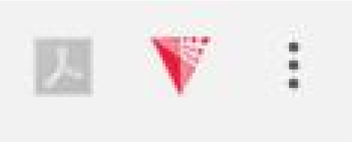


Figure 1-1: Example of web extension. Source: self made

Extension files are zipped into a single .crx package that the user downloads and installs. This means extensions do not depend on content from the web, unlike ordinary web apps.

In this project we will develop a web extension. The design will be as minimalist as possible, by clicking on the icon the application can be used. We will use HTML, JavaScript and for the part of the server we use Python. We will connect JavaScript and Python using the Flask framework that is available for Python.

Flask is a minimalist framework written in Python that allows you to create web applications quickly and with a minimum number of lines of code. We will use this framework to connect the Javascript of each plug-in with our web application.

1.3 Goals

The main goal of this project is to investigate to the extent to which Russell conjugations are used to bias rhetoric, to develop tools to make readers aware of such rhetorical tricks, and to investigate how such tools affect readers perceptions of bias and evaluation of information.

In this research project, we built and deployed a prototype bias-revealing browser plugin. This web extension reveal hidden sources of emotive bias in news stories and web pages. The following is a breakdown of some specific objectives to simplify the treatment of them:

1. Development of a web extension so that users can use and see the results on the website.
2. Analysis and classification of the different words according to their emotive conjugation depending on the context and its meaning.
3. Filtering of the different synonyms that we are going to show in the applications according to their emotive conjugation, so that they have the same inference.
4. Development of a server that can process all the information we receive from the web extension and then return the synonyms.
5. Change the web page with the different synonyms so that the user can interact with all of them and can appreciate the different connotations that the text has.

6. Select the perfect technologies Today there are multiple technological solutions for the same problem, so it is very important to make a good choice in each and every aspect of the project, since this will determine the success or failure of a possible commercialization of the same, both from the economic point of view, as well as the feasibility of the solution.

7. Learn. This project tries to give the author a way to enrich her knowledge about these technologies.

1.4 Project general description

After the choice of Research Project, a stage of research and documentation about the state of the art related to the topics mentioned in the previous sections was initiated. This study, as will be reflected below, must be carried out continuously during the entire duration of the project.

However, once a large part of it was completed, it set out to start the development with the different components to see the limitations that could be found in the future. To verify the characteristics that were required of the components, a methodology like the one shown in Fig. 1.2 was used. Iterating over this loop, more information was obtained about the possibilities solutions for our problem and for analyzed the technology we have to use. To conclude, the precision and accuracy of the chosen solution will be tested.

1.5 Structure of the report

The memory consists of 6 chapters and 2 appendices. This first chapter introduces the project to be treated. Chapters 2, 3, 4 and 5 make up the main part of the report. State of the Art (Chapter 2), where we explain the technologies involved and

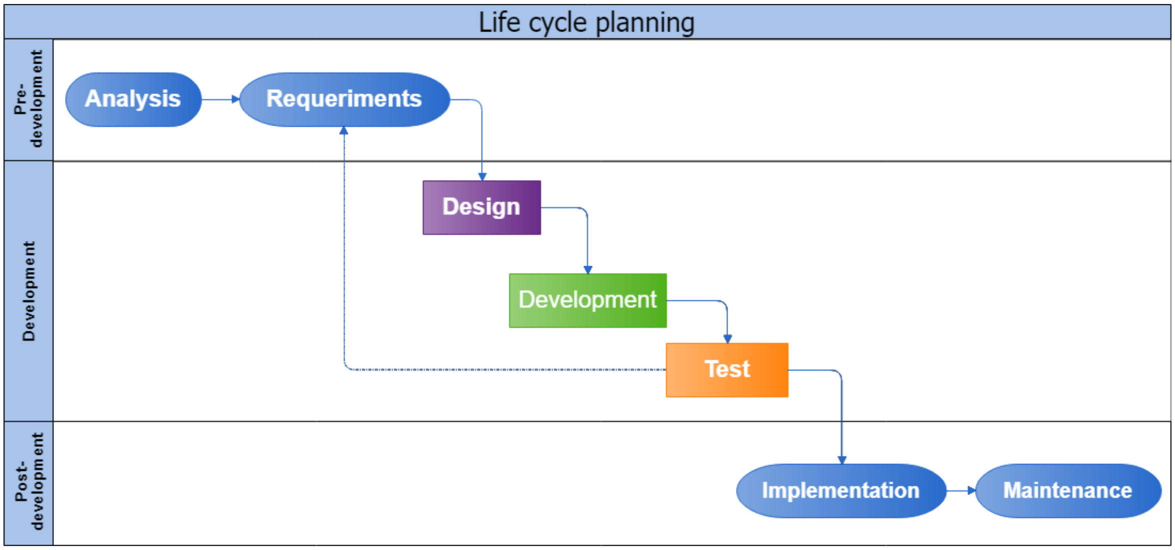


Figure 1-2: Life cycle planning

the theory of this project necessary to understand the subsequent sections; Analysis of the project (Chapter 3), which shows the entire design part related to technology; Proposed solution (Chapter 4), which consists of a description of the chosen system; and Development and implementation of the proposed solution (Chapter 5), which will resume the implementation of the entire project and how it has been managed from the beginning. Finally, in the chapter of Conclusions (Chapter 6) a review of the objectives achieved throughout the project will be made; and in Improvements and Future Work (Chapter 7) focuses on showing possible improvements to this project.

Chapter 2

State of the art

Under the next chapter, a review is made of the context in which the project is framed, that is, the state of the art or technique. The possible technologies and the different projects that are related to the application of the Research Project are analyzed. It also describes technologies that are fully related to the topics discussed such as Web extension and Corpus-based approach.

2.1 Description of technology and tools

Machine learning is a field of computer science that gives computer systems the ability to progressively improve performance on a specific task with the use of data. The name Machine learning was coined in 1959 by Arthur Samuel. Arthur Samuel wanted to get his computer to be able to beat him at checkers. How can you write a program, lay out in excruciating detail, how to be better than you at checkers? So he came up with an idea: he had the computer play against itself thousands of times and learn how to play checkers. And indeed it worked, in fact, by 1962, this computer had beaten the Connecticut state champion. Using this concept and keeping all the data collected from the games played invented the concept of machine learning.

2.1.1 Classification

One of the possible ways to solve our problem is using a part of the machine learning called classification. Classification is the problem of identifying to which of a set of categories a new observation belongs. In order to make this model we need data first to do a training. These data may have a label (i.e. positive or negative) or we may not have any label. For these two cases there are techniques and algorithms with which to deal with this problem and to be able to classify our new data.

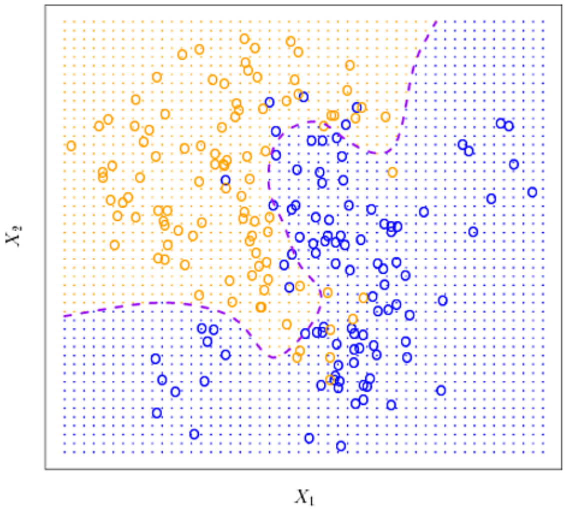


Figure 2-1: One type of classification algorithm. Source: http://www-bcf.usc.edu

An example would be assigning a given email into ”spam” or ”non-spam” classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient or features. Automatization of the classification process is therefore especially useful when big amounts of data need to be classified for purposes ranging from spam detection to image recognition. This report focuses on the classification of text data into two predefined classes based on its sentiment.

2.1.2 Sentiment classification

In the case of sentiment classification the data is categorized as positive or negative based on its connotation. It can determine the polarity of a text, i.e. whether the text carries a positive or negative connotation. Given a set of texts that discuss a common topic, a general opinion toward this topic can be determined by averaging the polarity of individual texts. There are two main approaches to sentiment classification based on a sentiment lexicon and machine learning as we can see in the Fig.2.2.

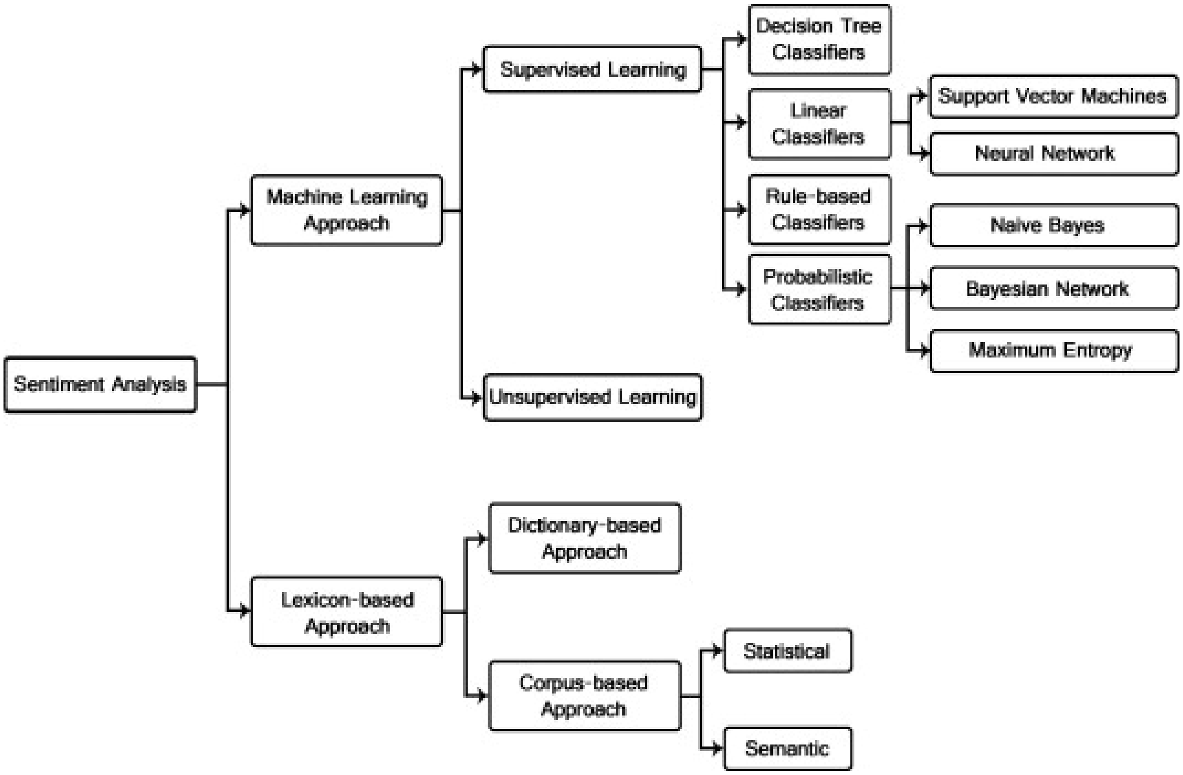


Figure 2-2: Sentiment classification techniques. Source: https://www.sciencedirect.com

First of all, using the lexicon based approach the text to be analyzed is tokenized into individual words whose polarity is looked up in a sentiment lexicon. The polarity of a word describes its underlying connotation, i.e. how positive or negative a word is. The text is then classified based on the sum of the polarity values of all the words.

On the other hand, the learning-based approach works by training a machine learning algorithm on a training set consisting of data which have been labeled as positive or negative prior to running the classifier. This data is then analyzed for patterns which indicate which combinations of words tend to be present in positive and negative texts.

2.2 Approaches to Corpus-based approach

The state-of-the-art in sentiment classification is essentially split into two approaches using a lexicon-based and a learning-based classifier respectively. There are three main classification levels in sentiment analysis: document, sentence, and aspect.

In the first level the entire document is taken into account, in the second only the sentences. Finally, only the aspects that we are classifying are taken into account. No matter what level you want to reach, the steps to follow in this algorithm are always the same. First of all we have to do a review of what we want to classify, whether documents or phrases. The next step is an identification of the feelings that we are going to look for in our documents, in a more positive document we will find words like good, happy. We make a selection of the features that we will find in our documents, since each text or type of document will have a different vocabulary. We tend to adapt our model to the documents or the vocabulary in which we are going to move. Once we have chosen our features we will make a classification of our documents. This process can be seen in the following figure.

In the following sections, we talk about some of the most used algorithms in sentiment analysis.

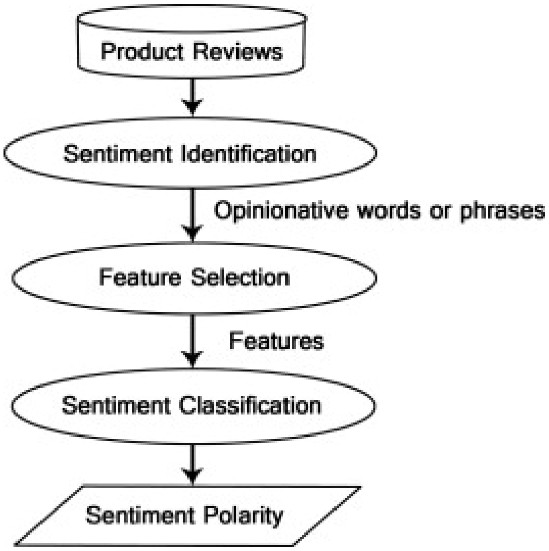


Figure 2-3: Sentiment analysis process on product reviews. Source: https://www.sciencedirect.com

2.2.1 The lexicon-based approach

In lexicon-based classification, documents are assigned labels by comparing the number of words that appear from two opposed lexicons, such as positive and negative sentiment. Creating such words lists is often easier than labeling instances, and they can be debugged by non-experts if classification performance is unsatisfactory. The lexicon-based approach uses an eponymous lexicon of words with classification weights associated to them. Depending on the implementation, the weights can be binary (positive/negative) or in a numerical range. These weights are referred to as the words polarity values and are defined by the sentiment lexicon used. The polarity value of each word that the text consists of are passed into an algorithm producing a score which decides the sentiment of the text. How the decision plays out differs between implementations, but a simple way is for negative values to indicate the negative class and positive values indicate the positive class.

Summation of polarity values

The first step when classifying a document is to trim the text we have. We have to eliminate special characters that can damage our algorithm, convert all the text to lowercase so that there is no difference between upper and lower case and finally divide the text into words without taking into account punctuation marks.

The next step is to see the polarity of each and every one of the words in the text we are analyzing and add them. If the final result is positive we will have a positive classification, otherwise it will be negative. There are libraries in python like AFINN that automatically calculate the polarity of a text taking into account this technique. With the algorithm explained above we have two problems that we will explain below:

Polarity threshold

When we account for the polarity of a text we hypothesize that the center between positive and negative is in the center of both and this is not correct. We may have negative texts that have been classified as positive by this threshold. This would cause our results not to be fully balanced and overload a part. That is why we are going to set a threshold so that the center is not at 0, this threshold does not have to be more on the negative side or on the positive side. When we do the research process we will give several values to the threshold and we will see the performance of our algorithm. We will choose the threshold based on the results obtained and according to the best accuracy.

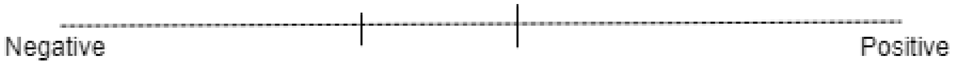


Figure 2-4: Binary classification threshold.

Negation words

The last problem we find in the lexicon-based approach is the negation. In some cases we may be denying a phrase which means that even if adjectives or nouns of positive polarity appear in the sentence they have to be implemented in the equation as negatives. If we consider the phrase:

"The weather does **not** look too **good**".

The word good has a positive polarity, but having the word not before causes its polarity to change drastically. This is why we have to take into account the negated phrases and apply a negative polarity to all the words that appear in that sentence, in order to solve the problem.

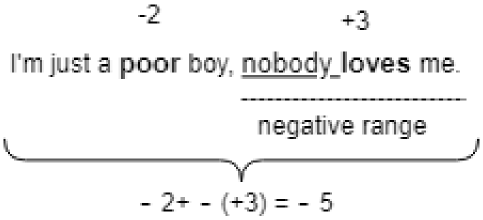


Figure 2-5: Polarity value.

Capital letters

As in the previous case we can be faced with the situation of using capital letters on the web page. In this case, the author wants to emphasize this word. We have to take this event into account and increase the polarity of this word.

Never or least sentence

In the case we use the word never or least in our phrase, the following words are conditioned by it. Therefore we have to take this event into account. Inspect the whole word for the word never and if yes, change the polarity of the following words.

Degree adverbs

In the case that we use adverbs, in many cases they increase or decrease the polarity of the word. It gives more intensity to the words that come later. Therefore we have to check the words that come before and in case of an adverb increase or decrease the result we get for that word.

But sentence

In the middle of a sentence we can find the word ’but’. This word drastically changes the polarity of the rest of the sentence, giving a meaning opposed to the previous one. Therefore we have this aspect and we change the polarity from the word ’but’.

Question and exclamation marks

When there are a series of question marks or exclamation marks they give greater emphasis to the previous sentence. This is increased if instead of 1 they use more than 3. Therefore, in our program we take into account the amount of exclamation or question mark that the author uses to increase the polarity.

2.2.2 The learning-based approach

The learning-based approach for sentiment classification utilizes machine learning to build a classifier. In our case, we do not have data that is classified in advance that can help us, so we need to apply unsupervised learning techniques to develop this approach in a future project. In text classification, it is sometimes difficult to create these labeled training documents, but it is easy to collect the unlabeled documents. The unsupervised learning methods overcome these difficulties. Many research works were presented in this field including the work presented by Ko and Seo [81]. They proposed a method that divides the documents into sentences, and categorized each sentence using keyword lists of each category and sentence similarity measure. In this project, we will not work on this approach. I just want to highlight it for possible future projects.

2.3 Similar projects

Currently, through search on the Internet it is possible to find projects related to the one described in this report. However, on rare occasions one can find one that integrates all the technologies together to form a complex system like the one developed in this Research Project.

Chapter 3

Analysis of the project

This chapter focuses on analyzing the different parts of the project, such as the objectives to obtain a series of requirements, as well as the most convenient technology to use for the proposed solution. To do this, diagrams will be used to show both the requirements capture phase and the functional analysis of the system to be developed.

3.1 Service concept

A software service is a functionality of this one implemented by software and/or people that the user receives as a benefit for the sake of security, comfort or leisure. The particularities of a service are the following:

- *The services are perceived by users with different degrees of quality.*
- *The services are not only an isolated product, they involve a controlled implementation, a sometimes remote administration, and a maintenance of equipment and software.*

> - *Some services involve the collaboration of external systems that can be operated by people with a availability of 24 hours for 365 days.*
>
> *The most important aspects of a service are the following: response time, capacity, resource utilization, failure rates, recovery capacity, administration and maintenance. Jos Luis Fernandez Sanchez, requirements analysis seminar.*

The characteristics of a service that have been mentioned above are part of very important aspects of the project, and several of them will be the basis of some of the requirements that will be obtained throughout this chapter. Once described what is a service, it can be seen that the development of the project is framed in this area, and the following sections will detail the design of that service.

3.2 Analysis of requirements

The process of analyzing requirements is a very important aspect in any type of project, since the list of requirements generated at the end of this process will condition the entire evolution of the project. In addition, the management of requirements during the execution of the project is equal, or even more important, since it is possible that new requirements may be eliminated, modified or added, and the project will have to adapt in the best possible way to the new needs for satisfy them. The typical stages that are usually followed in the process of analyzing requirements are the following:

Capture and analysis of requirements It is the first stage that must be carried out. There are several sources to obtain the requirements, such as: business objectives, domain of knowledge, stakeholders in the project and operational environment of the system. The techniques to capture these requirements are also established, and it is important to follow these methodologies since they are standardized processes and better results are obtained than if they were not followed. The most techniques Common are: market analysis, interviews, scenarios of system use, prototypes, guided meetings and observation of similar systems in operation.

Specification of requirements This stage consists of completing the information of the requirements generated by the previous stage. It is important to assign to each requirement a series of attributes, which will help complement and improve the list of requirements obtained in the previous stage. The attributes that accompany each requirement are: identification, priority for the client, criticality from the point of view of the system, technological viability, risk and origin of the requirement. Once this is completed information, the modifications of the requirements that happen throughout the project will be integrated in an easier way.

Feasibility of requirements This stage has as main objective the generation of a traceability matrix through which it can be observed in a visual way that the functionalities of the system cover all the requirements that have been obtained in the previous stages, since said matrix consists in arranging in columns the list of requirements obtained previously, and in rows the functionalities offered by the system.

Requirements management This stage is carried out throughout the project, and it is essential to have a good management of the requirements because if, for example, it happens that the project is carried out for a specific client, it is normal that there is changes in the requirements during the duration of the project, and if the changes are not managed well can incur large costs and delays in deliveries.

In the realization of the present project it was wanted to apply some of the concepts of analysis of requirements previously, it has been decided to carry out the stage of capture of requirements by means of the use of scenarios and cases of use. It is expected to get a list of functional and non-functional requirements of the system.

Once all the necessary information has been compiled with the previous techniques, we proceed to make use cases of the system. In Figure 3.1 you can see the use cases that have been identified for the system.

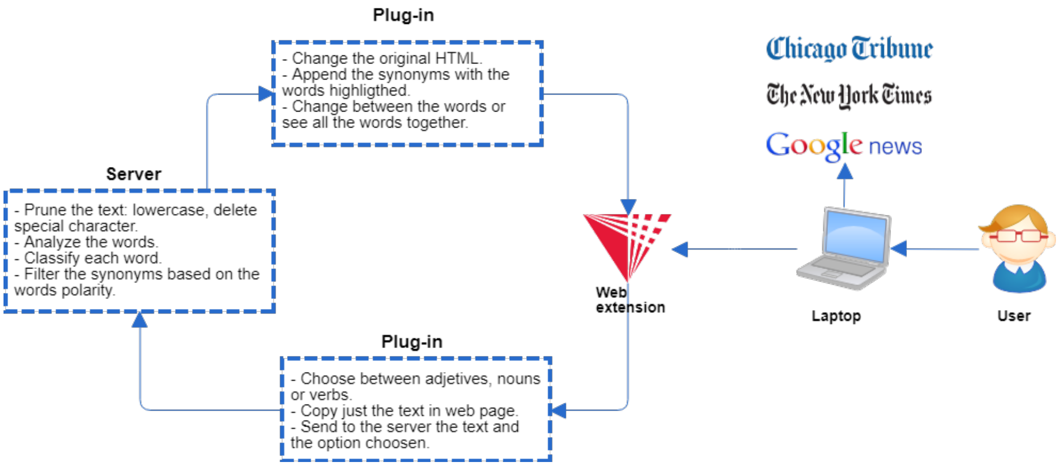

Plug-in
- Change the original HTML.
- Append the synonyms with the words highligthed.
- Change between the words or see all the words together.
Chicago Tribune
The New York Times
Google news
Server
- Prune the text: lowercase, delete special character.
- Analyze the words.
- Classify each word.
- Filter the synonyms based on the words polarity.
Web extension
Laptop
User
Plug-in
- Choose between adjetives, nouns or verbs.
- Copy just the text in web page.
- Send to the server the text and the option choosen.

Figure 3-1: Case of use.

The description of each use case is detailed below:

Table 3.1: Case of use number 1

Id	**CU1**
Name	Plug-in start
Principal components	User, plug-in, server and news website
Start event	Click on the plug-in
Description	The user clicks the plug-in and selects an option between adjective, noun or verb while viewing a news web page

Each use case can have one or several flows of interactions between the actors and the system, and each one constitutes a phase.

For the CU1 use case, thirteen phases are defined:

- The user selects one of the options in the pop-up window. (phase 1.1)
- The user clicks the plug-in. (phase 1.2)
- The plug-in collects all the text. (phase 1.3)
- The plug-in sends the text and the selected option to the server application. (phase 1.4)
- The server prunes the text: lower case, delete special character. (phase 1.5)
- Analyze each of the words. (phase 1.6)
- Score polarity of the words. (phase 1.7)
- Filtering of synonyms based on their polarity. (phase 1.8)
- The server sends the result to the plug-in. (phase 1.9)
- The HTML of the initial web page is modified according to the results obtained. (phase 1.10)
- The synonyms are added and the words used are highlighted. (phase 1.11)
- You can change the words or you can observe them all together. (phase 1.12)
- The user can see and interact with the new visualization. (phase 1.13)

The detailed description of all phases is made in the following tables:

There are slight variations of the phases described above, but those cases will be taken into account directly in the development of the final applications. With the

Table 3.2: Phase1.1: Options

ID	**Phase1.1**
Name	Options
Case of use id	CU1
Pre-conditions	User connected to the internet
Post-conditions	Phase1.2
Description	The user can analyze the web page based on their selection.
Requirements	Not applicable.

Table 3.3: Phase1.2: Plug-in

ID	**Phase1.2**
Name	Plug-in
Case of use id	CU1
Pre-conditions	Phase1.1
Post-conditions	Start pruning the HTML code.
Description	The user presses the plug-in button to start.
Requirements	We have to have a choice selected from: adjective, noun or verbs.

Table 3.4: Phase1.3: HTML Text

ID	**Phase1.3**
Name	HTML Text
Case of use id	CU1
Pre-conditions	Phase1.2
Post-conditions	Send a POST to the server application.
Description	Collect all the text that we find on the web page where the plug-in has been selected.
Requirements	Not applicable.

Table 3.5: Phase1.4: Send information

ID	**Phase1.4**
Name	Send information
Case of use id	CU1
Pre-conditions	Not applicable
Post-conditions	The server has to receive the information.
Description	We send the selected option and the text of the web page.
Requirements	Server has to be working and we can establish a connection with it.

Table 3.6: Phase1.5: Prune text

ID	**Phase1.5**
Name	Prune text
Case of use id	CU1
Pre-conditions	Phase1.4
Post-conditions	Not applicable.
Description	Prune in received text: lower case each word or delete special character.
Requirements	Not applicable.

Table 3.7: Phase1.6: Analyze

ID	**Phase1.6**
Name	Analyze
Case of use id	CU1
Pre-conditions	Phase1.5
Post-conditions	
Description	Analyze each of the words and their context.
Requirements	Not applicable.

Table 3.8: Phase1.7: Score polarity

ID	**Phase1.7**
Name	Score polarity
Case of use id	CU1
Pre-conditions	Phase1.6
Post-conditions	Each word is classified.
Description	Score polarity between positive or negative polarity of each of the words.
Requirements	Not applicable.

Table 3.9: Phase1.8: Filter

ID	**Phase1.8**
Name	Filter
Case of use id	CU1
Pre-conditions	Phase1.7
Post-conditions	Filtering of synonyms with the same polarity as the word.
Description	We make a study of the polarity of the synonyms of this word and classify them according to their emotive conjugation.
Requirements	Not applicable.

Table 3.10: Phase1.9: Send back

ID	Phase1.9
Name	Send back
Case of use id	CU1
Pre-conditions	Phase1.8
Post-conditions	The plug-in must receive the server information.
Description	We send the result of the score polarity to the plug-in so that it modifies the HTML of the web page.
Requirements	Need to do the score polarity.

Table 3.11: Phase1.10: New HTML

ID	Phase1.10
Name	New HTML
Case of use id	CU1
Pre-conditions	Phase1.9
Post-conditions	The user can see a new web page.
Description	Modification of the HTML to add the synonyms and the user can interact.
Requirements	Not applicable.

Table 3.12: Phase1.11: Interact

ID	Phase1.11
Name	Interact
Case of use id	CU1
Pre-conditions	Phase1.10
Post-conditions	The user can interact with the different synonyms.
Description	The user can change the words he reads or he can put all the synonyms together.
Requirements	Not applicable.

above phases it is possible to extract a list of both functional and non-functional requirements, which will form the basis of the development and implementation of the system.

The requirements of the system or service that will be obtained at the end of this section are the result of the sum of the user requirements obtained in the stage of capture of requirements and technological knowledge. It is important that the requirements that are obtained comply with the following properties:

Abstract The requirement must be independent of the solution.

Not ambiguous There should be no confusion in the interpretation of the requirement.

Traceable It is important to know if the requirement has been originated in consequence of another requirement.

Ascertainable It must be possible to verify that the final system meets a specific requirement. For this, a technique that is usually used is that when you specify a requirement, you think at the same time how it is going to check that the system complies.

Taking these properties into account, the functional requirements (FR) of the system are specified below, which should indicate what actions the system should be able to perform:

FR01 A connection must be established between the plug-in and the server.

FR02 The system must be able to send information between the plug-in and server and vice versa.

FR03 The system must be able to classify.

FR04 The system must be able to filter the synonyms.

FR05 The user can watch all kind of web pages and have the same result.

FR06 The user must be able to interact with all the words.

And the non-functional requirements (NFR), which should be oriented to the quality of the service and how the actions should be performed, are:

NFR01 The system must have a 99.99% availability. We can ensure that we will meet the demand for software for our Web extension. .

NFR02 The web extension must have an intuitive use and do not need additional instructions.

NFR03 The exchange of messages between the application of the web extension and the server should be carried out as quickly as possible.

NFR04 The system must be easily expandable and conserved by the developers adjusting to the needs of the potential clients.

NFR05 The interaction with the system must be intuitive for all users.

NFR06 The system has to be designed to be well maintained and the application can be developed and its features can be expanded.

3.3 Functional analysis

This section tries to describe what the system is capable of performing. To do this, we will mention the individual functionalities that the system can perform through the different components of the system. These functionalities are not sufficient to describe the complete behavior of the system, therefore, it should be complemented with the capabilities. In the first place, the basic functionalities in the form of a tree are shown in Figure 3.2 The functionalities have been divided by the components that are connected to the web extension and that in the end system must be integrated to work together and offer capabilities collaborating with each other.

The Fig 3.2 functionalities are the basis of the system's capabilities. These capabil-

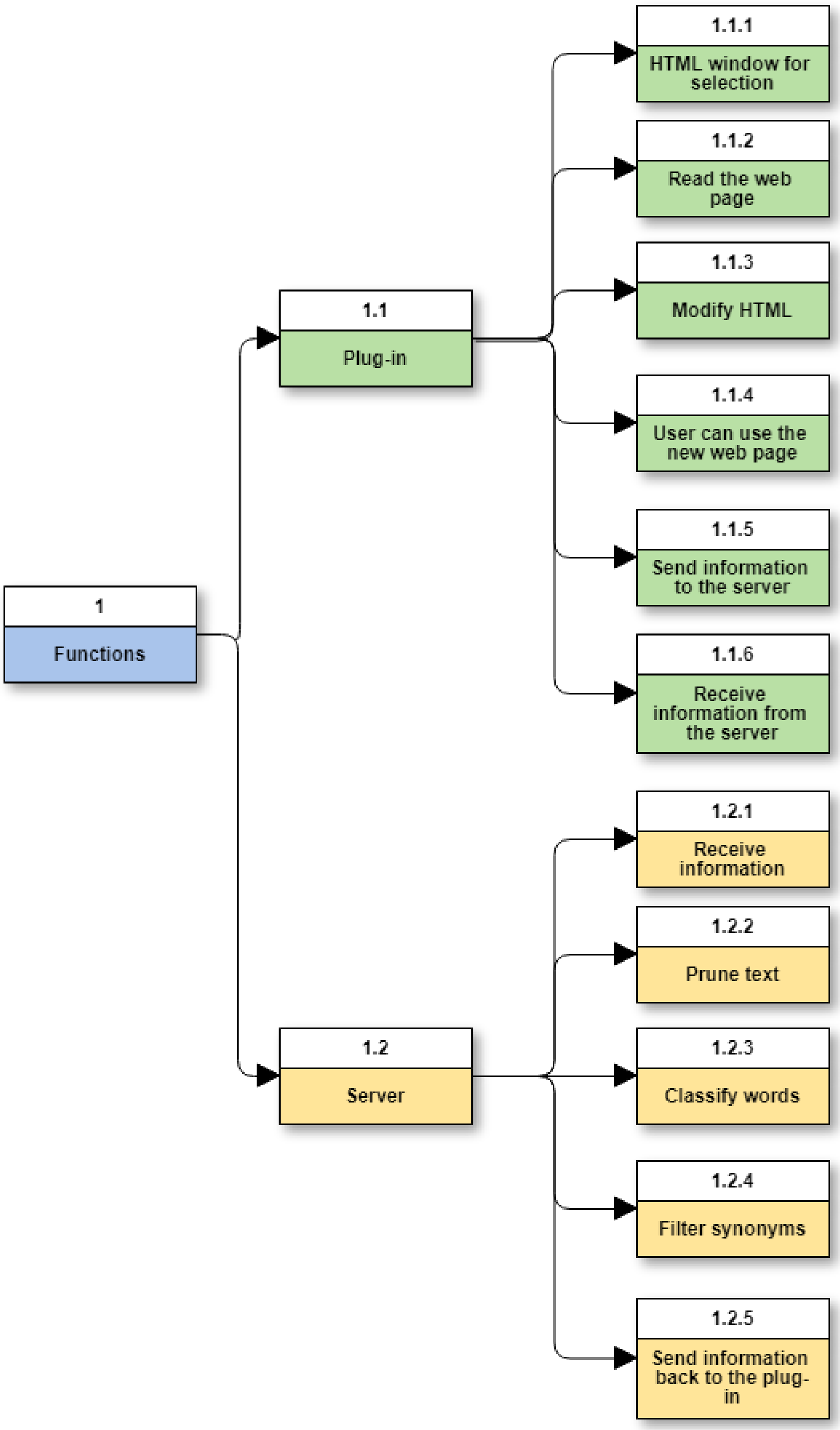


Figure 3-2: Functionalities.

ities, specified in Table 3.3, serve to carry out the requirements specified in sec. 3.2, and are ultimately based on the specified basic functionalities. Generally, functionalities of different components will be combined to generate complicated capabilities.

Chapter 4

Proposed solution

4.1 Summary

In this project we have developed a plug-in for Chrome in JavaScript and a background in python to perform the process of analysis of synonyms and adaptation of the new page in HTML. Before being able to use it, we had to analyze some 17,000 news web pages to make an extensive vocabulary with words to know its polarity. In the following points we will explain each of the steps.

4.2 Architecture

The architecture that we have made in this project is the following: on the one hand we have the plug-in that we have made using HTML JavaScript and CSS. A connection between the plug-in and the background using the Flask tool for python. And in the background part we have python and a vocabulary that we have created by analyzing 17,000 news web pages.

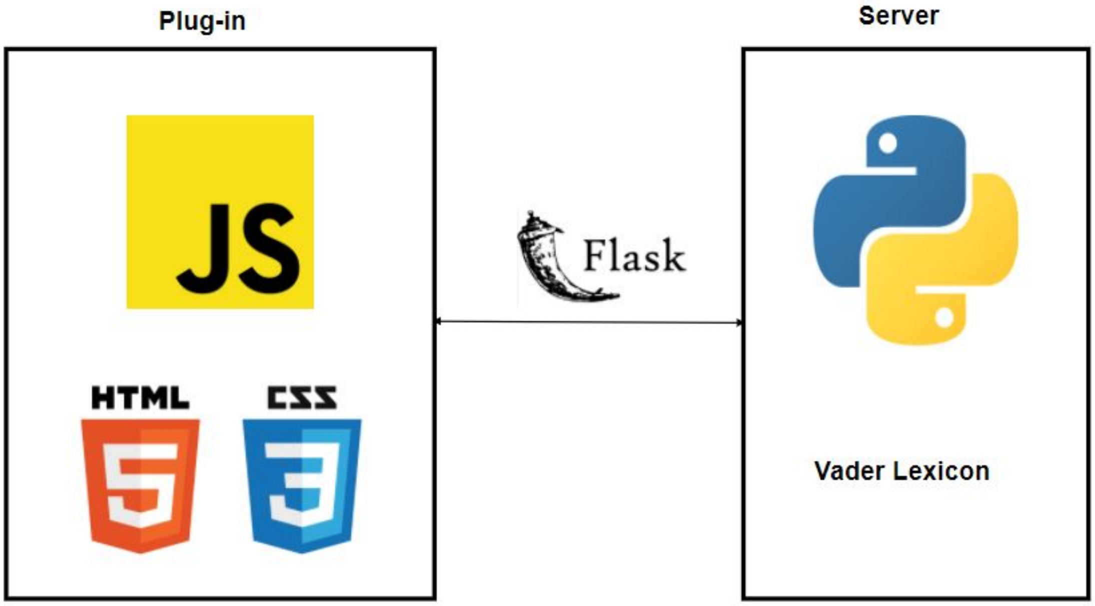
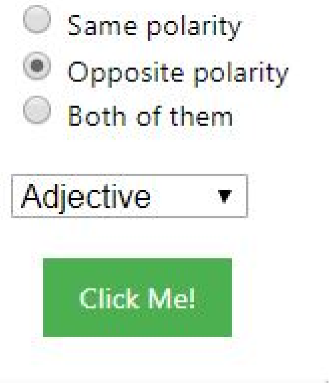


Figure 4-1: Architecture

4.2.1 Web extension

In the web extension we have the following possibilities first of all we can observe the synonyms that have the same polarity as the one we have selected. In case we want synonymous with the opposite polarity or we want to see all the synonyms. In all cases we can select among all adjectives, noun or verbs that appear on the website. Only the words that have at least one synonym in our vocabulary dictionary will be selected. Otherwise, it will not be highlighted. Once we click the web extension it will pass all the background information to analyze it. Then send all the information to the website and the words will be highlighted. To use it we will simply have to click on the word and we will have a menu with all the possible synonyms that we can use. If we press any synonym, the original word will be replaced and we can read the text with this new word. Another of the options we have will be to see all the synonyms in the text. In this case the web extension will substitute the word for all



○ Same polarity

● Opposite polarity

○ Both of them

Adjective ▼

Figure 4-2: Web extension example

the synonyms and we can read in text with all the available synonyms.

4.2.2 Server

In the background we have a server using Python. In this case the URL of the web page will arrive to be able to extract the text, the type of word that we want to analyze (adjective, verb or noun) and that it is of the same or different polarity than the original word.

With this data the Python file analyzes all the words and in case of finding an adjective, in the case of being analyzing adjectives, it will look for synonyms in the NLTK dictionary. With these words we will go to our dictionary and look for the synonyms that have the same polarity, in the case that the user has selected the same polarity. Once we have the words, python will add in HTML and CSS format the button and the different options returning the new HTML to our web extension. The web extension shows this HTML on the web page, so it is not necessary to open a new page.

4.2.3 Classification

We have used the classification part to create the dictionary that the background uses. The following features to analyze the feelings of each of the phrases of the 17,000 news web pages that we have downloaded.

- Negation words: The last problem we find in the lexicon-based approach is the negation. In some cases we may be denying a phrase which means that even if adjectives or nouns of positive polarity appear in the sentence they have to be implemented in the equation as negatives.

- Capital letters: As in the previous case we can be faced with the situation of using capital letters on the web page. In this case, the author wants to emphasize this word. We have to take this event into account and increase the polarity of this word.

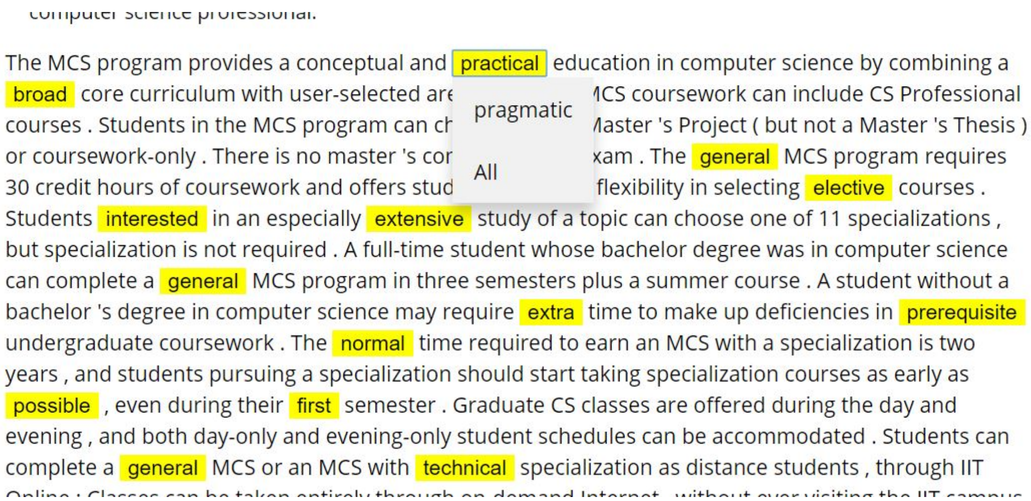
- Never or least sentence: In the case we use the word never or least in our phrase, the following words are conditioned by it. Therefore we have to take this event into account. Inspect the whole word for the word never and if yes, change the polarity of the following words.

- Degree adverbs: In the case that we use adverbs, in many cases they increase or decrease the polarity of the word. It gives more intensity to the words that come later. Therefore we have to check the words that come before and in case of an adverb increase or decrease the result we get for that word.

- But sentence: In the middle of a sentence we can find the word 'but'. This word drastically changes the polarity of the rest of the sentence, giving a meaning opposed to the previous one. Therefore we have this aspect and we change the polarity from the word 'but'.

- Question and exclamation marks: When there are a series of question marks or exclamation marks they give greater emphasis to the previous sentence. This is increased if instead of 1 they use more than 3. Therefore, in our program we take into account the amount of exclamation or question mark that the author uses to increase the polarity.

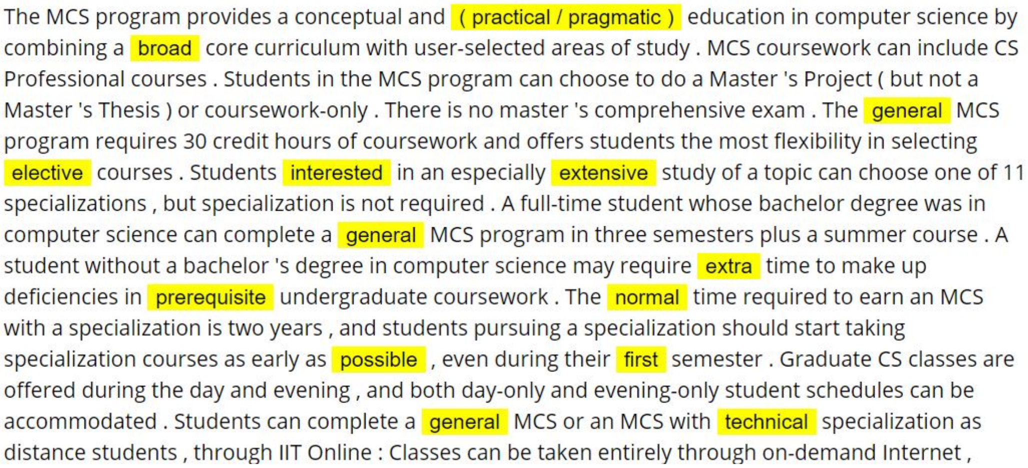
Once analyzed a sentence we saved in a dictionary each of the words used in this sentence and the polarity that they had in that part of the text. At the end of the entire study we made the average of all the data obtained, to obtain an approximation of the polarity of each of the words. For this we have taken into account the position in the sentences, the number of times the word was used in a negative character or the number of adverbs that increased the polarity of the words. When we finished we created a dictionary with all these words so that it would be easier and faster to pass them to the web extension.



computer science professional.

The MCS program provides a conceptual and practical education in computer science by combining a broad core curriculum with user-selected are MCS coursework can include CS Professional courses . Students in the MCS program can ch Master 's Project (but not a Master 's Thesis) or coursework-only . There is no master 's cor xam . The general MCS program requires 30 credit hours of coursework and offers stud flexibility in selecting elective courses . Students interested in an especially extensive study of a topic can choose one of 11 specializations , but specialization is not required . A full-time student whose bachelor degree was in computer science can complete a general MCS program in three semesters plus a summer course . A student without a bachelor 's degree in computer science may require extra time to make up deficiencies in prerequisite undergraduate coursework . The normal time required to earn an MCS with a specialization is two years , and students pursuing a specialization should start taking specialization courses as early as possible , even during their first semester . Graduate CS classes are offered during the day and evening , and both day-only and evening-only student schedules can be accommodated . Students can complete a general MCS or an MCS with technical specialization as distance students , through IIT Online : Classes can be taken entirely through on-demand Internet , without ever visiting the IIT campus

Figure 4-3: Example 1



The MCS program provides a conceptual and (practical / pragmatic) education in computer science by combining a broad core curriculum with user-selected areas of study . MCS coursework can include CS Professional courses . Students in the MCS program can choose to do a Master 's Project (but not a Master 's Thesis) or coursework-only . There is no master 's comprehensive exam . The general MCS program requires 30 credit hours of coursework and offers students the most flexibility in selecting elective courses . Students interested in an especially extensive study of a topic can choose one of 11 specializations , but specialization is not required . A full-time student whose bachelor degree was in computer science can complete a general MCS program in three semesters plus a summer course . A student without a bachelor 's degree in computer science may require extra time to make up deficiencies in prerequisite undergraduate coursework . The normal time required to earn an MCS with a specialization is two years , and students pursuing a specialization should start taking specialization courses as early as possible , even during their first semester . Graduate CS classes are offered during the day and evening , and both day-only and evening-only student schedules can be accommodated . Students can complete a general MCS or an MCS with technical specialization as distance students , through IIT Online : Classes can be taken entirely through on-demand Internet ,

Figure 4-4: Example 2

Chapter 5

Conclusions

Once the project is finished, it is time to review the goals that were defined at the beginning of the project to see if it has been possible to achieve all of them, as well as the new goals that have appeared during the execution of the project.

- Development of a web extension so that users can use and see the results on the website.
- Analysis and classification of the different words according to their emotive conjugation depending on the context and its meaning.
- Use of different classifiers and see the performance of each one to show a better result to our users.
- Filtering of the different synonyms that we are going to show in the applications according to their emotive conjugation, so that they have the same inference.
- Development of a server that can process all the information we receive from the web extension and then return the synonyms.
- Change the web page with the different synonyms so that the user can interact with all of them and can appreciate the different connotations that the text has.

- The adequate choice of technologies has allowed to implement all the features planned in the developed system of the proposed solution.

- Despite the high initial design load in the project, and the need for individual verification of the behavior of each component, it has been achieved integrate all this into a practical application ready to be used by a client without the need of technical knowledge.

- The project has served to enrich the author's knowledge in all the technologies used throughout it, creating a broad base of skills that can serve professionally in the near future.

Chapter 6

Improvements and future work

To conclude the report, we will proceed to mention the possible improvements of the proposed solution and the future work that would make sense to evolve the system along the same lines as before. Some are related to changes in the modules already implemented in order to improve their performance or quality, others that consist of adding new elements to increase the functionality of the system, and others to be able to apply the same technology used in this project to other fields. Below is an enumeration of all:

A more elaborate implementation of the lexicon-based classifier was deemed unnecessary, since the achieved results were sufficient to satisfactorily answer the problem statement. Notwithstanding, the performance of the proposed hybrid model depends on the performance of each of its parts. The upper performance limit is implicitly set by the used learning-based classifier, since the hybrid model cannot perform better than any of its individual parts, and it is evident from the results (table 5.2) that the learning-based classifier outperforms the lexicon-based. It would be a mistake, however, to disregard the latter as bearing little effect on the end results. This is due to the fact that the performance of the learning-based classifier depends to a large degree on the quality of the training data, which under the hybrid model is derived through the lexicon-based model. Thus, the performance of the lexicon-based classifier has a direct impact on the results, and could therefore be analyzed for opportunities for improvement.

An improvement could be augmentation of the negation range feature, to allow the range to extend backwards. This way phrases such as I expected the movie to be good, but it wasnt , would be correctly labeled as negative instead of positive since the negation range around wasnt would include good and switch its polarity value.

In a similar fashion, the learning-based classifier could be improved as well. Improvements to its performance could possibly be achieved by optimizing more of the available parameters that were not brought up in this report.

Further improvements could be made in the area of natural language processing, so that elusive language concepts such as sarcasm and irony, which have an effect on the connotation of a given phrase would be properly recognized. Such advancements would undoubtedly lead to higher classification performance. On a final note, it could be interesting to investigate and compare the performance of unsupervised learning-based classifiers, which do not require any data other than the data to be classified by using clustering algorithms. This means neither a lexicon nor training data is needed, and speaks for the convenience of this model. The interesting question would be if the performance of such approaches is comparable to those explored in this report.

Bibliography

[1] Eric R. Weinstein, *What scientific term or concept ought to be more widely?*, https://www.edge.org/response-detail/27181

[2] Fredrik Sommar and Milosz Wielondek, *Combining Lexicon- and Learning-based Approaches for Improved Performance and Convenience in Sentiment Classification*, Degree project in computer science, KTH Royal institute of technology

[3] Revolvy, *Emotive conjugation* , https://www.revolvy.com/main/index.php

[4] Google, *What are extensions?*, https://developer.chrome.com/extensions

[5] Flask, http://flask.pocoo.org/

[6] Walaa Medhat Ahmed Hassan Hoda Korashy, *Sentiment analysis algorithms and applications:*,School of Electronic Engineering, Canadian International College, https://www.sciencedirect.com

[7] Jessica Guynn, *Facebook data of 50M users exploited by Trump data firm, say reports; firm suspended*, https://www.usatoday.com/story

[8] Wikipedia, *Machine Learning*, https://en.wikipedia.org/wiki/Machine_learning

[9] Wikipedia, *Statistical classification*, https://en.wikipedia.org/wiki/Statistical_classification

[10] Cornell University, Department of Computer Science *Performance measures*, https://www.cs.cornell.edu/courses/cs578/2003fa/performance_measures.pdf

[11] Jacob Eisenstein, *Unsupervised Learning for Lexicon-Based Classification*, https://arxiv.org/abs/1611.06933

[12] Finn rup Nielsen, *"A new ANEW: evaluation of a word list for sentiment analysis in microblogs"*, Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages, Volume 718 in CEUR Workshop Proceedings: 93-98. 2011 May, Matthew Rowe, Milan Stankovic, Aba-Sah Dadzie, Mariann Hardey

[13] John Rothfels, Julie Tibshirani, *Unsupervised sentiment classification of English movie reviews using automatic selection of positive and negative sentiment items*, https://nlp.stanford.edu/courses/cs224n/2010/reports/rothfels-jtibs.pdf

[14] Ko Youngjoong, Seo Jungyun, *Automatic text categorization by unsupervised learning. In: Proceedings of COLING-00*, The 18th international conference on computational linguistics; 2000.

[15] Yulan He, Deyu Zhou, *Self-training from labeled features for sentiment analysis*

[16] Wikimedia, *Chicago Tribune Logo*, https://upload.wikimedia.org/wikipedia/commons/thumb/c/c4/Chicago_Tribune_Logo.svg/2000px-Chicago_Tribune_Logo.svg.png

[17] Wikimedia, *The New York Times logo*, https://upload.wikimedia.org/wikipedia/commons/7/77/The_New_York_Times_logo.png

[18] Wikimedia, *Google-News logo*, https://upload.wikimedia.org/wikipedia/commons/2/23/Google-News_logo.png

[19] Liparas, D., HaCohen-Kerner, Y., Moumtzidou, A., Vrochidis, S., & Kompatsiaris, I. , *News articles classification using Random Forests and weighted multimodal features.*,In Information Retrieval Facility Conference (pp. 63-75). Springer International Publishing.

[20] Dua, D. and Karra Taniskidou, E., *UCI Machine Learning Repository* , [http://archive.ics.uci.edu/ml], Irvine CA: University of California, School of Information and Computer Science

[21] Dheeru, Dua and Karra Taniskidou, Efi, *Machine Learning Repository*, 2017, http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences

[22] Chen, Y., Skiena, S. , *Building Sentiment Lexicons for All Major Languages.*, In ACL (2) (pp. 383-389).

[23] Hutto, C.J. Gilbert, E.E. , *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14).*, Ann Arbor, MI, June 2014.