Departamento de Ingeniería Informática

TRABAJO FIN DE GRADO

Research, development and evaluation of a practical model for Sentiment Analysis

Autor: Jorge Beltrán de la Cita

Director: David Griol Barres

Leganés, Junio 2014

Título: Research, development and evaluation of a practical model for Sentiment Analysis
Autor: Jorge Beltrán de la Cita
Director: David Griol Barres

EL TRIBUNAL

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día .... de .... de ... en ...., en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

VOCAL

SECRETARIO PRESIDENTE

Acknowledgements

First of all, I would like to express my gratitude to my professor David Griol for the opportunity to be involved in Sentiment Analysis field. Moreover, thank him for the support during the development of the project and the guidance and advices in the worst moments.

Secondly, I would like to thank my classmates and all professors I have met during these five years at University, for all those moments that I will remember forever.

Finally, I thank my family and friends for standing by me during this last years. Thanks for your support in the most difficult moments and your advice when things were going wrong. Specially, I thank my parents, my mate Rut, and my friend Javi, for being so constant on backing my decisions and trusting on me in every single moment.

Abstract

Sentiment Analysis is the task of extracting subjective information from input sources coming from a speaker or writer. Usually it refers to identifying whether a text holds a positive or negative polarity. The main approaches to carry out Sentiment Analysis are lexicon or dictionary-based methods and machine learning schemes. Lexicon-based models make use of a predefined set of words, where each of the words composing the set has an associated polarity. Document polarity will depend on the feature selection method, and how their scores are combined. Machine-learning approaches usually rely on supervised classifiers. Although classifiers offer adaptability for specific contexts, they need to be trained with huge amounts of labelled data which may not be available, specially for upcoming topics.

This project, contrary to most scientific researches over this field, aims to go further in emotion detection and puts its efforts on identifying the actual sentiment of documents, instead of focusing on whether it may have a positive or negative connotation. The set of sentiments used for this approach have been extracted from Plutchik's wheel of emotions [1], which defines eight basic bipolar sentiments and another eight advanced emotions composed of two basic ones. Moreover, in this project we have created a new scheme for SA combining a lexicon-based model for getting term emotions and a statistical approach to identify the most relevant topics in the document which are the targets of the sentiments. By taking this approach we have tried to overcome the disadvantages of simple Bag-of-words models that do not make any distinctions between parts of speech (POS) and weight all words commonly using the *tf-idf* scheme [2] which leads to overweight most frequently used words. Furthermore, in order to improve knowledge, this projects presents a heuristic learning method that allows improving initial knowledge by converging to human-like sensitivity.

In order to test proposed scheme's performance, an Android application for mobile devices has been developed. This app allows users taking photos and introducing descriptions which are processed and classified with emotions. Classification that may be corrected by the user so that system performance statistics can be extracted.

Keywords: Sentiment Analysis, Text Analysis, Knowledge Base System, Lexicon, Heuristic-Learning, Android.

Resumen

El Análisis de Sentimientos consiste en extraer información subjetiva de lenguaje escrito u oral. Habitualmente se basa en identificar si un texto es positivo o negativo, es decir, extraer su polaridad. Las principales formas de llevar a cabo el Análisis de Sentimientos son los métodos basados en dictionarios y en aprendizaje automático. Los modelos basados en léxicos hacen uso de un conjunto predefinido de palabras que tienen asociada una polaridad. La polaridad del texto dependerá los elementos analizados y la forma en la que se combinan sus valores. Las aproximaciones basadas en aprendizaje automático, por el contrario, normalmente se apoyan en clasificadores supervisados. A pesar de que los clasficadores ofrecen adaptabilidad para contextos muy específicos, necesitan gran cantidad de datos para ser entrenados no siempre disponibles, como por ejemplo en temas muy novedosos.

Este proyecto, al contrario que la mayoría de investigaciones en este campo, intenta ir más allá en la detección de emociones y pretende identificar los sentimientos del texto en vez de centrarse en su polaridad. El conjunto de sentimientos usados para este proyecto está basado en la Rueda de las Emociones de Plutchik [1], que define ocho sentimientos básicos y ocho complejos formados por dos básicos. Además, en este proyecto se ha creado un nuevo modelo de AS combinando léxicos para extraer las emociones de las palabras con otro estadístico que trata de identificar los temas más importantes del texto. De esta forma, se ha intentado superar las desventajas de los modelos *Bag-of-words* que no diferencian entre clases de palabras y ponderan todas las palabras usando el esquema *tf-idf* [2], que conlleva sobreponderar las palabras más usadas. Asimismo, para mejorar el conocimiento del proyecto, se ha implementado un método de aprendizaje heurístico que permite mejorar el conocimiento inicial para converger con la sensibilidad real de los humanos.

Para evaluar el rendimiento del modelo propuesto, una aplicación Android para móviles ha sido desarrollada. Esta app permite a los usuarios tomar fotos e introducir descripciones que son procesadas y clasificadas por emociones. Clasificación que puede ser corregida por el usuario permitiendo así extraer estadísticas del rendimiento del sistema.

Palabras clave: Análisis de Sentimientos, Analisis de Textos, Léxicos, Sistemas basados en Conocimiento, Aprendizaje Heurístico, Android.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter we describe the precedents of this project, its specific goals and the different phases its planning was scheduled into. Furthermore, the required materials for the development of the project (both hardware and software) are discussed and a detailed project budget is presented.

1.1 Precedents

Nowadays Internet is one of the most popular communication channels. Since it was popularized in 1990s, year by year it has expanded rapidly so that currently 39% of world's population are Internet users. Its huge growth can be explained by the low-cost infrastructures needed to supply basic access. This fact is making possible not only developed countries having high rates of Internet penetration but also developing countries to experience an unexpected growth. Concrete percentages of Internet users can be found in the Table 1.1

Table 1.1: World population with access to Internet

	Per 100 inhabitants								
	2005	**2006**	**2007**	**2008**	**2009**	**2010**	**2011**	**2012***	**2013***
Developed	50.9	53.5	59.0	61.3	62.9	67.3	70.5	73.4	76.8
Developing	7.8	9.4	11.9	14.7	17.5	21.2	24.5	27.5	30.7
World	15.8	17.6	20.6	23.2	25.7	29.5	32.7	35.7	38.8

However, the creation of Internet was much more significant than the birth of another way for providing the same communication possibilities that TV, radio of telephone could offer. The World Wide Web supposed the origin of a new platform with few limitations and allowed developers to make the most from their creativity.

One of the results of Internet new possibilities was the creation by the middle of 1990s decade of the first online communities, such as Geocities (1994) [8] and Theglobe.com (1995) [9] . These early communities were focused on bringing people together in the net around chat-like pages. At the same time as first communities were build, some online stores appeared. This is the case of Intershop, Amazon.com or EBay, which supposed a revolution for the market, since it was possible to buy from home with just some clicks. Moreover, few years later appeared the first social networks, allowing people to exploit other ways of communication by linking friends or meeting people by locations or interests. The creation of these new technologies in such a sort period of time lead to a socialization of the net and therefore many people started feeling attracted by the World Wide Web.

Since the emergence of Sixdegrees.com in 1997 [10] or Friendster in 2002 [11], many other social networks have been developed such as MySpace [12], LinkedIn [13] until these days, where Twitter [14], Facebook [15] or Instagram [16] dominates the market. The creation of these new platforms was the response to a rapid increase in the number of users. which experienced a significant growth: when in 1999 Geocities was purchased by Yahoo [17], it had 19 million users, by the end of 2013 Facebook had 1.2 billion monthly active users [18]. In order to analyse the big picture, Figure 1.1 describes social networking usage since 2005 in the USA. As we can see, the global internet usage has growth from a small 8% of population to an outstanding 73%. Indeed, taking in account young adults from 18 to 29 years old of the latest collected data, we can perceive that 9 out of every 10 people uses social networks.

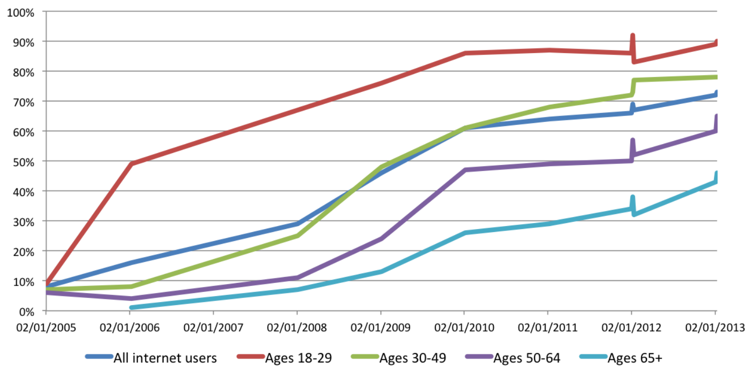


Figure 1.1: Percentage of Social Networks users in USA [3]

However, these numbers are not only related to social networks. In the last years, online retailers have become very popular due to the ease for purchasing from home and the security protocols which have minimized online banking cyber-crime. Figure 1.2 shows that every year more European Union companies are starting to get presence on Internet and currently the number of enterprises selling online is close to 15%. We can also perceive that the percentage of enterprises purchasing online has become steady around 20%.

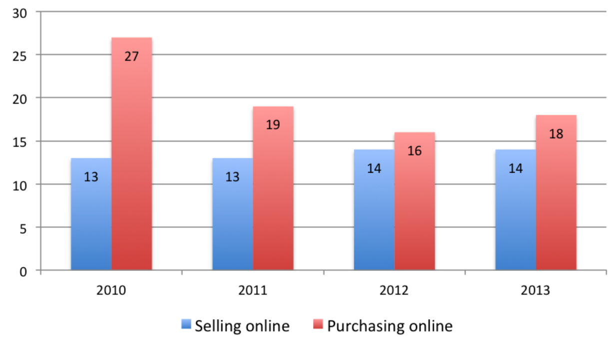


Figure 1.2: Enterprises selling and purchasing online in Europe [4]

Nevertheless, although the number of enterprises selling and purchasing online is not growing rapidly, Figure 1.3 shows that the amount of people buying over the Internet is growing each year in Europe. Actually 38% of European population have ordered some kind of good or service through the Internet. Moreover, if we analyse more mature online markets such as United Kingdom we can see that more than 7 out of every 10 people in the country do online purchases.

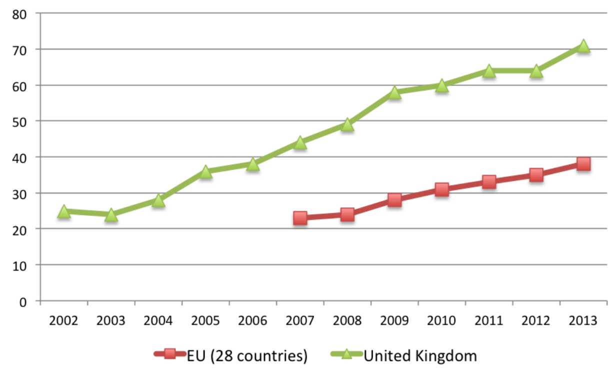


Figure 1.3: People purchasing goods or services online in Europe [5]

All these statistics are showing that the actual retail sales trend is leaning towards online market. Indeed, according to Forrester Research Report [19], European online retails sales will grow 11% a year from 2012 to 2017, when online sales will stand for 248.3 billion dollards compared to 145.6 billion dollars forecasted for 2012 as Figure 1.4 depicts. Furthermore, taking into account the numbers from Figure 1.5 we can see that in USA E-commerce retails sales in 1998 stood for a meaningless 0.19% and by the end of 2012 they represented more than a 5%, depicting again the importance of online stores.

Regarding previous data we can assert that Internet came to be a revolution in almost every aspect of our lives, from social relationships to our economy. This fact has motivated the need for analysing the huge amount of information that is being generated daily. The

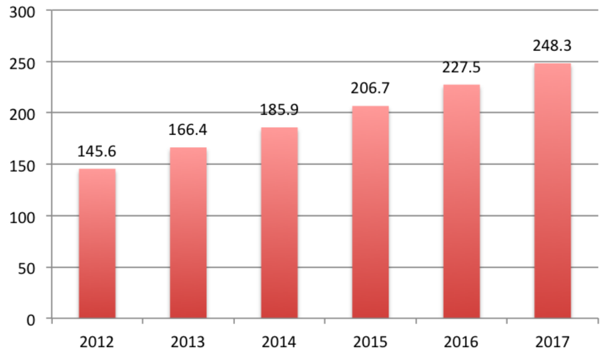


Figure 1.4: European e-commerce forecast from Forrester Research

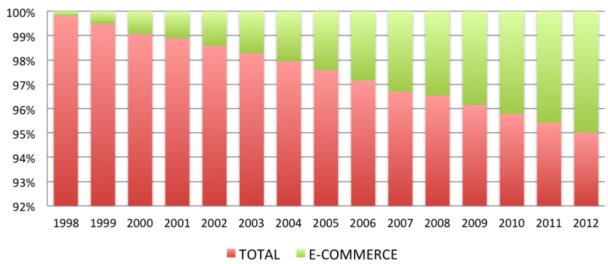


Figure 1.5: US E-commerce Retail Trade Sales vs Total [6]

need to analyse the World Wide Web got its response in the emergence of Text mining when Internet was becoming popular in the late 1990s. Many companies and researchers saw the chance of using text analytic techniques to get tones of information from the Internet. Part of them, such as online stores, were not only interested on collecting high quality data but also on finding ways of extracting affective information from products reviews or client opinions as a way to learn from real market experience. In order to detect and infer emotional information from human input some Natural Language Processing tools were used among other techniques which composed the research field known as Sentiment Analysis [20], which is in charge of extracting human opinion or attitude towards some topic.

Sentiment Analysis usually focuses on extracting the polarity of text, which means recognizing whether they are positive or negative, but can also attempt to detect more complex emotions. There are two main approaches towards this field: knowledge-based systems and machine learning models. Knowledge Base schemes are usually based on lexicons or dictionaries which stores emotional information evoked by words, for instance if an adjective is positive, negative or neutral. This kind of systems performs similarly in every context so that they are very useful for analysing texts referred to new events or concepts, although they may not be so accurate in particular contexts. On the other hand, machine learning approaches usually lies in supervised learning algorithm to classify texts. This classifiers require large annotated training samples before they start classifying texts properly. The advantage of this latter model is the easiness for creating specific-purpose model for concrete contexts.

Nowadays, Sentiment Analysis is commonly used over social networks and online stores. The reason is that using sentiment analysis over user generated information can detect market trends, users polarity respect some specific product or even more complex information such as political reactions to some kind of events, which can determine important decisions about a future investment or how founds should be distributed in over research projects in a technology company. Therefore, sentiment analysis study is the core of this project, which aims to give an overview of the actual state of development and performance of the latest researches, proposes a new model for sentiment analysis trying to identify complex emotions and develops a prototype of a mobile application for its performance evaluation.

The proposed model for Sentiment Analysis tries to combine Knowledge Base advantages with the adaptability features presented in machine learning approaches by having a knowledge base capable of learning from users interactions by applying a heuristic learning algorithm. Moreover, the basic functionalities of knowledge-based systems are extended by trying to detect the most relevant topics in texts using an statistical approach, so that sentiments related to this entities get more weight in global text emotion values.

1.2 Goals

As stated in the previous section, this project has well defined goals framed in the Sentiment Analysis field. The first purpose of this project is the study of the different research and market approaches to Sentiment Analysis. This study includes investigation about the different models proposed to analyse emotions in texts and also motivations to use this techniques in the different scopes where opinion mining is commonly used.

Apart from giving an overview of the actual state of Sentiment Analysis research, the intention of this project is also to extend one of the basic models used in the field. In order to extend this model and with the purpose of innovation, this project will try to broaden the set of sentiments detected, instead of limiting research to identify users polarity towards some specific topic as most systems usually do.

Moreover, the planning of the project includes not only the proposal of a extended model for emotion detection but also to evaluate its performance by taking advantage of social networks to check user agreement with the identified sentiments. To do that a social network prototype for Android devices will be developed and will allow users to introduce some text descriptions for the snapshots they upload which will be later analysed by the implemented model to produce a set of sentiments output. At the end of this process, the user will be able to agree with the proposed sentiments or set the correct ones, helping the analysis algorithm to learn from errors in the latter case.

Last but not least, this project might be interesting from a psychological point of view as it will allow the comparison of different artistic interpretation of same emotions, since all photos uploaded to the social network will be public and users will be able to filter images by the emotions extracted from their associated descriptions.

1.3 Material

This section collects the material required to develop this project:

Hardware

- Laptop
- Android mobile phone
- VPS 4 Cores 8GB RAM 100GB
- VPS 8 Cores 16GB RAM 200GB
- Kimsufi Dedicated Server

Software

- ADT IDE
- Eclipse IDE
- MySQL Workbench
- ConceptNet5
- Open Xerox Morphological Analysis
- Apache Tomcat
- Apache Solr
- Texmaker
- BibDesk
- Adobe Photoshop CS4
- Adobe Illustrator CS4
- Visual Paradigm
- Gantt Diagram Maker

1.4 Planning

This project was divided in several phases during the initial planning. The tasks composing each of these phases are described below:

Analysis
This phase is intended for collecting and analysing information related to Sentiment Analysis field, such as emotion classifications, lexicons, semantic networks, etc. Main approaches to the field are studied to determine the actual state of research in SA. Moreover, research papers are analysed to see performance comparison between different SA methods and most common challenges are identified.

Design
The tasks accomplished within this phase are two: definition our model and design of the mobile application. Information gathered during the study of research projects in the Analysis phase is used for defining the structure of the model. Moreover, both application structure and visual layouts are designed in this phase.

Development
This phase can be divided into three sub-phases.

Knowledge Base construction
Building the Knowledge Base is the most important task of the development phase. This task is focuses on determining and storing the emotional information associated to concepts and some other features required for sentiment analysis.

Sentiment Analysis scheme
After building the KB, the required functions to perform analysis of texts are implemented during this phase. The developed functionalities include parsing texts, determining most relevant topics and final computation of text sentiments. Furthermore, the required functions to access the knowledge are developed.

Server-side application
Tasks composing this phase are related with the development of the application logic and implementation of the required API, so that mobile application can easily access server logic through a well defined communication protocol. Application logic is integrated to call Knowledge Base functions when required in server operations.

Android app implementation
During this phase, Android application prototype is implemented from scratch. Views and logic are developed and application is integrated with the server API.

Evaluation
This phase is accomplished from two points of view. Firstly a objective analysis of

model performance is executed based on data gathered from Android prototype. Secondly a subjective evaluation is carried out through measuring user agreement and disagreement with several aspects of the system.

Documentation

The last phase of the project is reserved for documentation. During this phase, project's report is generated in order to describe in detail the work done in the rest of phases. Moreover, project presentation slides are created.

Once different phases and tasks into which the project is divided have been established, the temporal planning is developed in order to assign tasks a specific order and time estimation. In order to do that, a Gantt diagram is created with the purpose of having a better understanding of project phases. Moreover, Gantt diagram allows having a good tool for tracking project process and meeting deadlines.

Considering an exclusive dedication of 8 hours a day 6 days a week, the resulting Gantt diagram is presented in Figure 1.7.

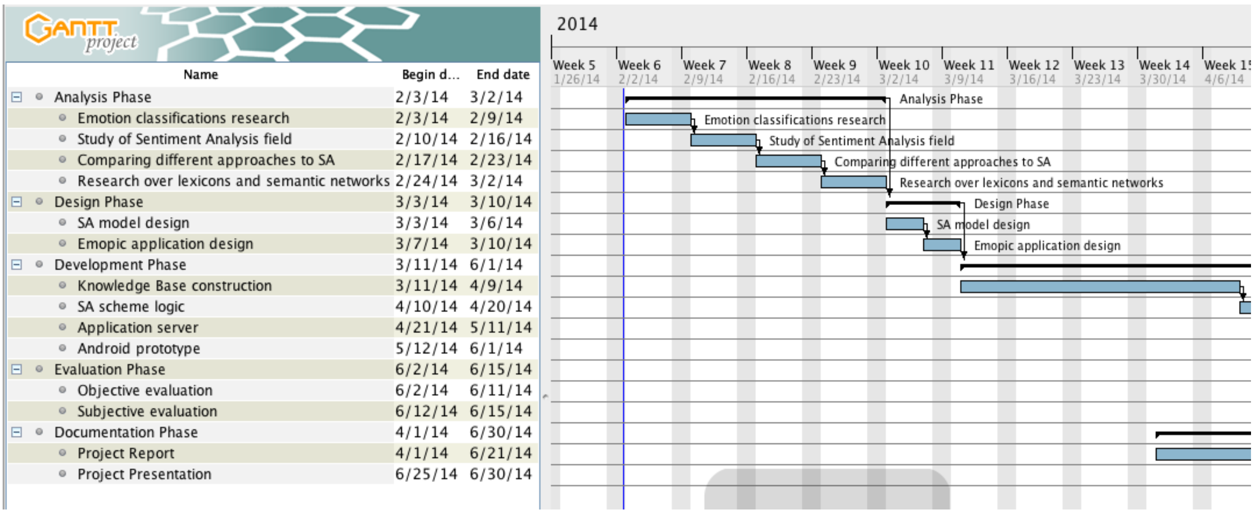


Figure 1.6: Gantt diagram of temporal planning of the project - Part 1

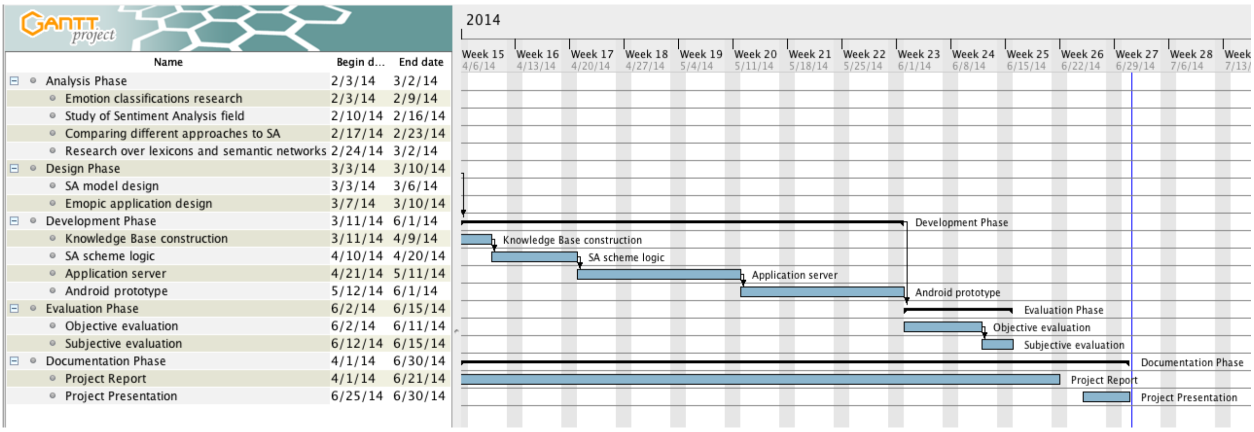


Figure 1.7: Gantt diagram of temporal planning of the project - Part 2

1.5 Budget

In this section the budget for the described project is described. Different phases duration and both technical and human costs are depicted before the final budget is presented.

1.5.1 Tasks

Analysis

- Emotion classifications research
  Duration: 7 days
- Study of Sentiment Analysis field
  Duration: 7 days
- Comparing different approaches to SA
  Duration: 7 days
- Research over lexicons and semantic networks
  Duration: 7 days

Design

- SA model design
  Duration: 4 days
- Emopic application design
  Duration: 4 days

Development

- Knowledge Base construction
  Duration: 30 days
- SA scheme logic
  Duration: 11 days
- Application server
  Duration: 21 days
- Android prototype
  Duration: 21 days

Evaluation

- Objective evaluation
  Duration: 10 days
- Subjective evaluation
  Duration: 4 days

Documentation

- Project Report
  Duration: 82 days (while shared with other phases additional 4 hours were dedicated to documentation phase). Therefore only 6 days of full dedication added to the 82 half-time days gives a total dedication of 47 days.
- Project Presentation
  Duration: 6 days

1.5.2 Resources

Here are presented the cost details of material listed in Material section of this chapter together with the specific costs of human resources.

Hardware resources

- Laptop. Cost: 1200 €
- Android mobile phone 500 €
- VPS 4 Cores 8GB RAM 100GB. Cost: 15 €
- VPS 8 Cores 16GB RAM 200GB. Cost: 60 €
- Kimsufi Dedicated Server. Cost: 5 €

Software resources

- ADT IDE. Cost: 0 €
- Eclipse IDE. Cost: 0 €
- MySQL Workbench. Cost: 0 €
- ConceptNet5. Cost: 0 €
- Open Xerox Morphological Analysis. Cost: 0 €
- Apache Tomcat. Cost: 0 €
- Apache Solr. Cost: 0 €
- Texmaker. Cost: 0 €
- BibDesk. Cost: 0 €
- Adobe Photoshop CS4 Trial version. Cost: 0 €
- Adobe Illustrator CS4 Trial version. Cost: 0 €
- Visual Paradigm Student Edition. Cost: 0 €
- GanttProject 2.5. Cost: 0 €

Human resources

Taking into account that the average cost of a computer engineer in Spain: 35 €/hour a summary with the human resources costs is shown in Table 1.2.

Table 1.2: Human resources cost summary

Phase	Days	Price
Analysis	28	7,840€
Design	8	2,240€
Development	83	23,240€
Evaluation	14	3,920€
Documentation	47	13,160€
Subtotal	180	50,400€

1.5.3 Final Budget

The final costs of the project are detailed in Table 1.3.

Table 1.3: Final Project Budget

Concept	Price
Software Resources	0€
Hardware Resources	1,780€
Human Resources	50,400€
Subtotal	52,180€
21% IVA	*10,957.80€*
Total	**63,137.80€**

Chapter 2

State of Art

In this chapter the context in which this project is developed is discussed. Firstly, the main different emotion classification models are presented and compared. Subsequently the two main research fields in charge of detecting emotions in human input: Affective Computing and Opinion Mining are described in details to give an overview of the main ways of analysing information gathered from humans. Afterwards, in order to approach the topic of this project the principal Sentiment Analysis schemes such as Latent Semantic analysis, Bag-of-words, Support Vector Machines are depicted. Once main approaches have been described this chapter will focus on the one chosen for this project: bag-of-words model. Last but no least, different alternatives for each of the modules composing this scheme will be discussed and compared to motivate the decisions taken during the project.

2.1 Classification of emotions

Since Charles Darwin stated that emotion were defined biologically and they were common to all humans in 1872 [21] , many anthropologists and psychologists began to study how people from different cultures expressed emotions. This research process lead to the most popular theory in the 1950s that believed on the opposite idea: emotions were the result of cultural contexts in learning. Based on this new belief, the American psychologist Paul Ekman found that there were many similarities between cultures on defining which emotions describe common facial expressions. Concretely Ekman discovered through his studies that there were six basic universal emotions: anger, disgust, fear, happiness, sadness, and surprise. However, he also found that cultural prescriptions were an important limiting factor for the expression of emotions in some specific contexts. Later in the 1990s, Ekman extends his set of basic emotions to include some other emotions that were not necessary represented by facial expressions.

Based on Ekman studies, many other psychologists proposed different models for representing emotions into one or more dimensions. Some of this models are described in the following sections.

2.1.1 Circumplex model

In 1980 the psychologists James A. Russell published a Circumplex model [22] *"as a way psychologists can represent the structure of affective experience and as a representation of the cognitive structure that laymen utilize in conceptualizing affect"*. The most important differentiation from htis model to others is that Russell stated that unlike the common descriptions of emotions in separate dimensions independent one from the others, the affective dimensions were related by a function of **pleasure-displeasure** and **degree-of-arousal** as shown in Figure 2.1. In order to prove his theory he scaled 28 adjectives expressing emotions in four different ways including similarity between terms and analysis with several subjects self-reports of their emotion states.

Circumplex model uses a bi-dimensional representation being pleasure (or valence) attached to the horizontal axis and arousal to vertical axis. The origin of the coordinate system represents a medium arousal and a neutral valence. Distribution of emotions describes a circular pattern centred in the origin. This model has been applied in researches about stimulation based on words and on emotional facial expression such as an study on *Machine vision based recognition of emotions using the circumplex model of affect* in [23]. The valence-arousal pair is also used in Vector model [24], although in this case the arousal is an underlying dimension an vector direction is defined by the value of valence, which can lead to a positive value (shift top the vector) or negative (shift down vector).

The main difference between both models is the range of emotions they can represent. In circumplex model can represent high intensity emotions with neutral polarity, however vector model holds that neutrality cannot be felt with intense emotions.

2.1.2 Plutchik's model

Yet in 1980, professor Robert Plutchik developed his famous theory of emotion [1]. His study defined that there are eight basic emotions biologically primitive related with survival and reproduction in animals and humans. The eight emotional states were divided in four pairs of opposites emotions: anger versus fear, sadness versus joy, disgust versus trust and surprise versus anticipation. Therefore he differentiate four affection categories that can represent independent opposite primary emotions in different degrees of intensity. Moreover, he stated that all other affections, known as complex emotions, are combinations of the eight basic ones. Plutchik's model was illustrated with *The wheel of emotions* shown in Figure

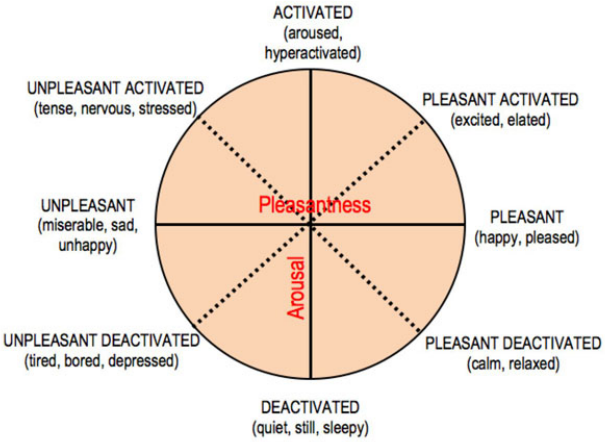


Figure 2.1: Emotional representation in Circumplex model

2.2, which is the 2D version, and in a cone-shaped 3D representation. Each pair of contrary emotions are located in opposite places in a wheel of eight spokes. Furthermore each basic emotion is mapped to a colour. Each of the eight colours are drawn with higher intensity as they get closer to the center of the wheel, representing this way stronger emotions. As long as you look further from the center the colours get softer representing less powerful emotion states. Combinations of contiguous spokes produce the eight advanced emotions that Plutchik proposed: aggressiveness, contempt, remorse, disapproval, awe, submission, love and optimism.

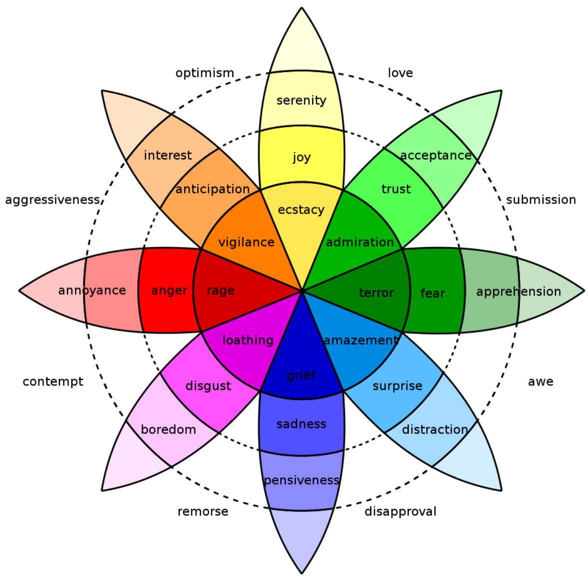


Figure 2.2: Plutchik's wheel of emotions

2.1.3 PAD emotional state model

The PAD model was developed on 1997 by Albert Mehrabian and James A. Russell based on Mehrabian's previous work [25]. and Russells notions from Circumplex model described in Section 2.1.1 . They proposed a representation of the different emotions in a three-dimension-space 2.3 . The dimensions are Pleasure, Arousal and Dominance and is commonly used for the study of non-verbal language in psychology. The *Pleasure-Displeasure* scale is usually limited to a set of 16 possible values. *Arousal-Nonarousal* scale measure intensity like in Russell's first model which is restricted to 9 specific values. Eventually, the new dimension introduced in this model, *Dominance-Submissiveness* scale, represents whether it is a passive or active emotion. For instance, being anger and fear both unpleasant emotions, anger is dominant and fear is submissive. PAD model is commonly used for marketing purposes for measuring customer emotions towards products and extract conclusions on how having positive intense emotions affects customer spending and shopping time. Apart from marketing, PAD is also used to map emotions from PAD Space to facial expressions of characters in order to show basic and complex emotions or assigning emotions to avatar faces.

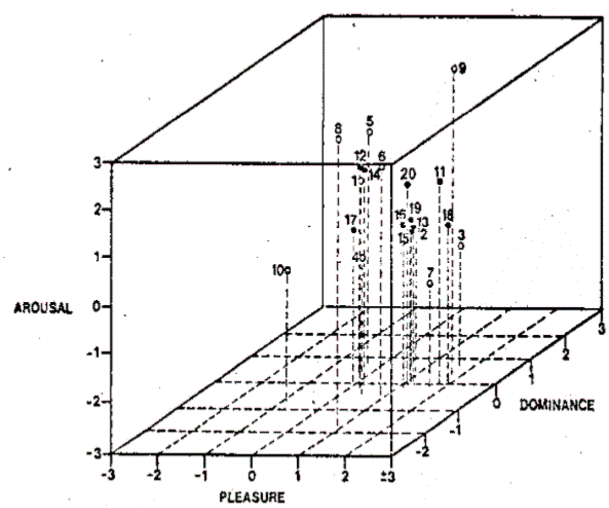


Figure 2.3: Emotional representation in PAD model

2.1.4 Hourglass

Hourglass model [26] is an affective 3D representation based on Plutchik's theories 2.1.2. The model was proposed to build an alternative for the already developed emotional classifications which fit better the requirements of opinion mining applications. It is inspired on the idea that there are four independent dimensions that can represent any human emotion. This four categories, known as *Pleasantness*, *Aptitude* , *Attention* and *Sensitivity*, are mapped to biological mind resources that may be activated or deactivated to compose the different emotional states. The biological meaning of activating resources is usually related to help our body to react properly to the current emotion. For instance, trust emotion will power the relaxation of our body and suppress other resources that may prepare ourselves to a dangerous situation.

Each of the dimensions can take six different levels of intensity [-3, +3] to compose a total of 24 basic emotions, which are the ones proposed by Plutchik's wheel of emotions, being each of them expressed in three levels of strength. Like in Plutchik's model, complex emotions are the result of combinations of categories, which means activation of several resources at the same time, e.g. 'love' will be composed of a positive value for *Pleasantness* and *Aptitude* categories. The final distribution of primary emotions in the 3D model is shown in Figure 2.4.

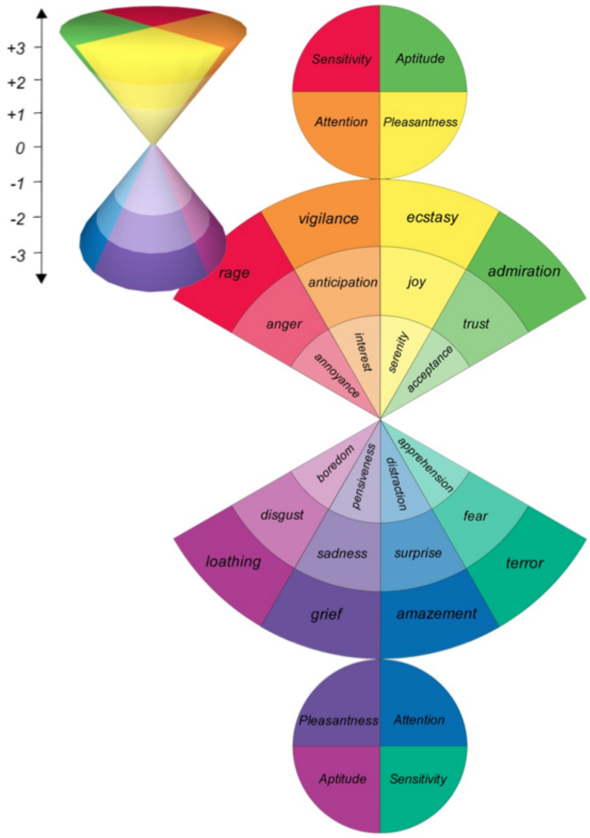


Figure 2.4: The Hourglass of emotions

2.2 Affective Computing

In the previous section the principal models for emotions representation were analysed. In this section one of the most important research fields associated to human sentiments is depicted. Affective Computing is defined as the study and development of computer systems which can recognize, interpret, process or simulate emotions. Affective Computing is a combination of computer science, engineering and psychology and some other science related to cognition and education. This field was coined by Professor Rosalind Picard in 1995 [27] as a response for the recent neurological studies which suggested that human emotion was not trivial in cognition processes, but an important factor for thinking, making decisions and almost every interaction situations with other people or the environment surrounding us. Her studies were aimed to detect and interpret human emotions to build better computer systems capable of providing better interaction with humans and taking better decisions by adapting computer behaviour to emotions perceived from people.

Affective Computing is applicable from two different perspectives: recognizing emotional states in humans and showing emotions in machines. The first area consist in gathering information from humans through sensors for latter recognition of patterns which can be mapped to specific emotional states. Sensors can retrieve many kinds of information such as temperature, facial expressions, voice, heart rate or sweating level. Once the information has been collected, it is time to apply machine learning techniques to extract desired meaningful output which will be mapped to an emotions representation model so that computer can calculate the specific emotional state. The second perspective deals with designing computer systems capable of simulating human emotions. The emotional states showed by machines are usually related to a learning process from human interaction instead of biological factors in human emotions, and are already limited to a set of predefined states.

Since these two perspectives, there are many computing technologies which are associated to this research field, such as detecting emotions from speech, facial expression, physiological signs or visual aesthetics. Emotional Speech takes advantage of biological changes in body that affects speech to recognize speech features variations and extract emotional states. Pitch, intensity, formants and speed are examples of features analysed in this kind of affection recognition. It is proved to be a reliable method having an average recognition success of more than 80% into negative and positive emotions [28], and even a higher accuracy rate (about 89%) in other studies such as Tomas Pfister and Peter Robinson's work in 2010 [29] which classified speech into nine classes from the Mind Reading emotion corpus.

Other technology is Facial affect detection which analyses contraction and relaxation of muscles to infer subject current emotions. Combinations of these muscle actions compose the definition of the physical expressions of emotions. This mapping between emotions and facial muscles action was created in 1978 by Ekman and Friesen and is called the Facial Action Coding system [30]. This technology has some issues such as the need for very pronounced expressions for proper emotion recognition and also the lack of adaptability to head rotation, which leads to a decrease in success rate. In order to achieve better results this technology is often combined with hand gestures recognition.

Hand gestures are just a possibility of the body gesture technology whose aim is to analyse reflexive and concious human gestures to identify emotions. The most expressive parts of human body are indeed hands, but there are other meaningful elements such as shoulders where important information can be gathered in order to create recognized patterns that express emotions. There are two main approaches in this field. Firstly, a 3D model is used to collect information from some parts of the body to extract meaningful features. The second model was proposed in 1997 [31] and is based on apparentness, and makes use of pictures and videos for affection recognition.

Last but not least, measuring physiological signs is the latest approach for emotion recognition. This field is called Physiological monitoring and it has been proved to be a high reliable method. The three main signs which are usually monitored are Blood Volume Pulse, Galvanic Skin Response and Facial Electromyography. The latter sign has been used in recent studies by Petrantonakis and Hadjileontiadis[32]. The results of this approach are promising with a performance of 83.33% well classified emotions by using Support Vector Machines (SVM) and an affection set composed by Ekman's basic emotions, and getting closer to perfect classification when using less than five categories.

Therefore, research projects on Affective Computing are proving that detecting emotions in humans through the collection of data from human inputs is not only possible, but also accurate enough to outperforms human capacity in most of the technologies described above. In the next section the focus is put on Opinion Mining, which is another research field associated with emotion detection although these studies are related to human inputs in text-form.

2.3 Sentiment Analysis

Sentiment Analysis (SA) -also known as Opinion Mining- is the field in charge of extracting emotional information from human texts by using natural language processing tools, text analysis and computational linguistics. The task is usually focused on identifying the polarity of a text or the emotions of a subject towards some topic. The basic sentiment analysis algorithms try to determine whether a piece of text is positive, negative or neutral (called polarity of a text). However, there are also advanced approaches that aims to identify emotional states based on some affection models.

The principal applications of opinion mining are related to marketing and social networks. Marketing main purpose is using SA for determining the attitude of customers towards products. It became a great source of valued information when the first online stores were launched and customer reviews popularized. By applying Sentiment Analysis techniques to customer opinions companies can gather information about products acceptance and market trends. Therefore opinion mining is nowadays a key point for market decisions and allow enterprises to offer products adapted for clients requirements and focus innovation on most demanded features.

Moreover, as social networks started growing in the beginning of the 21st century they became one of the most important sources of information for many people and also a great place for sharing thoughts about *trending topics*. This fact has turned very attractive the possibility of analysing the huge amount of information generated every day on this social networks to gather users sentiments towards important event and extrapolate opinions to the whole society, provided that internet penetration is growing rapidly. Conclusions extracted from the analysis of this information are not only used by brands to know their popularity like with product reviews analysis, but also by other media or governments to figure out political reactions. For instance there are many studies focusing on *Twitter* as it is one of the preferred tools by internet users for sharing opinions and therefore constitutes a huge corpus for researches. An example of this research is described in [33], which aims to detects polarity of tweets overcoming other studies limitations when analysing, for instance, sarcastic tweets by applying some pre-processing before sending text to classifier.

Although the main uses of sentiment analysis have been described above, there are different levels of classification. The first one is classification by *polarity* whose main goal is product or movie reviews. Pang's work in 2002 [34] is an example of these studies. Other way of classifying was identifying whether texts contains subjective information, or in the contrary they are objective. Therefore two different classes are used: subjective and objective. The main challenges of this model are the agreement over a definition for subjectivity, and the fact that objective text may contain subjective sentences. This classification method is usually a harder problem than polarity one. However, Wiebe and Riloff work [35] results showed an accuracy around 70%.

The last analysis level is the finer one: feature-based sentiment analysis. The two main concepts around this model are features and entities. Entity stands for any word or predicate acting as a noun who is the target of sentiments. Features, on the contrary, refer to any aspect or attribute associated to an entity. The harder tasks within this model are identifying main entities, detecting their associated features, and extracting emotions from each of the features. Recent studies of this model have been applied to *Twitter* [36]. Feature-based sentiment analysis model is the one used in this project.

2.4 Approaches to Sentiment Analysis

Along this section the two different approaches towards opinion mining will be described. Moreover, some of the machine learning specific techniques will be presented for further contextualization.

2.4.1 Knowledge-based systems

The first approach are Knowledge-based systems (KBS). Knowledge base is often used for storing semantic information. On the other hand, inference engine is in charge of identifying which emotions are expressed on text by applying the set of predefined rules. There are many publicly available lexicons about subjective concepts that can be used for the process of building the knowledge base, such as WordNet-Affect [37] or SenticNet [38]. These lexicons will be presented in Section 2.5 so long as this project makes use of some of them. However, knowledge-based systems are usually used together with other tools for improving overall performance of sentiment analysis algorithms. The main advantage of this approach is that it performs similarly in almost any context, provided that a complete knowledge base is used. However, the lack of training makes difficult improving performance as problem may also be caused at knowledge base construction stage and therefore harder to solve without adapting the entire inference engine to support new rules.

2.4.2 Machine learning

Machine learning approaches to sentiment analysis are based on applying known machine learning and natural language processing tools over text for emotion recognition. There are many different methods varying on sophistication. Here below some of them are depicted.

- *Latent Semantic analysis* is based on the theory that similar-meaning words occupy same places on text. From this idea LSA technique analyse relations of documents searching for terms similarity. An occurrence matrix is produced to represent terms and the number of times they appear in documents. Later this matrix is simplified so that it becomes computationally feasible to make calculations as in case the set of documents is too large, the occurrence matrix may exceed computer resources in further computations. Therefore, the matrix is simplified and terms are compared by taking their associated vectors. Then documents can be clustered based on the terms they contain.
- *Support Vector Machines*, contrary to LSA, are supervised learning models. An annotate dataset is needed for training the algorithm, which represents examples as points

in a two dimensional space. In Sentiment Analysis this set is composed by text with their appropriate associated emotions (usually polarity for linear classification). The algorithm calculates the hyperplane which better split the texts providing the highest available margin between the two categories. New examples are then evaluated so that the algorithm predicts which category should the example be mapped to.

- *Bag-of words* uses occurrences of words in documents as features for classification. The *bag* of words refers to the different words contained in a document or set of documents. A bag of words is created for each of the categories into which our algorithm aims to classify documents. For each example document that is to be classified, a dictionary of terms present in the document is build. A numerical value representing the number of times the word appears -called *term frequency*- is associated with its corresponding dictionary entry. Afterwards a term weighting algorithm is used to assign weights to document words and the a probabilistic model is applied for classification comparing document dictionary and the different bag-of-words.

- *Holder - Target* is one of the most advanced sentiment analysis approaches. The main challenge of this approach is computing properly the direction of sentiments so that holders and target are not confused. This is usually achieved by pattern matching. Once holder and target are recognized, emotion classification of documents can be determined by assigning a higher weigh to sentiments associated to more frequent targets and then applying any classification algorithm.

2.5 Sentiment-word Dictionaries

As stated before in 2.4.1, along this section some of the main lexicons publicly available for English language will be described. These lexicons are usually used for building Knowledge-bases which, indeed, is the case of this project.

- *WordNet* was published in 1995 by G.A. Miller [39]. It is the result of automatically analysing a huge amount of information published on the Internet. WordNet was released in 2006 and is defined as 'an online lexical database designed for use under program control. English nouns, verbs, adjectives, and adverbs are organized into sets of synonyms, each representing a lexicalized concept. Semantic relations link the synonym sets.' [40]. Therefore it can be used for semantic disambiguation by looking at the proximity between words in the whole net.

- *Sentiwordnet* was developed by A. Esuli and F. Sebastiani in 2006 [41] . Their authors describe this project as a lexicon that takes WordNet synset and associates to each of the term three numerical values for describing objectivity, and how positive and negative they are. The assigned values are the result of applying several supervised classifiers to the glosses or definitions associated to each of the synsets.

- *SenticNet* lexicon was created at the MIT Media Laboratory in 2009. It was build upon 14,000 concepts extracted from ConceptNet common-sense networks, which is depicted in Section 2.6. Semantic and emotion information is assigned to concepts by querying this network. Since it was published SenticNet has evolved rapidly building lots of tools and techniques around it to conform Sentic Computing [42], which has become an interdisciplinary approach for sentiment analysis.
- *NRC emotion lexicon* was built by using Amazon's Mechanical Turk, which allows researchers from the NRC in Canada to get human annotations about emotions associated or evoked by their proposed synset, getting more than 95% of correct assignments. S. Mohammad and P. Turney work[43] uses Plutchik's wheel of emotions as the emotion representation model and has been used in recent projects such as Transprose [44] which tries to create a soundtrack based on sentiment analysis of novels.

2.6 Semantic networks

Apart from the use of dictionaries for building Knowledge bases, there are many studies suggesting and exploiting the possibilities of building new KBs with information extracted by querying semantic networks. For instance Tsai work on building sentiment dictionaries based con commonsense networks support the use of ConceptNet for this purpose assuming semantic relations within terms implies sharing a common sentiment [45]. Within this section some of the most remarkable semantic networks are presented.

- *Lexipedia* [46] is an online visual-based dictionary and thesaurus. This semantic network depicts words and their semantic relationships (fuzzynyms, synonyms and antonyms) in centered graph which can be queried online for free. This project supports six languages including the most popular European ones.
- *ConceptNet* [47] is a hypergraph that represents concepts of real world as nodes and labels relations between them. These concepts and relations are generated by analysing the World Wide Web content applying natural language processing tools and machine learning techniques. ConceptNet contains information and relations about basic knowledge such as 'dogs and cats are mammals', scientific knowledge, e.g. 'Java is a programming language' or information about latest news. The main sources for building this semantic network are dictionaries, wikis, wiktionaries and other research projects. It is developed and maintained by MIT researchers.
- *ProBase* [48] is built by searching World Wide Web sites and logs to gather information. There are four principal elements composing ProBase: concepts, instances, attributes and relationships. Concepts are classified into 2.7 million categories and there are not only strict relationships, but also probabilistic values are assigned to links between

concepts in order to reflect, for instance, similarity. It is developed by Microsoft and it is not publicly available yet.

After detailing the context of our project, the fact that the proposed model combines most of the different techniques or components described above may leads to think that its complexity can be detrimental for the overall performance. However, taking the essential advantages of each of these techniques may balance the disadvantages observed in the simple approaches towards Sentiment Analysis research. Building the project's knowledge based both on lexicons and semantic networks provides the possibility of composing a faithful representation of concepts' emotions. Moreover, allowing the knowledge to learn based on its accuracy guarantee that initial generated knowledge's errors will be corrected during training period. Last but not least, detecting topic relevance inside texts give the algorithm the ability for understanding which emotions are more important with respect to the overall message of the text.

Chapter 3

Sentiment Analysis Algorithm

In this chapter the proposed sentiment analysis model is presented. The main components composing the scheme -knowledge base, analysis module and scoring system- are described in details in terms of architecture and functionality. Furthermore, the specification of the learning algorithm will be discussed presenting the method used for improving the overall performance of the proposal.

3.1 General description of proposed model

The proposed model for sentiment analysis is a knowledge-based system that uses annealing for improving the actual knowledge. It aims to extend common sentiment classification of text, which is usually focused on polarity, to a higher level so that texts are categorized by the emotions they evoke. This means that instead from detecting whether a piece of text is negative, neutral or positive, the goal is specific human emotions recognition. For this purpose, a limited set of emotions have to be selected from one of the existing emotion classifications accepted by psychologist community.

After a strict study of the principal affective models -described in Section 2.1- and considering computational requirements the selected model is a modification of *Hourglass emotion representation*. The main reason of the choice is the fact that this model is based on Plutchik's wheel of emotions which proposes eight basic emotions contrary to Ekman's initial classification that defines only six primary affection states. However, although having more categories increases analysis complexity, plutchik's model can be reduced into four categories -as there are four pairs of opposite emotions- so that, indeed, the analysis can be considered to turn out simpler. Hourglass model represents these four categories into four intensity scales with well-defined boundaries [-3,+3] where negative and positive values of the same categories are mapped to opposites emotions. Therefore, although algorithm becomes simpler, this scheme also allows a more comprehensive representation than Ekman's one, by extending the total number of emotions. Moreover, having a bounded set of acceptable values simplifies the storage design and results evaluation.

Regarding overall architecture, the proposed model is based on four key elements. Knowledge is the first module, two of them are related to analysis process and the fourth one is associated with adaptive learning for improving performance. Knowledge Base is the source of information used by the analysis module to extract sentiments values from words. It is the most important element of the model as its adaptive nature guarantee refinement with training provided that there is not malicious feedback.

Secondly, analysis module is in charge of word analysis. By splitting texts in sentences an tokenizing words this module can query the Knowledge Base for extracting emotional information or whether words are modifiers or carry negation. Moreover, this module identify entities that appeared in the text and keep track of the number of occurrences of each in a similar way bag-of-words models do by using occurrences vector.

Once the entities have been identified and words are annotated with values from KB module, the Scoring module computes overall text entities relevance and assign a weighting factor for each of the words carrying emotional information -also known as *concepts*. After detecting and assigning concepts a weight for each of the four independent emotional categories the whole text emotions are computed and therefore text is classified.

The last stage of the model deals with knowledge learning issue. As the model proposed a fixed equation for computing text emotions there is a need for improving algorithm automatically so that results are not exclusively dependant on Knowledge Base quality. In order to achieve that, the learning module takes as input the provided analysis from users when they disagree with the results of the sentiment analysis, and computes a learning factor to modify sentiment values of involved concepts. Assuming users introduce correct emotional information about their own texts, this module compares the differences between the initial output and user's data and compute specific modification for each of the categories which user disagrees with.

However, in order to provide a detailed description of the model the different modules are depicted separately in the following sections.

3.2 Emotion classification model

As stated before, this project uses a modified version of *Hourglass* as its emotion representation model. The justification for the choice are given in previous section. However, it is required to explain what kind of changes have been applied and the reasons to it. As M. Koppel and J. Schler asserted in their study about neutral categories [49], that the inclusion of a neutral category on a polarity problem can improve the overall performance of the system, provided that not everything in the world is black or white, but instead there are intermediate states. Thus, having a look at Hourglass model definition a clear conclusion can be extracted: the only way a text can be classified as neutral relies on getting all four independent categories null value, which is hardly possible.

In order to cope with this problem and considering there are many objective texts electable for being classified as neutral, we found defining a neutral threshold around zero value a valid solution for increasing chances of neutral category. As a consequence, Hourglass scale is displaced by the threshold in both negative and positive directions, but assigning the same range sizes for the seven -latter six levels of intensity of a category plus the neutral threshold- levels of intensity. Figure 3.1 depicts how the different four categories of *Hourglass* model will look like with after this modification.

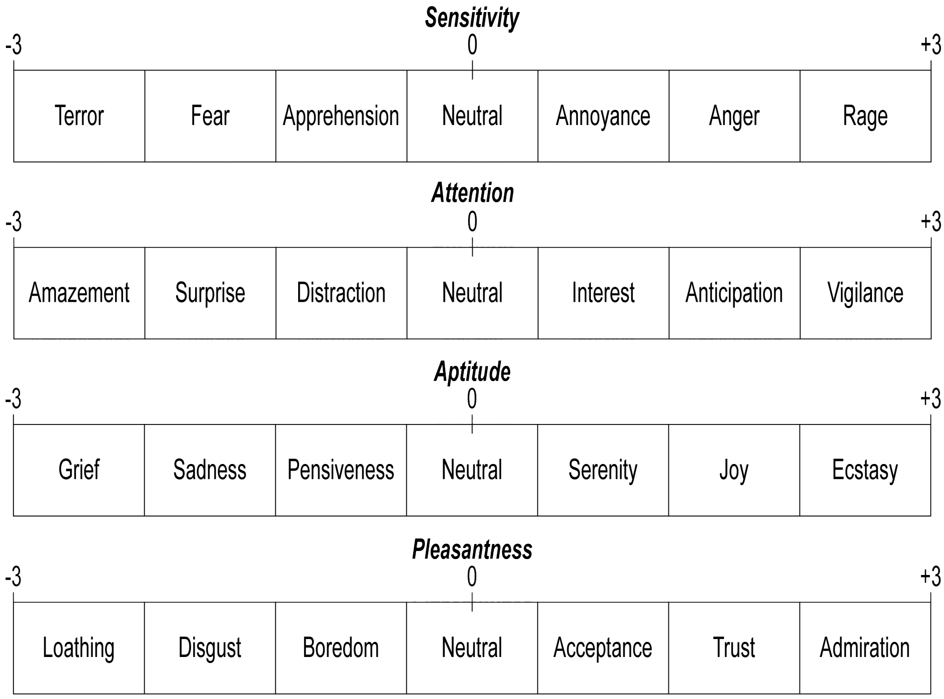


Figure 3.1: Hourglass scales modified to add a neutral level of intensity

In conclusion, bearing in mind the changes applied to *Hourglass* representation, the four independent categories which are considered for sentiment analysis in the proposed model would be composed of the following possible labels, described from negative maximum to positive maximum intensities, left to right:

- **Sensitivity:** [terror, fear, apprehension, *neutral*, annoyance, anger, rage]

- **Aptitude:** [amazement, surprise, distraction, *neutral*, interest, anticipation, vigilance]
- **Attention:** [grief, sadness, pensiveness, *neutral*, serenity, joy, ecstasy]
- **Pleasantness:** [loathing, disgust, boredom, *neutral*, acceptance, trust, admiration]

Nevertheless, for the purpose of better understanding, references within the project to positive and negative sentiments associated to each category are not referred by their specific label associated to the degree of intensity but with the medium intensity label as a generic label. For instance, positive sentiments in *Attention* category are referred as *Joy* independently of the positive intensity and negative sentiments of *Pleasantness* category are called *Disgust*. Therefore although the degree of intensities are taken in account in computations, terminology is simplified.

3.3 Knowledge Base

Considering the proposed model is defined as a Knowledge base system, it is understandable this element of the scheme holds the higher responsibility on the final output. In this project the goal of the Although common Knowledge bases are built upon ontologies, the specific requirements of the problem that is to be solved in this projects make possible the use of a simple database for representing our knowledge.

The two type of elements required for the proposed model are described here below:

Concepts A concept refers to the emotions associated to a specific pair of *word, part of speech*,being Part of speech (POS) is the grammatical function of a word inside a predicate. Words case is a special one. Instead of having all derivatives of each word, only primitive form are considered for knowledge, and rest of derivative words takes the same emotional values. The concrete details for each word category are described here:

Nouns Only singular form is considered, although they may have an irregular plural which could be harder to identify. Nouns containing prefixes and suffixes are the only exception.

Adjectives Positive form is taken and both comparative and superlative forms are discarded.

Verbs Infinitive form is the one stored in the Knowledge Base. Some exceptions are made for *-ing* forms acting as nouns such as *reading*, e.g. "The professor's reading about macro-economics was brilliant".

Adverbs Like adjectives, only positive form is considered discarding comparative a

Modifiers *Modifier* is an *n-gram* without associated sentiment states that may vary concepts meaning by either increasing, decreasing or reversing the emotions they evoke. They can be divided into two different categories:

Intensity modifiers This category is composed by those modifiers than may increase or decrease concepts' expressed emotions. For instance *n-grams* such as *much* or *a bit* increase and decrease respectively the affection of concepts they are associated with.

Negators There are some modifiers that instead of varying the intensity of emotions, they reverse the global affection of a concept such as an antonym would do. These words are called *negators* and *not* or *never* are representative examples of this set.

In order to create Knowledge Base as complete as possible for Sentiment Analysis purposes some sources of concepts were selected considering the different alternatives described in Sections 2.5 and 2.6. The two different approaches followed to build this project Knowledge Base are depicted in the next subsections.

3.3.1 First approach

On of the main challenges when building a Knowledge Base for Sentiment Analysis if choosing the proper sources of information that grant a complete and accurate representation of the world you are trying to describe. Within this project the only part of the world that needs to be represented is the set of different emotions that concepts evoke. For this purpose and considering Tsai studies [45] the first approach followed in the construction of the KB was to select a good dictionary from the publicly available ones so that entries can be used to query a semantic network in order to extract information about sentiments associated with them.

From dictionaries listed in Section 2.5, WordNet is probably the most complete English dictionary available. However, many of the concepts included in WordNet are objective and therefore are not required to be considered in our knowledge as they do not carry any emotional information. From the other three options, we found that both SenticNet and NRC lexicon are great dictionaries containing already annotated entries with the required affective information but no *part-of-speech* information is provided. Finally, we find SentiWordNet being a project based on WordNet but limiting the set of included concepts to those which evokes some kind of emotion or polarity. Consequently, although it does not include any annotation for emotional categories we have defined in our emotional representation model, SentiWordNet provides POS tagging, differentiating same words in several entries depending on the amount of POS that can be associated to them, so it turns out to be the perfect choice to start querying a semantic network.

Although semantic or common-sense networks contains huge amounts of interconnected information and usually support several languages, they are not capable of storing all linguistic derivations of concepts and they mostly choose the primitive form of words. Therefore, with the purpose of making the most of our dictionary of concepts a strict order of steps has been designed so that we can assure almost every concept included in our source dictionary is found when querying the semantic network and thus, emotional information associated to it is added to our Knowledge Base. The algorithm makes use of a third party tool to perform the morphological analysis of words called. *Open Xerox's* analysis [50] extract both the lexeme and the prefixes or suffixes of any queries word, allowing to search semantic networks by lexemes or primitive words which are more likely to be found and adjusting extracted emotions by using the detected prefixes and suffixes. Algorithm 3.1 shows a pseudo-code explanation of the method used for querying concepts.

Algorithm 3.1 Querying concepts from dictionary into semantic network

Require: Concept *concept*.
Ensure: Emotional values of *concept*

```
Get morphological analysis from Open Xerox web service
if Web service returns null then
  return Concept with empty sentiments values
else
  Declare prefixfactor variable to keep track of negation in prefixes and initialize to 1.
  for all Concept prefixes do
    if Prefix is found in negative prefixes table then
      prefixfactor ← −1 ∗ prefixfactor
    end if
  end for
  Declare suffixfactor variable to keep track of negation in suffixes and initialize to 1.
  for all Concept suffixes do
    if Suffix is found in negative suffixes table then
      suffixfactor ← −1 ∗ suffixfactor
    end if
  end for
  negfactor ← prefactor ∗ suffixfactor{Compute negation factor}
  Query the semantic network with the primitive word detected by the morphological analysis

  return Returned values of sentiments from semantic network multiplied by the negation
  factor negfactor
end if
```

As explained before, the idea of using a semantic network to build a Knowledge Base for sentiment analysis is backed on Tsai's work which suggested that semantically related concepts tend to be more strongly related than those which are not. Extrapolating this theory, we can assert that concepts will be closer to nodes representing sentiments they evoke that nodes associated to emotions they are not related to. Therefore the semantic network selected for this project should meet some requirements like the possibility of measuring distance between nodes. After studying the possibilities and limitations of the semantic networks listed in Section 2.6 some conclusions arose:

1. *ProBase* is not usable due to the accessibility issues. As a restricted networks only researchers from Microsoft are able to use it.
2. *Lexipedia* is too limited as it does not offer an API for querying distances between nodes but only a visual online tool for testing purposes.
3. Therefore, the only valid solution is *ConceptNet 5* which offers a good online API that provides tones of information including relationships between nodes.

Consequently, after comparing all alternatives the semantic network that better fits the requirements of this project is *ConceptNet 5*.

At the time of requesting emotional information of concepts to the common-sense networks developed by MIT, one condition has been considered. Concepts from *SentiWordNet* having neutral polarity are discarded so that only subjective concepts are queried. For the purpose of querying the subjective concepts the onine provided API was used. The API allows three different kinds of requests for accessing data which are described here:

- *Lookup* is the basic search. It retrieves all information about a node, which means all edges including it. It is the simplest one a hard to parse because the returned amount of data is considerably big.
- *Search* is other sort of request that looks for specific information about a concept. Results can be filtered for by every kind of relationship stored in the network. For instance you can search for the different uses of an instrument or the parts of a body.
- *Association* is the last way for data retrieval. It provides the capability of finding concepts similar to others or measuring similarity between particular nodes e.g you can look for terms related with trip to the mountains or find out how similar are zebras and horses.

Considering again the requirements of our project, the last type of request is the one we need as it allows measuring how strong are concepts interrelated. However, when defining a *concept* in the construction of our Knowledge base at the beginning of Section 3.3, *concept* was described as a pair of *word, POS* instead of single terms used for querying *ConceptNet* network. The reason to ignore the part of speech concepts are associated with in our dictionary at the time of accessing data from the semantic network is not other but the fact that nodes of concepts in *ConceptNet* are not associated to any specific POS. Therefore, although emotional intensities of concepts having the same term but distinct part of speech will get the same sentiment values, it turns out to be useful for learning purposes. Storing them separately in the Knowledge Base as *word, POS* pairs allows to make distinction at analysis time and, as a consequence, they may evolve differently so that the multiple pairs of a term improve towards their real sentiment values.

Once the semantic network which is going to be used is defined and the ways for accessing the required data are known, the only missing element to start computing the emotions each concepts evoke and their intensities is describing the concept which are going to be used in relationship queries and the formula to combine them.

The concepts used as representative nodes of all the emotions that can be represented in our emotion classification model, are different for each of the four categories and have an associated weight value. The nodes and their weight values used for computing sentiment intensities evoked by concepts are detailed in Table 3.1.

Table 3.1: Sentiment nodes and weights

Weight	Sensitivity	Aptitude	Attention	Pleasantness
3	rage	admiration	vigilance	ecstasy
2	anger	trust	anticipation	joy
1	annoyance	acceptance	interest	serenity
-1	apprehension	boredom	distraction	pensiveness
-2	fear	disgust	surprise	sadness
-3	terror	loathing	amazement	grief

Analysing the mentioned table it can be observed that although the original *Hourglass* model was modified for including the *neutral* level of intensity so that all different levels in each category have the same range of values -and thus, same probability-, the nodes selected to query the semantic network correspond to the former *Hourglass* model labels. The reason to discard *neutral* level is the fact that is less common to find occurrences of neutrality concept expressed literally in text, and as a consequence the meaningful information that can be extracted from measuring the relationship between the concepts from our dictionary to the *neutral* concept may not be significant. However, the fact of ignoring neutral level when querying *ConceptNet* for computing sentiment values of concepts does not mean that a concept cannot end up lying into the neutral level of categories, because by applying the formulas that will be presented later, the level of intensity that a concept evokes for each of the four categories is dependant on the distance to the six nodes considered for each category, allowing the possibility of a situation where distances to positive nodes of a category are cancelled or balanced by their corresponding distances to negative ones. Therefore the computed sentiment value will result close to zero and will probably lie within the bounds of *neutral* intensity.

The algorithm applied for computing sentiment values of concepts is based both on distances of concepts to the selected representative nodes of all four categories and the weights assigned to each of them. The weights which are associated to each of the terms representing emotional categories are not trivial. They correspond to the maximum values of the different intensity levels of the original *Hourglass* model which is depicted in Figure 2.4. The reason for choosing this particular values instead of the median value of each level of intensity is the fact that the proposed algorithm tends to output low intensity values being hardly possible for a concept to be annotated with a *Pleasantness* intensity within the bounds of *ecstasy* level. As a consequence, applying the maximum values of each of the levels the algorithm inclination towards lower intensities is somehow balanced. Algorithm 3.2 shows the applied steps used to compute the sentiment value of one of the category, provided that the only differences with respect to other affection categories is the set of nodes used for distance measurement. Figure 3.2 shows an overview of the sequence for building the Knowledge Base following the steps described above.

Algorithm 3.2 Computing sentiment value of one affection category of a concept

Require: Term *concept*, Category *category*.
Ensure: Sentiment value for the specified category of the input term
$finalValue \leftarrow 0$
$targetNodes$ $\leftarrow$ Nodes of the passed *category* {E.g: loathing, disgust, boredom, acceptance, trust, admiration for *Pleasantness*}
for all $targetNodes$ **do**
  $nodeWeight \leftarrow AssociatedTargetNodeWeight$ {E.g: +3 for *Admiration*}
  $finalValue \leftarrow finalValue + DistanceToTargetNode * nodeWeight$
end for
return $finalValue$

As mentioned before, the main problems of this algorithm is that although the use of weights minimize the tendency of results to be placed in the lower intensity levels, the complexity and completeness of the *ConceptNet* common-sense tends to have relationship with as many nodes as possible, causing almost every relationship measurement requested during the execution of Algorithm 3.2 will return a value greater than zero. Moreover, many concepts usually appears related to opposite emotions depending on the context, so it is *ConceptNet* responsibility to detect contradictions and people who decide to make use of their API have to rely on their accuracy. Consequently, most of the six distances would contribute to increase or decrease the emotional category value provoking a balance which may cause many difficulties to get values lying within the most intensive emotional levels.

In order to cope with the previously mentioned disadvantages of the described algorithm and with the aim of improving the results of our Knowledge Base building process, a different algorithm has been designed. The approach followed by this algorithm is to maximize the emotional intensities of concepts. Therefore, instead of using the returned distances of all six nodes of each category, only the most significant are considered. Algorithm 3.3 depicts the step-by-step of this computation alternative.

Observing the pseudo-code, we can forecast that most concepts will have stronger emotional states. However, in order to get real evidences, a simulation of how a concept would be annotated by applying each of the algorithms presented before can be found in Table 3.2.

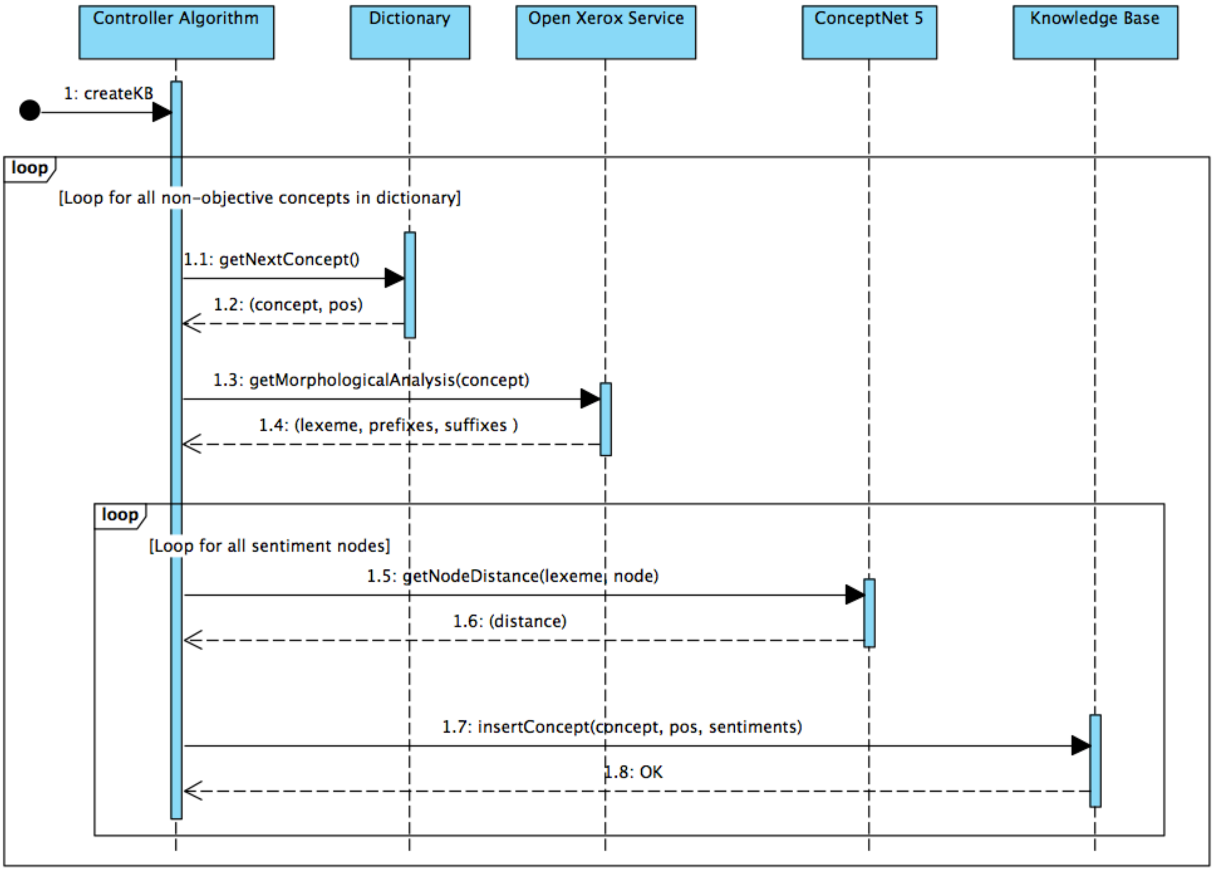


Figure 3.2: First approach overview

Table 3.2: Comparison of running the two different algorithms over simulated distances

Category / Node weights	3	2	1	-1	-2	-3	Algorithm 3.2	Algorithm 3.3
Sensitivity	0.7	0.4	0.5	0.2	0.6	0.3	0.18333333	0.3
Aptitude	0.3	0.1	0.3	0.1	0.4	0.8	-0.3166667	-0.64
Attention	0.4	0.5	0.1	0.9	0.4	0.5	-0.15	-0.6
Pleasantness	0.9	0.6	0.2	0.1	0.4	0.1	0.48333333	0.78

Algorithm 3.3 Computing sentiment value of one affection category of a concept

Require: Term *concept*, Category *category*.
Ensure: Sentiment value for the specified category of the input term

$finalValue \leftarrow 0$
$maxDistance1 \leftarrow 0$
$maxDistance2 \leftarrow 0$
$weight1 \leftarrow 0$
$weight2 \leftarrow 0$
$targetNodes \leftarrow$ Nodes of the passed *category* {E.g: loathing, disgust, boredom, acceptance, trust, admiration for *Pleasantness*}
$distances \leftarrow$ Distances to $targetNodes$ from *concept*{Array of distances preserving target nodes order}
$distances \leftarrow$ Ordered distances from max to min by distance and weight
$nodeWeights \leftarrow$ Ordered weights to match order in $distances$ array, so that distance at index i corresponds to its associated weight at i.
$maxDistance1 \leftarrow distances[0]$
$maxDistance2 \leftarrow distances[1]$
$weight1 \leftarrow nodeWeights[0]$
$weight2 \leftarrow nodeWeights[1]$
$finalValue \leftarrow (maxDistance1 * weight1 + maxDistance2 * weight1)/|weight1 + weight2|$
return $finalValue$

The results from the last two columns of the table show a determinant difference between both algorithms and back the previous predictions. Second algorithm provides more intense sentiment values for concepts, meaning that querying complex semantic networks such as *ConceptNet* may lead to get lot of noisy information -due to their high interrelation-that have to be properly filtered so that the process of building a Knowledge Base becomes more accurate. However, although being able to output a wider range of intensities should lead to a more precise representation of real emotions evoked by concepts, it does not guarantee better global results as four out of the six queried distances in each category are discarded.

Nevertheless, considering the Knowledge Base used by this project uses annealing learning to become more accurate with practice, applying any of the algorithms described before will end up converging towards the actual real emotions of each of the concepts (*word, POS*) sooner or later. The only significant differences would be observed during the training period as Knowledge Base is still refining its knowledge.

To this extent, the process of building our Knowledge Base started. Although everything worked as expected, a problem arose: the time spent on calculating sentiment values for each of the concepts of the *SentiWordNet* dictionary was around fifty seconds. The main reason for this situation was that for each single concepts 25 requests are needed. First request is done to get the morphological analysis of words through *Open Xerox web-service* and the other 24 are queries to the semantic network, as were are considering four affection categories with six target nodes each.

As the amount of total subjective concepts found on *SentiWordNet* dictionary was 41593, the estimated total execution time for building the Knowledge Base of our project was about 578 hours, which means 24 days, provided that no errors were thrown and *ConceptNet* servers never get down.

With these numbers the project was unfeasible to meet deadline the subsequent modules depends on the Knowledge Base. Therefore the algorithm needed an optimization: parallelism. Bearing in mind that for each of the concepts there are four independent categories with six concepts each representing intensity levels, the most optimized solution for minimizing the total time execution spent on each concept was to create 24 threads, being each thread used for requesting distance from the concept to a single node. By applying the corresponding code modifications, the new execution time for each concept was around 8 seconds, which means saving about 84% of the original execution time. As a consequence, the estimation for the total time spend on building the KB was updated to 4 days, which allowed the project to be finished before deadline.

However, the result of improving the algorithm by implementing parallelism led to the main problem of the project development. After a whole weekend gathering information from *ConceptNet*, MIT researchers understood that all requests sent from the computer which was constructing the Knowledge Base were part of a DoS attack, resulting in some changes on their API code which limited the number of connections per minute and block IPs for a period of time if exceeded.

As a result, the only solution left to continue building the KB was to follow the instructions from *ConceptNet* Wiki [51] to replicate the network.

With the purpose of replicating *ConceptNet* functionality the *Running your own copy* manual of the Wiki was followed. In order to run the heavy processes required by some functions of the network API, a VPS with 4 cores and 8GB RAM was hired. After a hard process for installing and configuring the *Solr* server and once the API server was executing both *Lookup* and *Search* request types, it turned out *Association* functionality was not supported because the Solr downloadable data did not include the association vectors of concepts.

Therefore, due to the lack of documentation, the only available solution was tried. Jumping to *Build Process* manual of the Wiki was a shock. The hired VPS was not enough for the computational requirements of building the semantic network hypergraph: at least 10GB of RAM were required. Even increasing the *swap* space was not possible due to the virtual environment where the VPS was installed over. Consequently, a new VPS with higher characteristics -8 Cores 16GB RAM and 200GB- was hired. Right before completing the step-by-step manual for building *ConceptNet* in our new machine an error was thrown in the last step: building the association vectors. After lot of time of searching over the internet, a forum response from one of the researchers of *ConceptNet* team suggests having at least a 40GB RAM server. As a result and due to the lack of funding, *ConceptNet* approach was discarded.

The total number of concepts which took associated sentiment values through requesting the selected semantic network was 5872, all having *adjective* as their related part of speech. The number of subjective concepts which were targeted at the beginning of this approach was 41593. Therefore, only a 14% of the total amount of concepts were included in the Knowledge Base.

3.3.2 Second Approach

In order to cope with first approach failure, other alternatives are analysed. The other semantic networks described in Section 2.6 are not an option because *ProBase* is not publicly available and *Lexipedia* does not provide required functionality for measuring distance between nodes. Consequently semantic network approach is discarded.

As a result, the only option remaining is getting emotional information from other sources. So, dictionaries containing sentiment information of terms may be used for to build our Knowledge Base. From the list of the most popular dictionaries presented in Section 2.5, we have to discard those whose terms do not contain annotations about the emotional categories we considering in our emotional model.

Applying this sifting it is found that only *NRC emotion lexicon* and *SenticNet* meet this requirement. However, although both dictionaries provides useful emotional information, none of them relates concepts with part of speech tags. As a consequence, Knowledge Base structure needs to be modified to support concepts with no associated POS --*1* in database- maintaining backwards compatibility so that it is able to benefit from information gathered from *ConceptNet* too. Moreover, as these lexicons also offers *polarity* values, the Knowledge Base changes also includes adding the needed column for storing polarity so that even though this project is not concerned about polarity of texts it is prepared for future improvements of the algorithm. Figure 3.3 shows the new approach to the Knowledge Base building process.

NRC emotion lexicon

NRC emotion lexicon can be downloaded for free from Saif Mohammad site [52]. It is formatted in plain text and contains boolean values for the eight Plutchik's basic emotions plus polarity information. When any of the eight emotions is marked as 1, it means this concept evokes this emotion. In order to convert NRC's classification to our four-category-

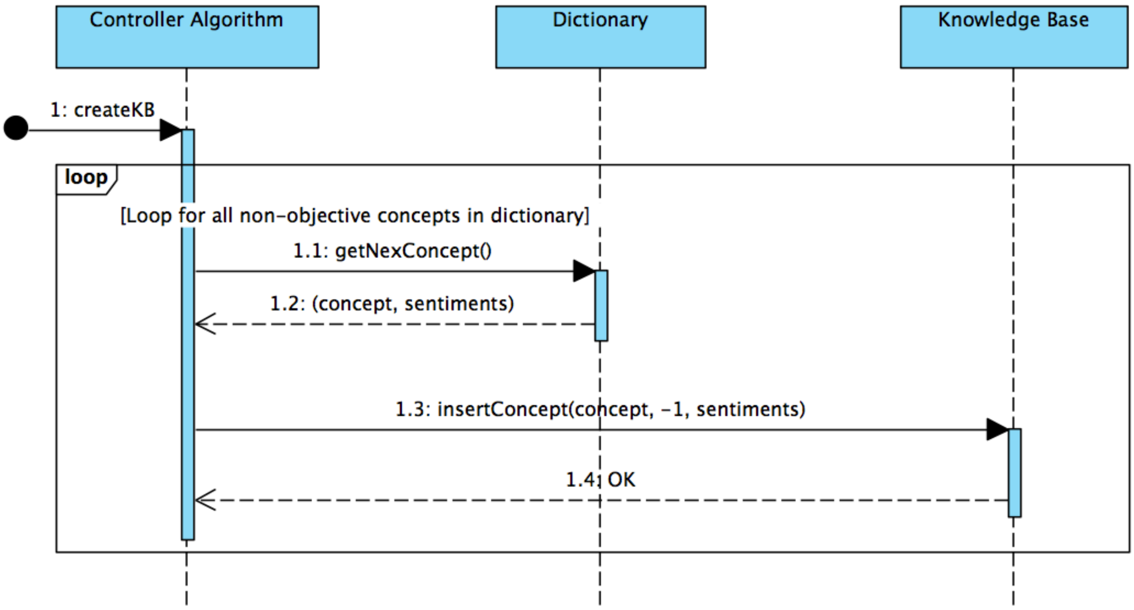


Figure 3.3: Second approach overview

model we have to combine both positive and negative levels of each category and assign a value for categories which are not neutral. The possible values for the four categories of our model after combining opposite sentiments from NRC lexicons are either 1, 0 or -1. The selected way of combining these opposites emotions lies in subtracting the value associated with the positive intensity minus the one of the negative intensity. Table 3.3 shows a real example of the annotations of *abandon* concep, their global average value and the mapped value in Knowledge Base. Polarity is included as a category with the purpose of showing the differences in calculation. There are three possible values for polarity are positive, negative and neutral, whose correspondent values are 1, -1 and 0 respectively. Therefore, the representation of polarity in our KB is the result of subtracting the positive minus the negative value.

The assigned values for the non-neutral categories are not trivial. They match the medium value of the either positive or negative intensities of the model used in this project, providing that Knowledge Base stores sentiment values in a range of [+1, -1] instead of [+3, -3] used for the model. Thus, to represent those values in the model scales they should be enlarged proportionally to model's range.

As a conclusion, parsing NRC lexicon makes the Knowledge Base to grow in 6425 concepts, all of them without any associated POS. This means that from the original total amount (14182) of terms included in NRC lexicon 7757 are completely objective -all sentiment values equal 0 and neutral polarity- and thus, discarded. Therefore, only 45.3% of the concepts from NRC dictionary are useful for our project.

Table 3.3: NRC - Knowledge Base mapping of concept *abandon*

Category	Labels	Values	Combined value	Value in KB
Sensitivity	Anger	0	-1	-0.5714286
	Fear	1		
Aptitude	Trust	1	1	0.57142857
	Disgust	0		
Attention	Anticipation	0	0	0
	Surprise	0		
Pleasantness	Joy	0	-1	-0.5714286
	Sadness	1		
Polarity	Positive	0	-1	-1
	Negative	1		

SenticNet

SenticNet lexicon is the other valid source of knowledge in this second approach. The lexicon can be freely downloaded from the project web page. Concepts are *n-grams* stored in XML format . Information provided includes semantic relations, sentiment values and polarity of concepts. Parsing this lexicon is very similar to NRC dictionary regardless of the file format.

The total amount of concepts contained in *SenticNet* lexicon is 14244. All of them containing significant information about sentiments or polarity.

At the end of this second approach, the process of building the project's Knowledge Base would be considered to be finished. However, due to the lack of time, *SenticNet* lexicon has not been imported so the total amount of concepts stored in the KB is 12297.

3.4 Sentence Analysis Model

Within this section the tools used for analysing sentences are presented. Furthermore, the different approaches to the analysis are described. Sentence scoring is not detailed along this section due to its complexity but in the following one.

3.4.1 Parsing

The process of parsing texts makes use of different Natural Language Processing tools. Parsing a sentence outputs a semantically analysis of sentences, containing part-of-speech tags organized in a tree of predicates, being the root the whole sentence and structuring the rest of nodes from bigger to smaller predicates towards tree leaves. With the purpose of executing parsing over user's texts, a research on available NLP libraries alternatives is needed in order to choose the best available option. Considering the three most popular general-purpose libraries: LingPipe, NLTK and OpenNLP, it can be observed that all of them includes both part-of-speech tagging and parsing functionalities, which are the ones required for this project. However, the fact that NLTK is written in Python constitutes an important gap as it turns library integration process harder. Therefore, NLTK is discarded.

OpenNLP

On the other hand, both LingPipe and OpenNLP meet desired requirements as they are general purpose libraries having good documentation and being in a mature stage of development. However, the fact that OpenNLP is maintained and developed by Apache tips the balance towards this library.

OpenNLP parser makes use of the Penn Treebank format, developed and maintained by the Penn Treebank Project of the University of Pennsylvania. The only limitation of the OpenNLP parser is that only individual sentences can be parsed. Consequently, before parsing sentences the text should be split into sentences by using OpenNLP probabilistic *Sentence Detector*, which offers a precision of 94% and a 90% recall. Once sentences are identified, the parsing process can start.

Penn Treebank Notation

As the OpenNLP parser uses Penn Treebank notation, details of this format are depicted here. The main convention of Penn notation is the fact that *parts of speech are defined on the basis of their syntactic distribution rather than their semantic function* [53]. As a consequence nouns used in the function of modifiers are tagged as nouns instead of adjectives. Penn notation considers 36 sort of part of speech. However, for sentiment analysis purpose and with the aim of simplifying the analysis model, only parts-of-speech which may carry some emotional information are used in the analysis algorithm. Table 3.4 shows the actual mapping between parts of speech considered by the Penn Treebank Notation and tags used for knowledge representation in this project.

Therefore, the problem is dramatically simplified as the analysis model of this project works over four different tags instead of the 36-tag-convention proposed by Penn Treebank Project.

Table 3.4: Penn Treebank tags - Knowledge Base tags mapping

POS	Penn Treebank tag	Used tag	POS in KB
Adjective	JJ	JJ	0
Adjective, comparative	JJR		
Adjective, superlative	JJS		
Noun, singular or mass	NN	NN	1
Noun, plural	NNS		
Proper noun, singular	NNP		
Proper noun, plural	NNPS		
Adverb	RB	RB	2
Adverb, comparative	RBR		
Adverb, superlative	RBS		
Verb, base form	VB	VB	3
Verb, past tense	VBD		
Verb, gerund or present participle	VBG		
Verb, past participle	VBN		
Verb, non-3rd person singular present	VBP		
Verb, 3rd person singular present	VBZ		

3.4.2 Detecting negation

One of the challenges of opinion mining is detecting negation. It is one of the hardest tasks due to language ambiguity, for instance detecting irony in texts is extremely difficult as it is usually related with the context or situation within the human is immersed. In this analysis model the process of detecting negation in sentences is not considered a priority, but it is required at least to build a simple implementation so that more evident negations are detected.

In order to do that, a proximity based approach is chosen. The assumption of this approach is the fact that a *negator* affect words or predicates close to it. As a consequence, words from text which are not identified as concepts, are queried into a database table that contains more common terms showing negation -*modifiers* with -1 factor- in English language. In case the word is found the sentiment values of the predicate containing the word are multiplied by -1. In case several negators are found in the same predicate, sign of emotional intensities will change as many times as number of negation words are found.

3.4.3 Modifiers

As defined in Section 3.3, *modifiers* are words that either increase, decrease or reverse meaning of words they describe or go with. As negation is considered a different problem from the point of view of opinion mining, it has been described previously. The approach towards detecting which concepts are affected by modifiers is the same followed for detecting negation. This proximity based approach allows the algorithm to know that sentiment values of the container predicate have to be adjusted using the factor of the detected modifier. In order to build the modifier - factor table in database, factor values have been approximated by word definition from The Free Dictionary [54]. The range of possible values taken by modifiers' factor is given by: $factor \in [-1] \cup [0,1) \cup (1,2]$. Being *-1* used for storing negators and rest of values for common modifiers. Table 3.5 presents some modifiers included in the project Knowledge Base.

Table 3.5: Some modifiers included in our Knowledge

Modifier	Factor
A bit	0.2
Lots of	2
Little	0.5
Less	0.3
Much	1.7

For instance, the analysis of the sentence *I love you much* will detect *love* as a concept and *much* as a modifiers, in this case affection the whole predicate *love you much*. Consequently, *love*'s sentiment values should be multiplied by factor *1.7* which is associated to *much*.

3.5 Text Scoring Scheme

After parsing process is finished and all concepts, modifiers and negators have been tagged properly, it is possible to begin with the computation of the final sentiment values of the text. Scoring process follows a bottom-up approach, where emotional intensities of tree leaves are computed firstly, climbing the text's tree step by step until root node sentiments are computed, which are indeed the emotions evoked by the whole text.

The scheme for computing the sentiments for each of the predicates, and therefore of the whole text, is determined by a fixed algorithm which relies on the Knowledge Base accuracy, a proximity based approach for modifiers and a topic detection module to detect the most relevant topics of a text so that sentiment of sentences related with them become more important in the final sentiments results.

3.5.1 Topic detection

Topic detection is, as negation detection process, a really complex research field. The approach to topic detection within this project is the simplest one. *Entities* are identified by searching possible targets of sentiments. The algorithm considers all words tagged as either nouns or pronouns. However, the lack of a set of rules for representing the direction of the verbs prevents the algorithm from being able to differentiate between holder and target of sentiments.

Nevertheless, the method used to recognize most significant topic in the text follows a similar approach to a bag-of-words algorithm.During the parsing process the possible targets are counted so that a vector of entities and their number of occurrences is created. In order not to alter topics relevance, both plural and singular form of same entities share a common counter so that they are counted together.

However, the hardest task within this approach is analysing pronouns. Language ambiguity turns pronoun-to-noun mapping into a extremely difficult process. To this extent, as the project is not focus on language disambiguation functionality in charge of detecting pronouns -although implemented- is deactivated so that only nouns are taken into account at the time of identifying most relevant topics. Here below there is a sample text with nouns highlighted and the resulting vector of occurrences.

> *Brazilian **people**, the one who live in **favelas** or cannot access higher **education**, don't want the **Football Cup**. Although **Brazil** has been traditionally a **country** with an important **football culture**, current economic **situation** has awaken many **people** who are becoming aware of the real **problems** and do not agree with the huge public **investments** on **facilities** for **World Football Cup event***

The set of entities found of the text ordered by appearance:

[people, favela, education, football, cup, Brazil, country, culture, situation, problem, investment, facility, world, event]

And the associated occurrence vector of the set of entities of the text would be:

[2, 1, 1, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1]

So as it can be observed in the example, the entity *football* is the one which appears most in the text, assuming that sentences where it is located should carry more weight in the total sentiment computation as it is likely to be the main topic of the whole text.To this extent, the number of occurrences of each entity is used by the scoring model as the weight assigned to the predicates where entities are placed.

3.5.2 Sentences weighting

The way sentences are weighted is based on entities occurrences. The process starts with smaller predicates going up to the bigger ones, from leaves to root. This bottom-up approach is needed as sentiments of a predicate are dependant of the smaller predicates contained on it. Therefore, first step lies in resolving leaf predicates so algorithm can keep on computing bigger predicates and so on until the root sentiments are calculated, standing the root node for the whole user's text.

The main idea of sentiment computation is that all elements having emotional information within a predicate have to be added to get the global predicate emotions. In case there is a modifier inside the predicate, the computed sentiments have to be multiplied by the modifying factor. Moreover, for each negator which is present in the predicate, the sign of the values of the four affection categories of the predicate has to be shifted once.

However, regarding the scoring model, a distinction is needed. On the one hand, when computing emotions of predicates not holding any entity, sentiment values need to be computed as if each of the children of the predicate had $weight = 1$. Therefore, all elements are treated equally and the average value of emotions will be the output. Despite of using $weight = 1$ for computations, the predicate is annotated with $weight = 0$ so that it keeps representing the fact that no entity is contained neither in this predicate nor in any of the descendants. On the other hand, predicates containing weighted elements -either entities or other predicates- has to compute separately weighted and non-weighted components. Weighted elements combines their elements following Equation 3.1, and predicate components with no weight are assigned average weight of weighted elements, assuming that if no entity is found on a predicate it affects equally to all siblings predicates of the tree.

Let be w_i the weight of a predicate and n the total number of sibling predicates which are being combined, the sentiment value of a category for weighted predicates can be defined as:

$$S_w = \frac{\sum_{i=0}^{n} w_i * s_i}{\sum_{i=0}^{n} w_i}, \qquad \begin{array}{r} \forall w_i > 0 \\ \forall s_i \neq 0 \\ s_i \in [-1, +1] \\ i = [0, n] \end{array} \tag{3.1}$$

The key point of this weighting method is that categories values are computed separately. Predicates with no sentiment value of the specific category which is under analysis is discarded. However, as categories sentiments are computed independently one of each other, predicates are also discarded independently for each category. Thus, a predicate may be discarded for *Sensitivity* and may hold sentiment values for the other categories. As a consequence, total sum of weights of a predicate is specific for each category, although the final predicate weight is the average of children weights regardless their sentiment values (discarding zero-weight elements).

3.6 Learning: Improving Knowledge Base

As stated in Section 3.1, this project make use of heuristic learning for improving the Knowledge Base used for sentiment analysis. The reason for choosing an annealing learning approach instead of a machine learning one is the fact that as a knowledge-based system, there is no need for probabilistic classifications offered by machine learning algorithms, but on the contrary users annotations can be used to converge to the optimal solution (value) of each concept in the knowledge base. Thus, applying an adaptive learning rate which allows faster learning during training process and which gradually slow down the learning rate as long as the optimal solution is getting closer. This method assures that system evolves correctly unless malicious annotations are provided, and therefore its performance converges with time towards human interpretation of sentiments evoked by texts.

In the case of our knowledge base, the features used for learning are *concepts*. Therefore, each word, POS pair will learn independently so that same concepts assigned to different parts of speech can evolve differently as they may evoke distinct emotions. The training period is of this project is been set in 1000 samples for each feature, meaning that most frequently used concepts will learn earlier and rare concepts' learning process may last longer.

As a consequence of the predicate weighting policy stated in Section 3.5.2, not all recognized concepts from a text account the same for each emotional category value computation. Therefore, learning process needs to guarantee that concepts involved in the computation of a category learn proportionally to their weight when computing sentiment analysis. Consequently, concept-weight relations are stored in a database for each of the not null sentiments of particular concepts so that they can used in learning process. The calculation of the actual weight a concept holds within the computation of the global text sentiment is performed considering predicates relevance from root to leaf nodes based on their weights, multiplying the proportional influence of each predicate in the computation of sentiment values. It is computed separately for the four categories.

In order to compute the adaptive learning for each record stored in the concept-weight relation database, equation 3.2 is applied considering the difference between the sentiment analysis output of the project algorithm and the correct sentiment values sent by the user. By the time the learning process has finished, all records of the concept-weight table associated to the specific text are deleted.

Let be U the set of sentiments of a text corrected by the user, M the sentiments calculated by the algorithm, W_{C_s} the weight of concept C for sentiment s and A_c the number of accumulated adjustments of concept C. Therefore the new value of each sentiment s of concept C is defined as:

$$C_s = C_s + \frac{(U_s - M_s) * W_{C_s}}{1 + (A_C/1000)} \tag{3.2}$$

The final model components and data flow between them is presented in Figure 3.4 in order to give an overview of the proposed Sentiment Analysis scheme.

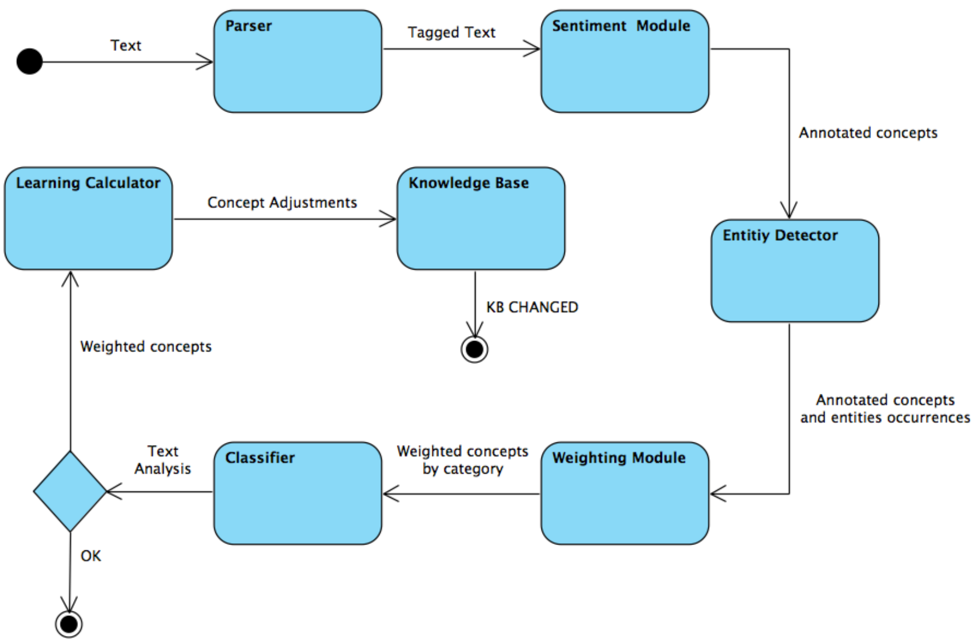


Figure 3.4: Data flow diagram of the proposed model

Chapter 4

Prototype for testing

As stated in the Introduction chapter of this report, one of the main goals of this project is to take advantage of social networks to teach the algorithm and measure its performance. For this purpose a MVC prototype has been developed. An overview of the prototype's structure is shown in Figure 4.1. Along this chapter, the main components of the application will be described in details focusing on their inner structure and functionalities.

4.1 Model

The model selected for the prototype is composed of two main elements. On the one hand, a MySQL database is used for storing user profiles, settings, authentication tokens for security and post related informations. On the other hand, filesystem is used for storing images uploaded by users.

The principal reason for making distinction between pictures and information is guaranteeing the portability of the database. Although storing images in database tables makes a more consistent model due to the fact that synchronization is assured, the number of expected picture uploads in a photo-sharing social network is huge. Therefore, given that *INSERT* and *UPDATE* operations are locking in MySQL and that insertion and retrieval of *Blob* elements are expensive as blob type was not initially supported, we can state that possible drawbacks derived from potential inconsistency between tables and filesystem are overtaken by the general application performance and thus we chose to store images separately in the filesystem and just keep references to them in database tables.

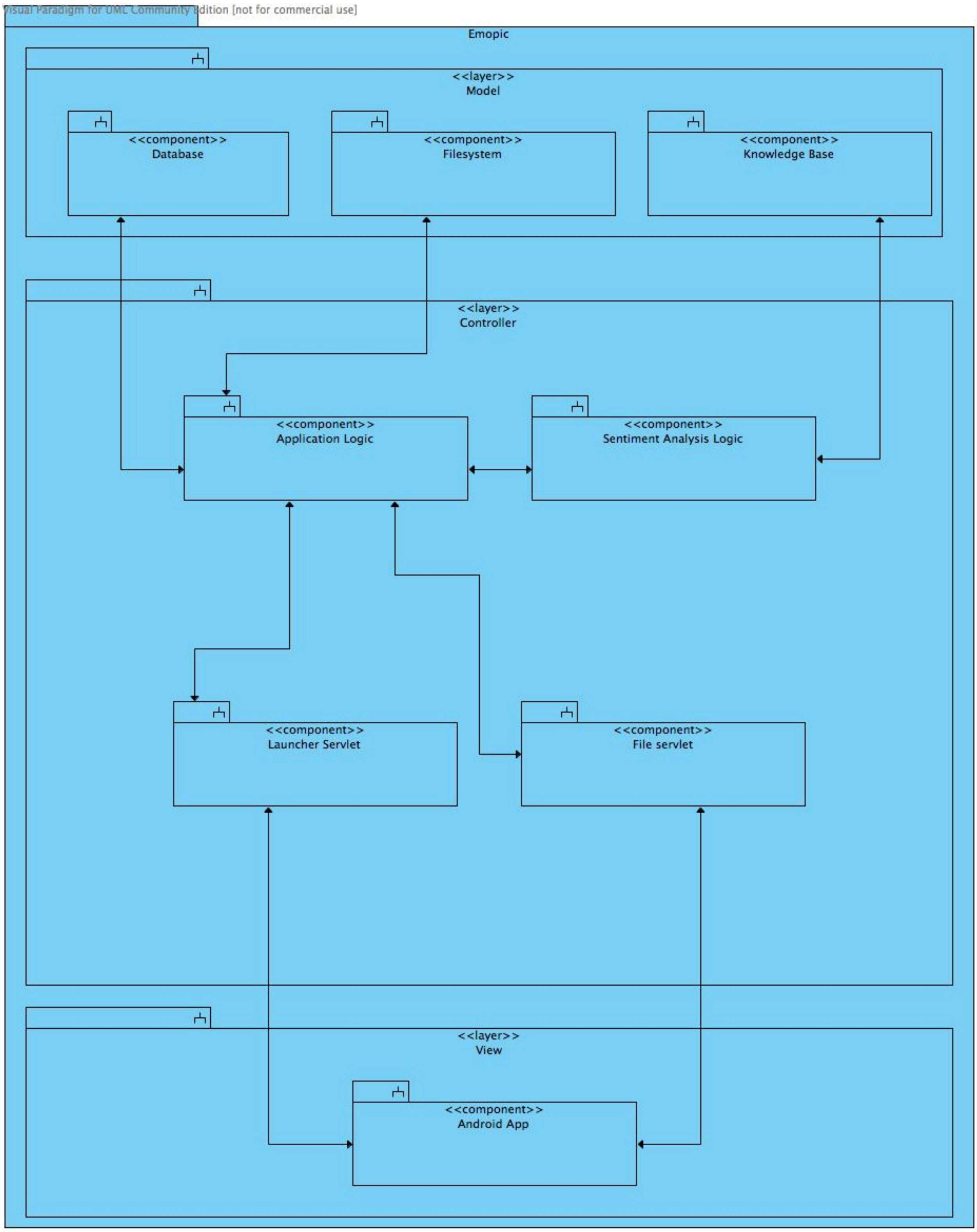


Figure 4.1: MVC prototype architecture

4.2 Server

The server module shown in prototype structure diagram corresponds to the *Controller* of the application. It is in charge of receiving user requests through the *View* and providing responses by retrieving or inserting information into the *Model*. Therefore, server is responsible of application logic and communication between the mobile app and the database.

Moreover, server integrates both the application logic and the sentiment analysis processing. As a consequence, server triggers application functions on users demand and also computes the required sentiment analysis processing to answer users requests.

Hence, server is able to communicate with the model through application functions and with the Knowledge Base through sentiment analysis logic in order to fulfil operations requested by users.

The server is coded in JavaEE 6.0. The main reasons for choosing Java as server's language are the amount of experience in Java programming of the author and the fact that Android platform is also written in Java. Consequently, choosing Java for the server does not increase project complexity and guarantees an easier approach towards the project as the required skills for code comprehension are not handicapped by the variety of languages.

The developed API for the server is based on Servlets, although it may be moved to web services in future for easing integration with Android application. There are two different servlets listening for user requests. One for images download requests -called *FIle servlet-*, and other for the rest of app operations. The reason for splitting the requests into two different servlets is the way of how communication protocol is designed.

On the one hand, the file servlet is used for matching URLs with files in the filesystem. It is a generic servlet which may serve any kind of file, although in this project is only used for accessing images. Request URL is used for retrieving the folder structure and name of the file to be downloaded and it is matched with its corresponding file path in server. Once located the image is served. For instance, querying the server with URL will download an image posted by user *peter* with name *littleboxes.jpg* if found in server.

On the other hand, the other servlet, which is called as *Launcher Servlet*, receives the rest of requests and triggers the needed operations so that desired responses are generated. In order to be able to know which functions are trying to be run by user requests, two different options are provided:

- *Multipart Request* is used for uploading images to the server during posting process. The reason to use this kind of content type for uploads is their efficiency for sending large amount of data such as files by dividing files into different parts which are sent in order. There are five compulsory parts for image uploading:

  - *scope* is used to define the group within the function that is going to be triggered is included. For image uploading scope is: *images*.
  - *method* is the name of the function to be run in the server: *images_ insert* is the one needed for uploads.
  - *image* Stream stands for the image data which is about to be sent to the server.
  - *postid* is needed to complete the reference to the image file in the corresponding database row.
  - *userid* is required for both placing the image in the right server folder structure and for checking whether the provided post id belongs to this user or not.

- *JSON Request* is the generic communication protocol used in the project, therefore it is considered the core of project's API. The JSON object which is needed for communicating with API is composed of three different elements:
  - *scope* is, like in multipart requests, the group where method is located.
  - *method* stands for the function to be called in the server.
  - *params* are the arguments required by the function specified in *method* field.

In both cases, the Launcher servlet extracts the *scope* and *method* from the request so that the function can be triggered with the passed arguments -either parts for multipart requests or *parameter* element for generic calls.

Once the requested operations have finished, the server produces a response. The format of the response is a JSON object with three different elements which allow the server to output what ever is the result from the query, either it has succeeded or not. Here are the definitions of the different elements composing server responses:

- *row* is used in two different kind of responses. On the one hand, when responses aim to retrieve some information from the server the *row* field is used as a counter of the number of retrieved elements from the query. On the other hand, for operations which are not requiring information but triggering internal server operations, this JSON field is used as a Boolean indicator of the succeed or failure of the operation.
- *msg* field is only used for occasions where the server execution fails and an error is thrown. The error message is assigned to this field so that developers can obtain the required information to modify their code or notifying a bug instead.
- *result* is the field used for embedding database queries' result. Whenever the *row* field is greater than zero this field will not be empty, assuming that the triggered function is for retrieving information.

Regarding security issues, as the application developed within this project is a prototype and therefore it is in its early stages of development, no security mechanisms have been implemented apart from user authentication. The reasons for this lack of security are two. First, being a prototype the main goal is focused on showing how API is used and giving an idea of what kind of application this project may have. Secondly, security is not in the list of project's goals described in Section 1.2, therefore no security issues have been considered in implementation due to the lack of extra time during project development.

4.3 Android App

The *view* component of the prototype is a mobile application for Android OS consisting of a social network for sharing photographies called *Emopic*. The main reason to choose this kind of mobile app is the popularity of similar products that are out in the market such as Instagram, Tumblr, 500px, etc. Thus, a good welcome can be expected providing that it offers some differential features that no other photo-sharing mobile app has.

The Android minimum Android version required to run the app is Android 4.1 Jelly Bean because as we can see in Figure 4.2, more than 70% of mobile phones based on Android platform are running this version or newer ones. Moreover, this version includes some new features which make the development process much easier and, provided it is a prototype, it is enough to target almost three-fourths of the potential users.

Version	Codename	API	Distribution
2.2	Froyo	8	0.8%
2.3.3 - 2.3.7	Gingerbread	10	14.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	12.3%
4.1.x	Jelly Bean	16	29.0%
4.2.x		17	19.1%
4.3		18	10.3%
4.4	KitKat	19	13.6%

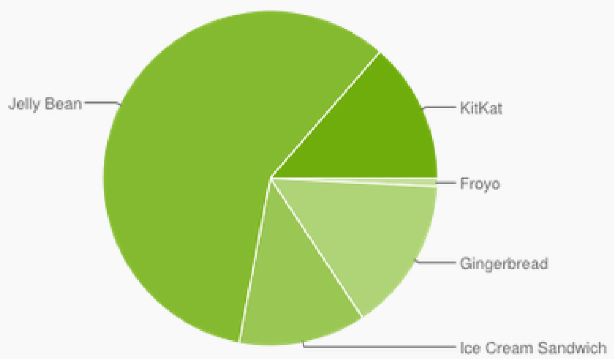


Figure 4.2: Usage statistics of different Android platform versions on June 4, 2014 [7]

Regarding app use cases, we can distinguish three main processes: accessing the app, browsing Emopic gallery and posting images. For the access issue there are two different actions: registering and logging in. The screen for both options is similar although Sign Up screen requires more fields to be typed. In case the user already has an account the logging access should be used. In the latter option, only username and password are required. Access process is required only once, as user credentials are stored in the mobile phone so there is no need for entering user and password each time the application is executed.

Apart from accessing the mobile app, there are two other use cases which are the most important functionalities of the prototype. On the one hand, browsing images can be done through two different screens. The initial screen shows all pictures shared in the system in chronological order starting from the newest in a timeline design like the FIgure 4.3 shows. The method used for loading images is a lazy loading algorithm that downloads pictures asynchronously from the server and is continuously discarding loaded images that are not being shown in the screen so that memory is freed and app does not crash. Therefore, traversing the timeline quickly would make the user notices the time spent to download and rescale the picture before it is presented on screen. However, there is no other way to do it when dealing with big amounts of images due to the limited memory available in Android devices.

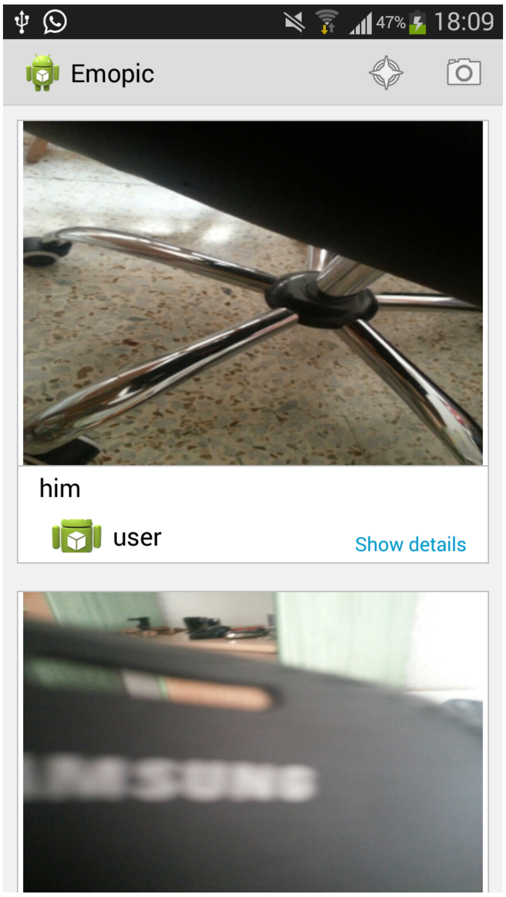


Figure 4.3: Timeline Screen

Furthermore, although all photos are accessible from this screen it may not be convenient for browsing specific images that are for instance one year old as it will took much time to load all pictures posted for a one-year period. In order to solve this problem there is another screen for browsing images called *Explore* which is presented in Figure 4.4. This screen allow users to filter Emopic gallery by the sentiments identified in the analysis of their description. Thus, users can for example filter images expressing *surprise* or *rage*. This feature is particularly interesting from a psychological and artistic point of view as it allows users to compare the different artistic representations of the same emotions in different photographers.

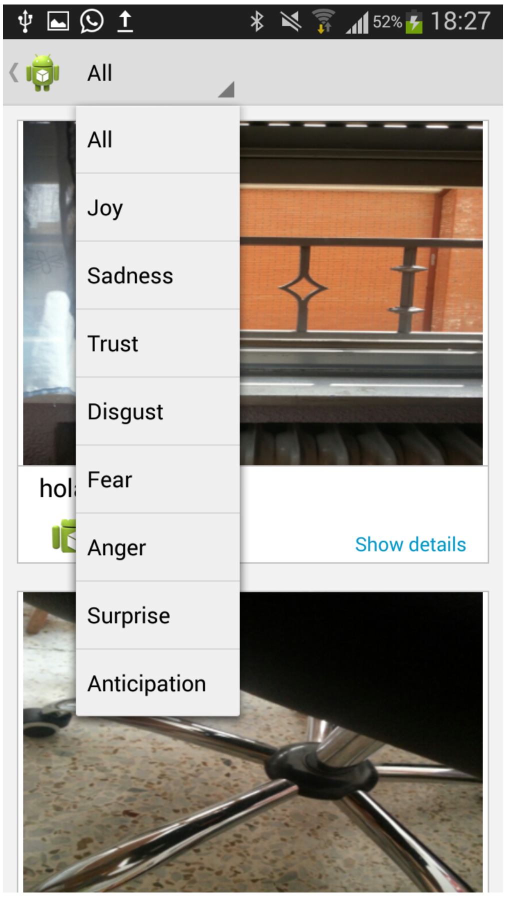


Figure 4.4: Explore Screen

On the other hand, posting process is the reason why the prototype exists. It stands for the steps taken since the user decides to capture a photo and share it on the app, to the association of sentiments to the image description. The process starts by tapping on the camera icon as depicted in Figure 4.5 and can be divided in four stages: take photo, type description, sentiment detection, sentiment correction. The snapshot stage makes use of the default camera service client provided by Android. This means the user does not need any special knowledge as it works like the traditional Camera app. When the user is done taking the picture, it is stored in a local directory until next step is completed so that post can be cancelled and no internet communication is wasted. Before storing the image locally for later uploading, the image is resized so that all pictures posted in Emopic have equal dimensions. The reasons to perform this resizing are basically three: firstly, the amount of internet bandwidth saved by reducing snapshots' size is considerably significant and therefore allows users navigate fluently through the timeline without requiring a long time to load pictures (providing the internet connection is not very poor). Secondly, taking in account that Emopic social network is intended as a tool for this project evaluation and not a commercial product, we can state that the need of reducing server costs are more important than guaranteeing high quality for big screens as for the moment, the app is only available for mobile phones and not tablets or computers where screen size and resolution are more important. Last but not least, storing small images assure the impossibility of high quality reproductions of the photos so that users copyright is somehow protected.

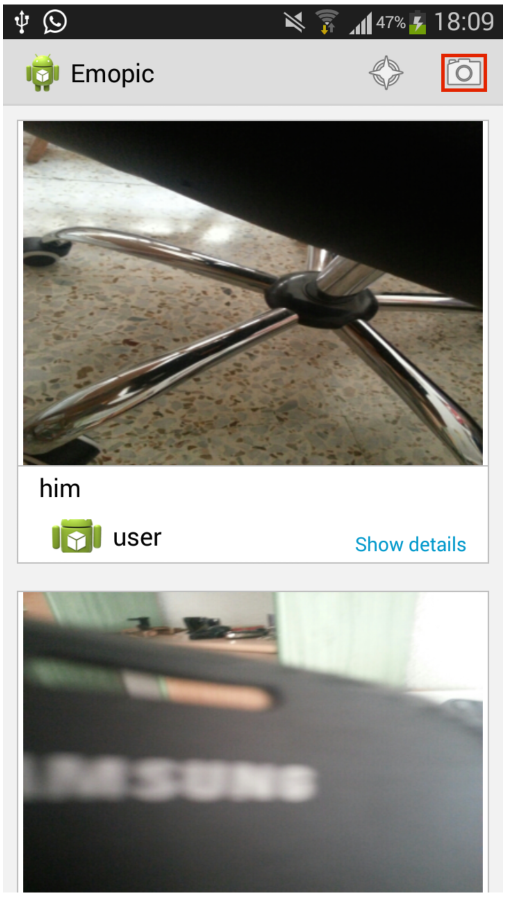


Figure 4.5: Highlighted Post Button

In the description screen shown in Figure 4.6 , the user is asked to enter a kind of long text describing the feelings or situations leading to the snapshot which is about to be posted. As soon as the user ends typing and clicks on finish button, the app starts uploading the image. Once this step is over, image has been uploaded to Emopic system and therefore posted, so user can no longer cancel the post. Indeed, as long as the app is a prototype, post deletion is not implemented yet.

Right after image is posted, Emopic server starts running the implemented SA algorithm

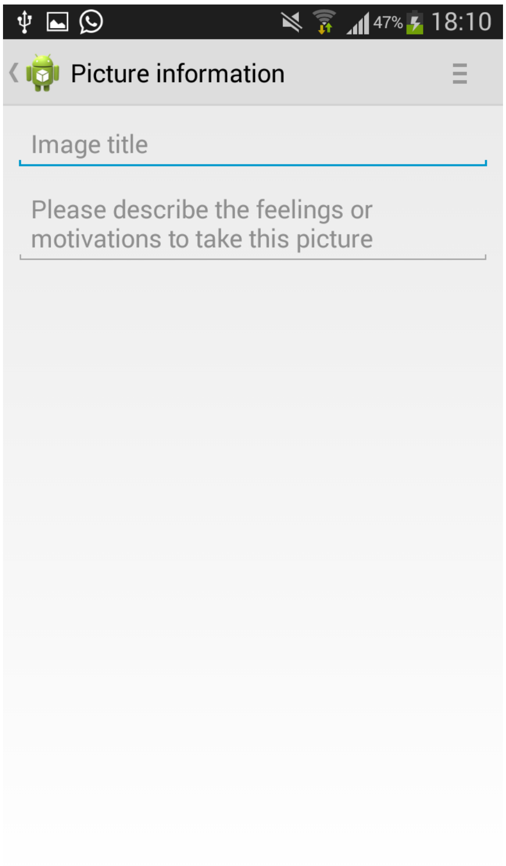


Figure 4.6: Description Screen

described in Chapter 3 over the provided description. The reason for running Sentiment Analysis in the server instead of locally is the need of many heavy resources for sentence tokenization and a huge database of concepts that will drain dramatically phone battery. Next screen of the app shows the results of the analysis. Sentiments detected in the text are shown with their respective intensities as can be seen in Figure 4.7. As it can be noticed, the sentiments representation in the app differs from the one used in the computations of the SA algorithm. This is because Plutchik's classification of emotions is more familiar for people than the *Hourglass* categorization model described in the *State of the Art* chapter of this project which only includes four affective dimensions that can both positive and negative intensities. Concretely, the list of emotions used for sentiment representation within the mobile application includes the eight basic emotions proposed by Plutchik: anger versus fear, sadness versus joy, disgust versus trust and surprise versus anticipation.

To finish posting process user have two possibilities: either accept the results of the text analysis if they match user's real emotions or correct if the analysis is not accurate enough. If user chooses to click on accept button the post is finished and user is driven to main screen. However, in case the user's choice correspond to correcting the results, the screen

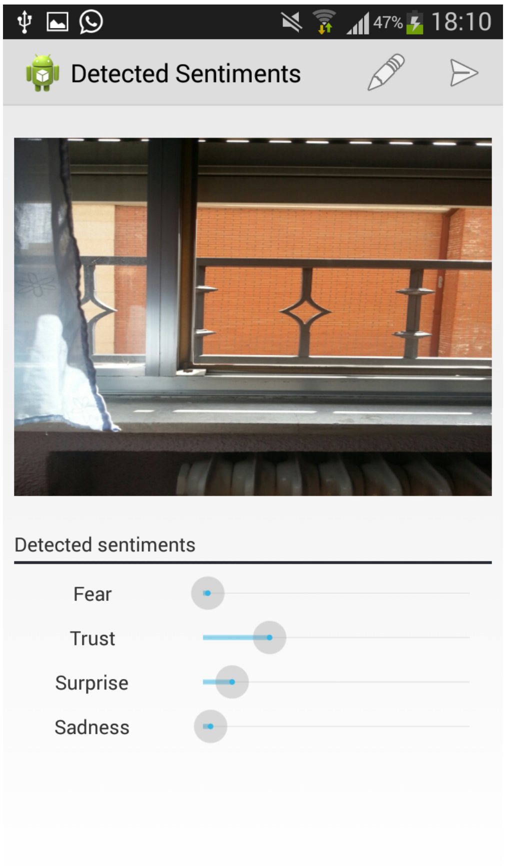


Figure 4.7: Analysis Result Screen

shown in Figure 4.8 allows to select the right intensities for the sentiments expressed in the post description. After that, user just has to tap over the accept button to send the new values to the server so that the previous assigned intensities are substituted by the corrected ones and the post is updated.

From the point of view of project evaluation, the most interesting part of users' posts are the last step described above. The choice made by the user in this stage of the posting process has a significant meaning for the learning of the developed algorithm. On the one hand, whenever the user accept the results generated by the text analysis means that the text has been correctly classified and thus, the SA has succeeded in detecting the description's emotions. On the other hand, every time a user disagree with the outcomes of the analysis and chooses to provide the right sentiments of the introduced description, the new sentiments are not only used to update post's classification but also to teach the algorithm by adjusting concepts weights as explained in the *Sentiment Analysis Algorithm* chapter of this project.

From the design point of view, the main characteristic of Emopic social network is the fact that all posted photographies are publicly available. Contrary to the most common approach of friends-list based social networks, by publicly sharing all the uploaded pictures users have access to a huge amount of content which gives users a chance for popularizing their pictures beyond their relatives and friends. Furthermore, they will also get access to new photography style and from a researcher point of view, Emopic gallery will be an opportunity for comparing how people from all around the world express their emotions through their photography.

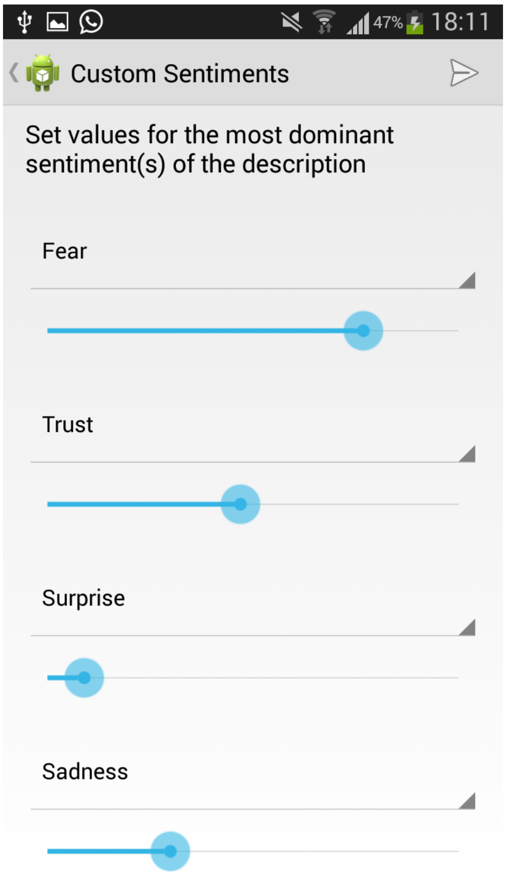


Figure 4.8: Custom Sentiments Screen

Chapter 5

Evaluation

The evaluation process of this project is accomplished from two different approaches. On the one hand, first approach is based on objective data gathered from application usage. On the other hand, second approach focuses on collecting data about user satisfaction. Along this chapter both kinds of evaluations are detailed centring attention on how data is obtained and what factors can be evaluated on each of the approaches.

5.1 Statistical performance

The first approach to project evaluation aims to measure algorithm performance mathematically. The prototype described in Chapter 4 is used during this kind of evaluation with the purpose of gathering data. The required data for this sort of analysis is collected during application execution. In order to test the global accuracy of the algorithm, the results of sentiment analysis of every text posted by users are stored in database in a specific table for generating statistics. Afterwards, the emotion values accepted by the users are saved in same post row so that if user agrees with the detected emotions it is considered a success, otherwise it is a fail. In any case, the database will contain a row for each post in the social network containing information about proposed analysis and accepted one. Therefore information from this table can be used to compute required statistics to evaluate algorithm performance.

The main factors commonly used for validating classifiers are *precision* and *recall*. In order to understand both concepts the Table 5.1 shows the terms used for describing the comparison between real condition of the text and classifier's prediction. On the one hand, *true* and *false* terms determine whether the prediction matches or not the condition of the text. On the other hand, *positive* and *negative* refer to classifier's prediction.

Table 5.1: Terminology for comparing real conditions to classifier's prediction

	Condition Positive	Condition Negative
Predicted Positive	True Positive (TP)	False Positive (FP)
Predicted Negative	False Negative (FN)	True Negative (TN)

Within the context of this project, precision is defined as the number of texts classified in a specific category that really belongs to this category -known as true positive- divided by the total number of texts classified in this category -the sum of true positives (TP) and false positives (FP)-. On the contrary, recall is described as the number of texts which have been classified properly in a category (TP) divided by the total number of texts that belongs to this category -positive (P)-. With the purpose of better understanding, the equation for calculating precision is described in Equation 5.1 and recall function is depicted in Equation 5.2.

$$precision = \frac{TP}{TP + NP} \tag{5.1}$$

$$recall = \frac{TP}{TP + FN} \tag{5.2}$$

Therefore, *precision* stands for the probability of finding a text properly classified within the texts classified in one category. Furthermore, *recall* is described as the probability of picking a random properly classified text in a search.

However, these factors are used to validate binary classifiers, so that positive and negative conditions can be assigned to each of the categories. In the proposed model, as there is a neutral range and both positive and negative categories are divided in intensity levels, it is not possible to measure performance through the use of *precision* and *recall*.

Instead of that, an analysis of the data gathered during execution is presented. First of all, Table 5.2 depicts the absolute success and fail rate for the sample texts both for categories and for the whole analysis -that is success in all four sentiment categories at the same time-. As it can be observed, the average success rate for categories is around 60% although the whole success rate is about 43%. This table only considers the exact values accepted by the users compared to the ones produces by the proposed model, meaning that no margin has been considered.

To this extent, it can be observed that in many occasions, some of the category values are not computed properly, therefore leading to an overall success significantly lower than the success rate per category. This situation is due to the way concept's sentiment values were computed in Knowledge Base building stage, as ConceptNet graph connects almost every

Table 5.2: Summary success / fail rate of algorithm.

	Success		Fails		Total	
Sentiments	Count	%	Count	%	Count	%
Sensitivity	28	50	28	50	56	100
Aptitude	40	71.43	16	28.57		
Attention	41	73.21	15	26.79		
Pleasantness	33	58.93	23	41.07		
All	24	42.86	32	57.14		

node so that there is always some computable distance between concepts and sentiment target nodes, thus being really hard to get zero values for any of the emotional categories. In order to filter this noisy results produced by the lower and insignificant intensity values, a margin can be defined for computing success rate, so that minor differences between user and model values are ignored. Table 5.3 shows the results of applying different margins when computing success rate.

Table 5.3: Comparison of success rate by category based on applied margin

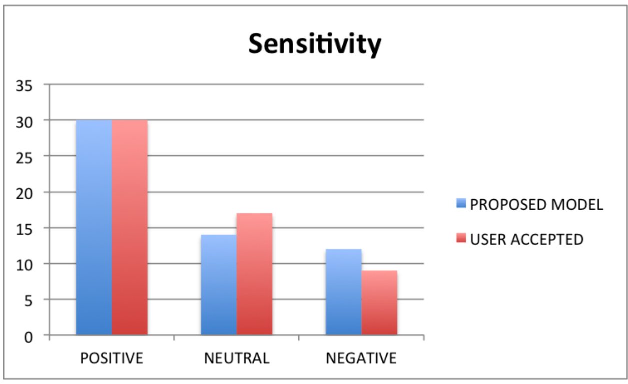
	Success Rate		
Sentiment	No margin	10% margin	20% margin
Sensitivity	50.00	67.86	80.36
Aptitude	71.43	80.36	87.50
Attention	73.21	92.86	98.21
Pleasantness	58.93	73.21	78.57

By applying these different margins, it is observed that success rate increases considerably. However, the application of these margins is not considered for sentiments of opposites signs in same categories, although they are in the margin threshold. Therefore in case the algorithm classifies a low intensity positive sentiment as a low intensity negative emotion it would be considered as a fail.

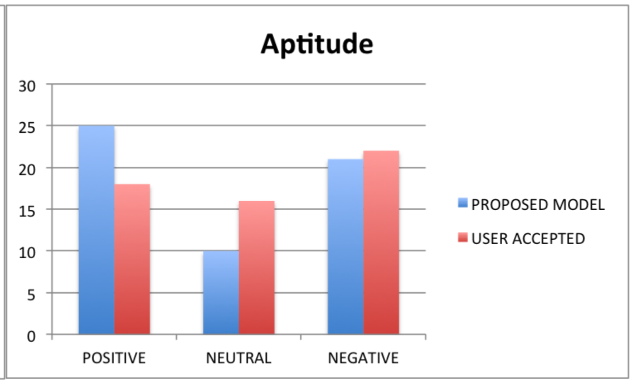
Nevertheless, the use of margins should be restricted to some limits as increasing the margin too much may distort model success rate. As a consequence, applying the 10% margin may be appropriate during the training phase of the model, and may be decreased once the classifier is trained, as long as results should be more accurate.

Moreover, in order to analyse the sample used for training the algorithm, some graphs showing the actual distribution of texts among categories are presented. Figure 5.1a to 5.1d describes the texts classifications by categories comparing the results outputted by the proposed model and the real sentiment values, provided by users either accepting algorithm results or setting their owns. As it can be observed, non of the categories shows a balanced distribution, so it can be stated that the training sample used for training this model is biased.

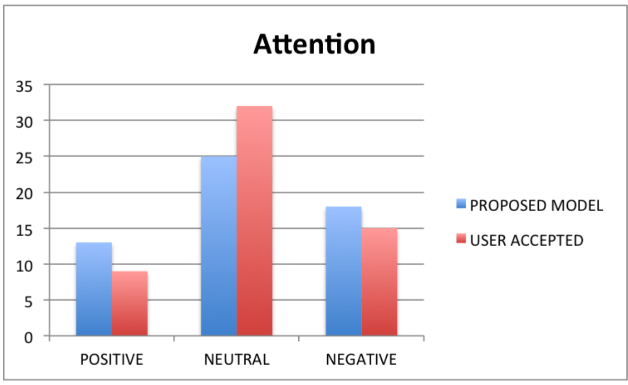
Figure 5.1: Intensity distribution of texts by category results



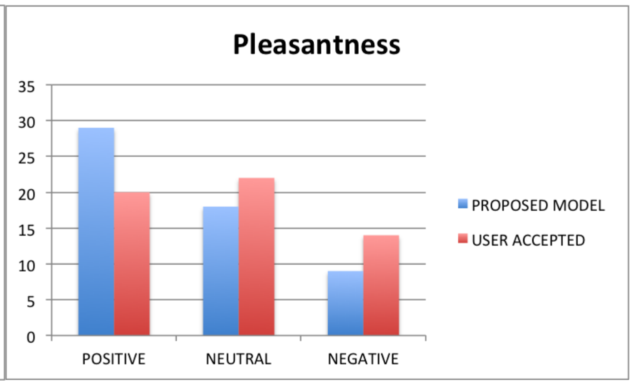
(a) Distribution among Sensitivity



(b) Distribution among Aptitude



(c) Distribution among Attention



(d) Distribution among Pleasantness

Observing the results from the presented tables and figures it may be stated that the proposed model does not perform as good as other sentiment analysis approaches. However, this fact cannot be evaluated with the given training set. The number of texts used for training this model is about fifty, which is too low. Therefore, the results of the evaluation at this stage of the training process may be not accurate enough and as a consequence, not representative of the overall performance of the model. In order to properly train the model about 10,000 samples should be used. Furthermore, category classification distributions should be balanced so that approximately the same amount of positive, negative and neutral samples are provided for each category.

The main reason which has led to this situation is the lack of available corpora. No publicly accessible corpora that matches the requirements of the model were found during the development of the project. Only some corpora with different categories were found, although they were not valid for the evaluation process of this project.

In order to cope with the drawbacks of not having a single corpus with a significant amount of annotated texts so that the model can be trained, some alternative solutions are studied.

- Building our own corpora may appear to be the simplest solution, although the amount of time required to build a balanced corpus with good sentiment annotations makes this solution unfeasible for this project. Moreover, the need of expert psychologists to supervise the building process turns this option very expensive and therefore, impossible to execute due to the limited budget.
- Trusting on users to train the algorithm may be a good option. It is totally free and no specific skills are required provided that the normal use of the mobile app will allow the the Knowledge Base to improve without users' notice. However, the risk of users giving up the prototype due to the sentiment analysis results computed in the early stages of the training process is high, as the model may not be accurate before training process is finished. Furthermore, trusting on users to train the model during the beginning of training process may be dangerous as some users may try to introduce malicious data so that instead of evolving concepts towards the real emotions they evoke, an involution may arise.
- Determining a *Beta* phase limited to certain kind of users during training process is the last alternative solution. Setting bounds to the number of *trainers* or beta users and having control to the type of users accessing the prototype may guarantee that no malicious inputs are found. The users that may suit best the role of trainer are researchers in Sentiment Analysis field and psychologists specialized in emotion classification. However, the success of a beta phase for training the proposed model with this sort of trainers lie in generating enough interest over the project so that people related with emotion studies become engaged with the model or the proposed approach.

Taking in account the budget and time restrictions of the project the only feasible solution for solving the lack of available corpora is opening a *limtied beta phase*. Therefore, after training process finishes, the overall performance of the model should be re-evaluated to determine whether the proposed approach towards Sentiment Analysis can compete with other studies in the field.

5.2 Questionnaire

The second approach towards project evaluation focuses on users opinions after using the developed prototype. In order to measure average user satisfaction with the developed mobile app and the results of texts classification, users are asked to answer a questionnaire so that they can express their opinions about several aspects of the application. This second sort of evaluation process focuses on subjective data, contrary to the objective data considered in the performance evaluation. However, although the first approach analyses data extracted from model analysis and users corrections, it is also conditioned by the subjectivity of the users while recognizing emotions.

The survey is available online at [55] and is composed of 10 questions asking about experience with technologies related to the project, app usability and sentiment classification results. Figures 5.3 and 5.4 are screenshots of the Google Form [56] used for creating the satisfaction survey.

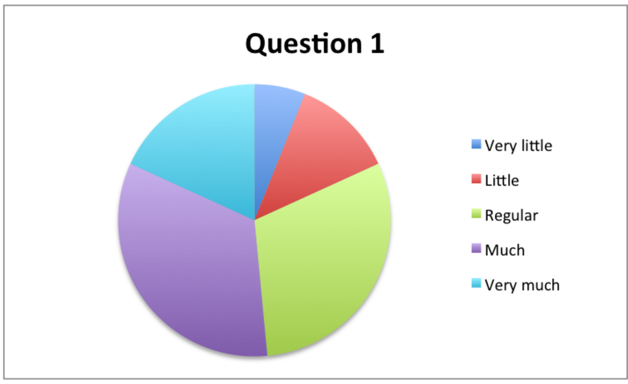
Questionnaire answers are automatically collected in a spreadsheet so that results can be used to extract statistical information. Figures 5.2a to 5.3f contain statistic graphs for each of the questions included in the satisfaction survey. As it can be observed in the answers of the three first questions, people participating in the survey are an heterogeneous sample, despite only 33 participants answered the questionnaire. Not all of them are common users of other image-sharing social networks although almost all of them are used to smartphones and Android OS. Furthermore, according to Figure 5.3a about 90% of participants know nothing or almost nothing about Sentiment Analysis.

Having a look at Figures 5.2d and 5.3d it can be observed that most users found the application usable and not very complex to use it. This fact, considering that the developed Android app is a prototype, can be considered as good news. Nevertheless, despite most of the participants agree with the set of sentiments used for representing emotions, almost half of them would prefer to have a larger list of sentiments so that more specific emotions are detected.

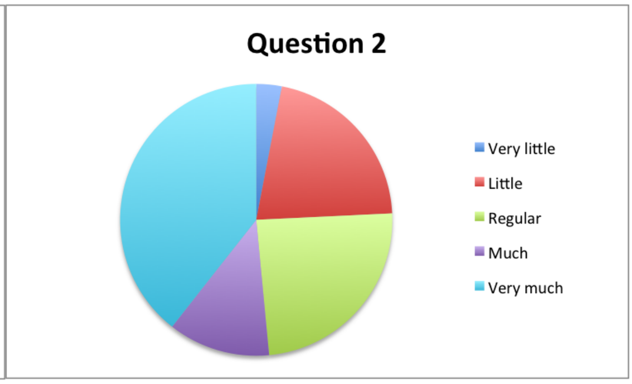
Regarding accuracy, most of the users agreed that the overall performance of the analysis of their texts were normal to great. Thus, taking in account the training stage of the classifier, may lead us to think that the overall performance of the trained classifier may be as good as other models for Sentiment Analysis.

Finally, looking at Figures 5.3e and 5.3f shows that most of the users would like the prototype to be finished an indeed, they would consider recommending the app to other people. Therefore, the assumption about using a training method based on popular social-network trends is proven to be correct.

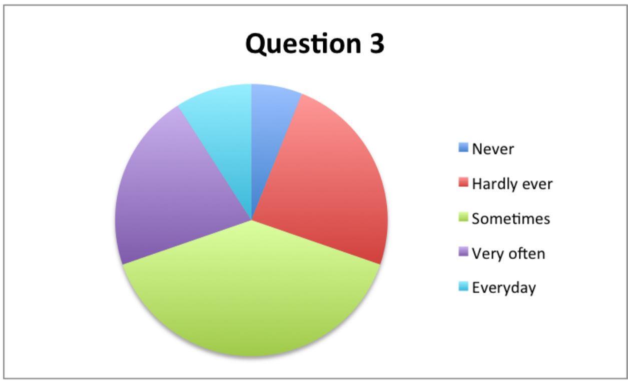
Figure 5.2: Questionnaire results. Questions 1 to 4



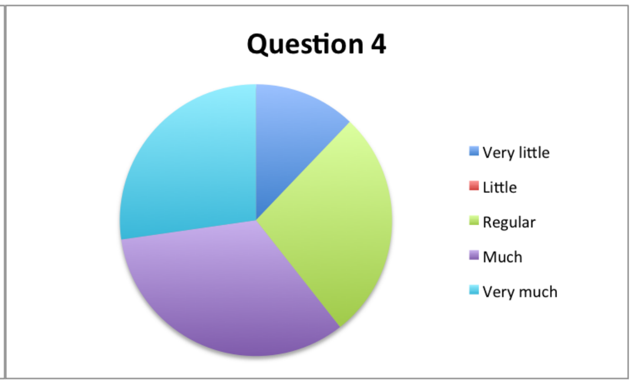
(a) Answers to Question 1 of the survey



(b) Answers to Question 2 of the survey

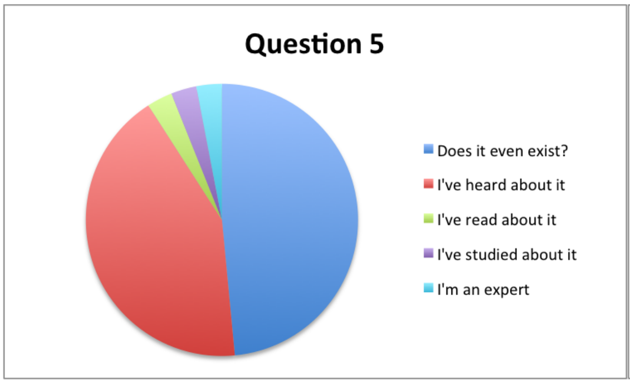


(c) Answers to Question 3 of the survey

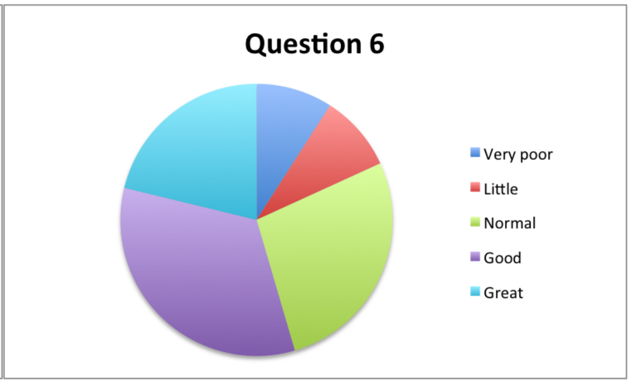


(d) Answers to Question 4 of the survey

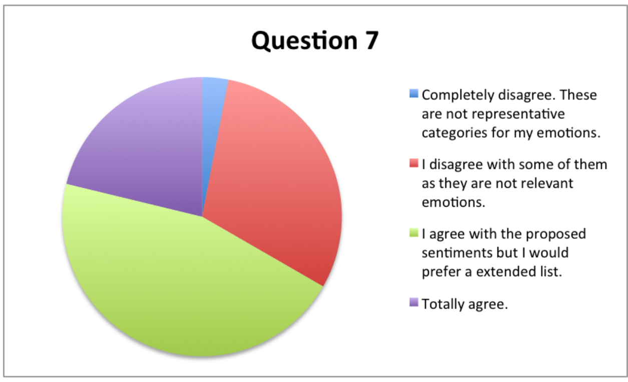
To this extent, it can be stated that the overall performance of the algorithm is more or less accepted by users in a very early stage, so as long as the classifier finishes its training period, the degree of acceptance will probably increase. Furthermore, the development of a social network application for training purposes has been a success, and finishing and launching the application to the the whole public may be considered.



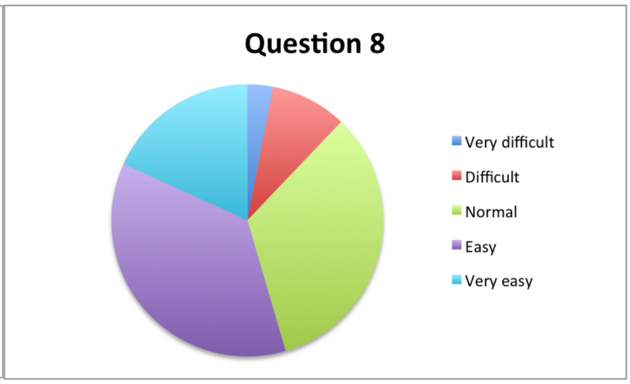
(a) Answers to Question 5 of the survey



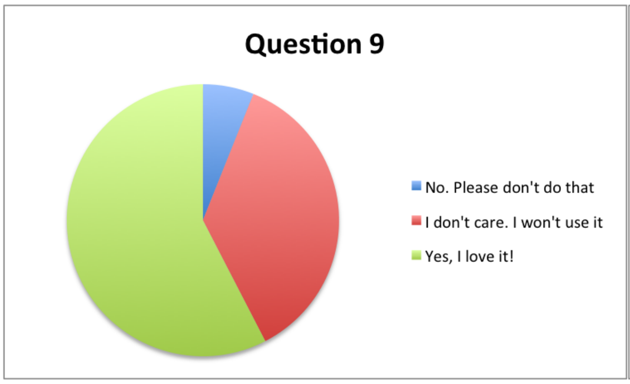
(b) Answers to Question 6 of the survey



(c) Answers to Question 7 of the survey

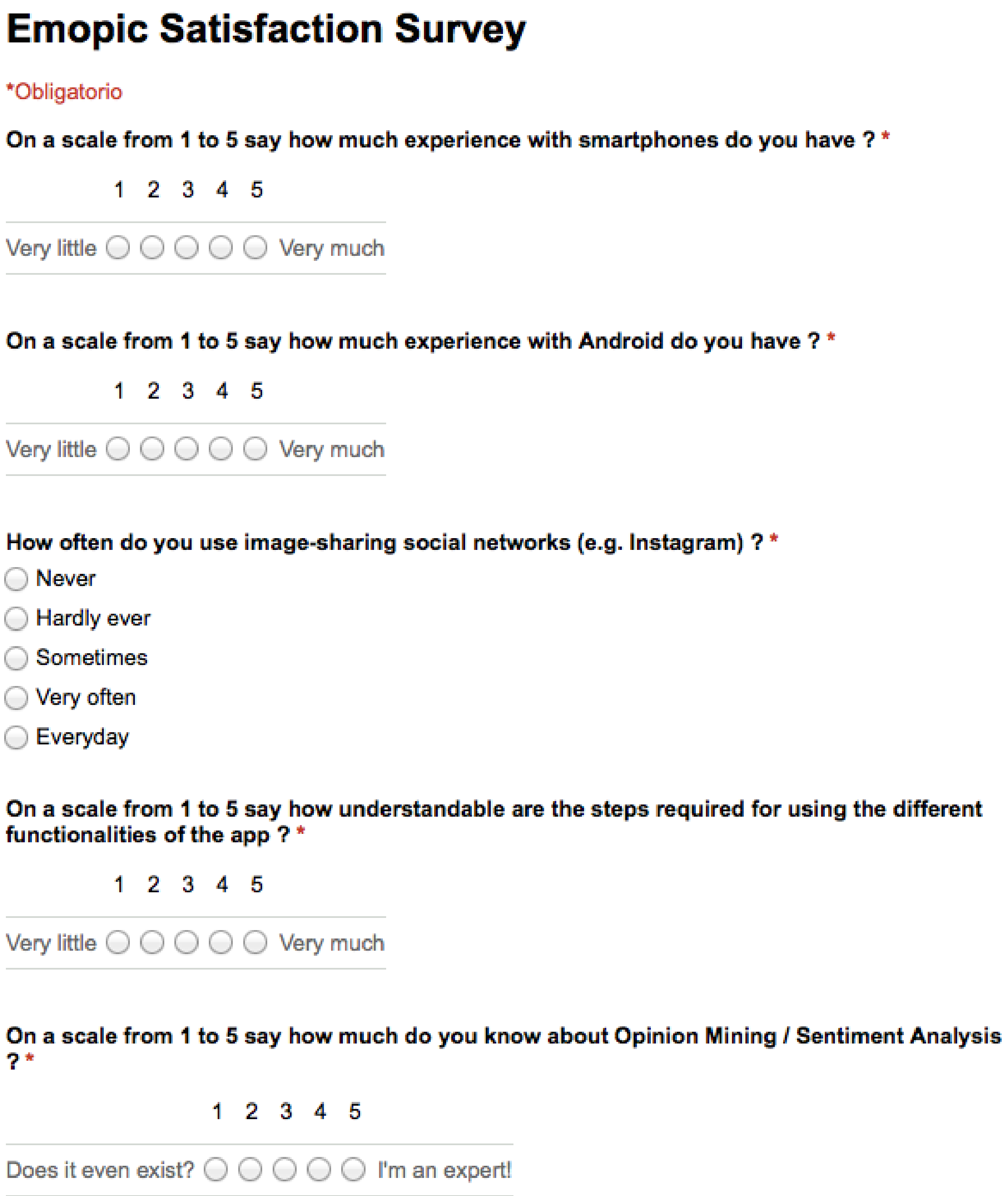


(d) Answers to Question 8 of the survey



(e) Answers to Question 9 of the survey

(f) Answers to Question 10 of the survey



Emopic Satisfaction Survey

*Obligatorio

On a scale from 1 to 5 say how much experience with smartphones do you have ? *

	1	2	3	4	5	
Very little	○	○	○	○	○	Very much

On a scale from 1 to 5 say how much experience with Android do you have ? *

	1	2	3	4	5	
Very little	○	○	○	○	○	Very much

How often do you use image-sharing social networks (e.g. Instagram) ? *

- ○ Never
- ○ Hardly ever
- ○ Sometimes
- ○ Very often
- ○ Everyday

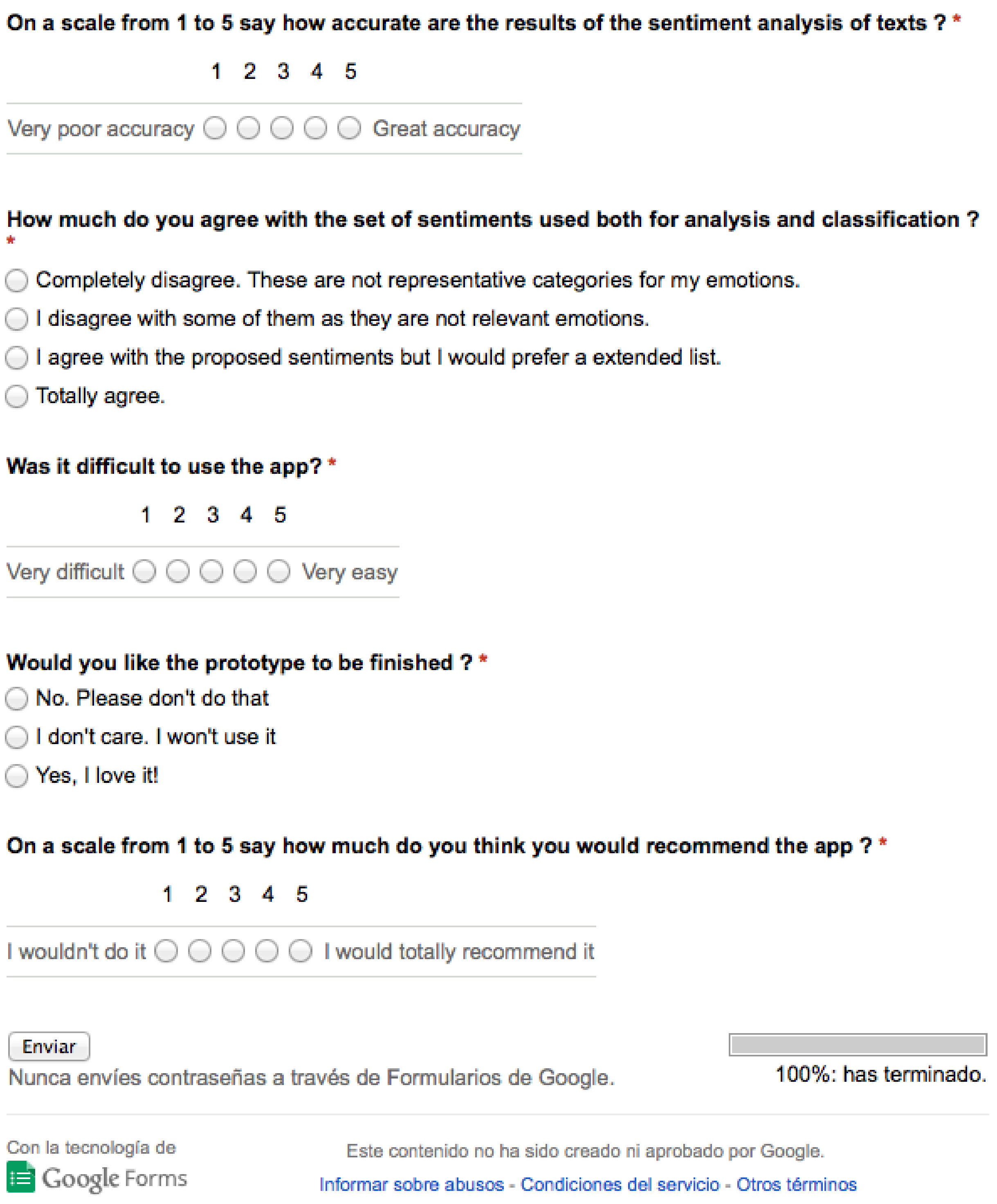
On a scale from 1 to 5 say how understandable are the steps required for using the different functionalities of the app ? *

	1	2	3	4	5	
Very little	○	○	○	○	○	Very much

On a scale from 1 to 5 say how much do you know about Opinion Mining / Sentiment Analysis ? *

	1	2	3	4	5	
Does it even exist?	○	○	○	○	○	I'm an expert!

Figure 5.3: Emopic Satisfaction Questionnaire Part 1



On a scale from 1 to 5 say how accurate are the results of the sentiment analysis of texts ? *

	1	2	3	4	5	
Very poor accuracy	◯	◯	◯	◯	◯	Great accuracy

How much do you agree with the set of sentiments used both for analysis and classification ? *

- ◯ Completely disagree. These are not representative categories for my emotions.
- ◯ I disagree with some of them as they are not relevant emotions.
- ◯ I agree with the proposed sentiments but I would prefer a extended list.
- ◯ Totally agree.

Was it difficult to use the app? *

	1	2	3	4	5	
Very difficult	◯	◯	◯	◯	◯	Very easy

Would you like the prototype to be finished ? *

- ◯ No. Please don't do that
- ◯ I don't care. I won't use it
- ◯ Yes, I love it!

On a scale from 1 to 5 say how much do you think you would recommend the app ? *

	1	2	3	4	5	
I wouldn't do it	◯	◯	◯	◯	◯	I would totally recommend it

Enviar

Nunca envíes contraseñas a través de Formularios de Google.

100%: has terminado.

Con la tecnología de Google Forms

Este contenido no ha sido creado ni aprobado por Google.

Informar sobre abusos - Condiciones del servicio - Otros términos

Figure 5.4: Emopic Satisfaction Questionnaire Part 2

Chapter 6

Conclusion and future work

In this section the author's conclusions are presented considering both the actual state and performance of the proposed model, and further improvements that may be implemented in order to increase the overall performance of the classifier.

6.1 Personal conclusion

Taking in account the performance results presented in *Evaluation* chapter of this project, the proposed model seems to be a promising alternative for Sentiment Analysis, as long as it provides good success rates per category although the general performance is not so significant. However, considering the lack of training of the classifier, it is expected to keep on improving through the use of the prototype.

Indeed, the learning feature of the proposed approach is on of the key elements of the model. The ability of adjusting concepts based on user criteria guarantee that concepts included in knowledge will end up converging with the real emotions each concept -*(word, POS)*- evokes, provided there is no malicious input,.

Moreover, one of the most important differences between this model and other Knowledge Base approaches to Sentiment Analysis is the way of building the knowledge. Considering both approaches to KB construction, the current amount of concepts represented in the knowledge (12297 entries) would increase significantly -even up to more than 50,000 terms- giving that only 14% of concepts from *SentiWordNet* lexicon were included in knowledge due to *ConceptNet* availability issues. As a consequence, the chance of including words from many different contexts would increase, and therefore a more general model will be achieved so that results over texts related to popular topics and those for particular contexts will be more homogeneous.

Regarding the developed prototype, it seems to back the assumption of taking advantage of social network popularity to test and train the proposed model. Moreover, the fact of being a fresh social network concept that is based on already top applications, but including some new features that no other product offers, turns this prototype into an attractive image-sharing application that group images by sentiments instead of common categories usually used by other services.

In conclusion, the proposed model seems to be a real alternative to most common machine learning approaches, by combining both the advantages of Knowledge Base systems and the learning-through-usage principle used by machine learning schemes. Although the proposed model is in early stages an many of the modules can be optimized, both the results collected from usage and users opinions show that the model can output accurate results. Despite not having an available corpora for properly testing the performance of the project, the tested examples shows that with a significant training sample global model results will improve. However, there is a lot of work to do in order to achieve success rate of other approaches to Sentiment Analysis which are in a more mature stage.

6.2 Future work

Apart from training the proposed model, there are many improvements which may refine the Sentiment Analysis results of the project. In this section some of the most relevant optimizations which can be applied are presented.

6.2.1 Extend lexicon

Although the number of concepts stored in our knowledge after finishing *ConceptNet* approach would be over 50,000 terms, the knowledge would not be complete. In order to construct even a more faithful representation of the language, some other lexicons should be explored so that the concept list is extended. For instance, including *SenticNet* lexicon into the project model would double the number of available concepts.

6.2.2 Topic detection

Topic detection is probably one of the features included in this project which can be improved most. Apart from counting occurrences, taking advantage of the information available in common-sense networks would provide a much more complete mechanism for detecting text's topics. For instance, storing hypernyms, hyponyms, and synonyms information of nodes in *ConceptNet* in our Knowledge Base, would allow to detect which whether an entity is an animal, an object or a person and even to know the relations between other entities so that it can be checked, for instance, if two entities which are animals belong to the same animal category.

6.2.3 Target - Holder

Apart from improving the topic detection feature, some modifications in the way targets of sentiments are identified may be implemented. For instance, storing specific rules for each of the verbs so that the direction of the action expressed by the verb can be detected, would make easier the process of recognizing which of the enitites related to an action is the holder of the sentiments and who is the target. This improvement would lead to changes in the weighting process as entities relevance would not be calculated through occurrences of all entities, but only restricted to the targets of emotions.

6.2.4 N-gram support

Including n-grams both in the Knowledge Base and in the parsing process may guarantee many expressions not to be split in single words but considered in a whole. Therefore, the sentiments related to this n-grams may vary significantly with respect to the actual computation of sentiments by combining the emotions evoked by each of the words of this expressions. Thus, a pattern matching process should be run before single words are analysed to allow common structures to be analysed as a group.

6.2.5 Languages

Last but not least, once some of the other improvements are implemented, it would be interesting to include support for other languages as most Sentiment Analysis projects are focused on English language. Therefore, porting this approach to other languages would allow researchers to test the proposed models in languages with different sentences structures.

Glossary

API
: API stands for Application Programming Interface. It is a specification of how software components of a system can interact with others systems of components.

Blob
: It stands for Binary Large Objects and are elements used by databases to store big data such as multimedia files.

Bug
: A software bug is an error in a computer system that causes unexpected behaviour or output, or even a crash in the system.

DoS Attack
: Denial-of-service attack is a computer attack which aims to make unavailable either a computer or a network. The interruption of services is done through sending huge amounts of requests to the system hosting the service so that it cannot respond respond to all of the, and the system collapses.

JSON
: JSON stands for JavaScriipt Object Notation. It is a lightweight format for exchanging information based on JavaScript language.

MVC
: Model-View-Controller is a software architecture pattern based on three layers. Model component represents the application data, which is presented to the users through the View layer and modified by Controller component.

N-gram
: It is a contiguous sequence of n items from a given sequence of text or speech.

Servlet
: It is a Java class used to extend the capabilities of servers that can be accessed by a host application via a request-response programming model.

SVM
SVM stands for Support Vector Machine, and are non-probabilistic binary linear classifier used for mapping inputs to one of two categories.

URL
It stands for Uniform Resource Locator, also known as web address. It is a character string that points to a resource in the Internet, for instance a webpage or an email.

VPS
Virtual Private Server is a virtual machine which runs an operating system and is sold in the Internet as a service so that customers hiring this service get *root* access to the server.

XML
XML stands for eXtensible Markup Language and is the most popular markup language. It is developed by the World Wide Web Consortium (W3C) and used for both saving and exchanging data.

Bibliography

[1] R. Plutchik and H. Kellerman, *Emotion: Theory, Research and Experience. Vol.1, Theories of Emotion*, vol. 1. New York: Academic Press, 1980.

[2] A. Rajaraman and J. D. Ullman., *Mining of Massive Datasets*. Cambridge University Press, 2011.

[3] P. I. Project, "Social media use by age group over time." http://www.pewinternet.org/data-trend/social-media/social-media-use-by-age-group/. Accessed: 2014-06-20.

[4] "Eurostat - e-commerce by enterprises: summary of eu aggregates." http://epp.eurostat.ec.europa.eu/portal/page/portal/eurostat/home/. Accessed: 2014-06-20.

[5] "Eurostat - individuals having ordered/bought goods or services for private use over the internet in the last three months." http://epp.eurostat.ec.europa.eu/tgm/table.do?tab=table&init=1&plugin=1&language=en&pcode=tin00067. Accessed: 2014-05-20.

[6] "2012 annual retail trade report," tech. rep., U.S. Census Bureau, March 2014.

[7] "Dashboards — android developer." http://developer.android.com/about/dashboards/index.html. Accessed: 2014-06-10.

[8] "Geocities." http://geocities.yahoo.com/. Accessed: 2009-10-01.

[9] "Theglobe.com." http://www.theglobe.com/. Accessed: 2009-10-01.

[10] "Sixdegrees.com." sixdegrees.com. Accessed: 2001-01-01.

[11] "Friendster." http://www.friendster.com/. Accessed: 2014-06-20.

[12] "Myspace." https://myspace.com/. Accessed: 2014-06-20.

[13] "Linkedin." https://es.linkedin.com/. Accessed: 2014-06-20.

[14] "Twitter." https://twitter.com/. Accessed: 2014-06-20.

[15] "Facebook." https://www.facebook.com/. Accessed: 2014-06-20.

[16] "Instagram." `http://instagram.com/`. Accessed: 2014-06-20.

[17] "Yahoo." `https://es.yahoo.com/`. Accessed: 2014-06-20.

[18] "Facebook annual report 2013," tech. rep., Facebook, Inc., 2013.

[19] M. Gill, Z. D. Wigder, M. O'Grady, and D. Roberge, "European online retail forecast, 2012 to 2017," tech. rep., Forrester Research, Inc., March 2013.

[20] K. Ahmad, *Affective Computing and Sentiment Analysis: Emotion, Metaphor and Terminology*, vol. 45. Springer Publishing Company, Incorporated, 2013.

[21] C. Darwin, *The Expression of the Emotions in Man and Animals*. Appleton, New York, 1872.

[22] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[23] R. Niese, A. Al-Hamadi, M. Heuer, B. Michaelis, and B. Matuszewski, "Machine vision based recognition of emotions using the circumplex model of affect," in *Multimedia Technology (ICMT), 2011 International Conference on*, pp. 6424–6427, July 2011.

[24] M. M. Bradley, M. K. Greenwald, M. C. Petry, and P. J. Lang, "Remembering pictures: Pleasure and arousal in memory," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 18, pp. 379–390, March 1992.

[25] A. Mehrabian, *Basic Dimensions for a General Psychological Theory: Implications for Personality, Social, Environmental, and Developmental Studies*. Social Environmental and Developmental Studies, Oelgeschlager, Gunn & Hain, 1980.

[26] E. Cambria, A. Livingstone, and A. Hussain, "The hourglass of emotions," in *Cognitive Behavioural Systems* (A. Esposito, A. Esposito, A. Vinciarelli, R. Hoffmann, and V. Müller, eds.), vol. 7403 of *Lecture Notes in Computer Science*, pp. 144–157, Springer Berlin Heidelberg, 2012.

[27] R. W. Picard, "Affective computing," 1995.

[28] M. Hoque, M. Yeasin, and M. Louwerse, "Robust recognition of emotion from speech," in *Intelligent Virtual Agents* (J. Gratch, M. Young, R. Aylett, D. Ballin, and P. Olivier, eds.), vol. 4133 of *Lecture Notes in Computer Science*, pp. 42–53, Springer Berlin Heidelberg, 2006.

[29] T. Pfister and P. Robinson, "Speech emotion classification and public speaking skill assessment," in *Human Behavior Understanding* (A. Salah, T. Gevers, N. Sebe, and A. Vinciarelli, eds.), vol. 6219 of *Lecture Notes in Computer Science*, pp. 151–162, Springer Berlin Heidelberg, 2010.

[30] P. Ekman and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, 1978.

[31] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 677–695, July 1997.

[32] P. Petrantonakis and L. Hadjileontiadis, "Emotion recognition from eeg using higher order crossings," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, pp. 186–197, March 2010.

[33] F. H. Khan, S. Bashir, and U. Qamar, "Tom: Twitter opinion mining framework using hybrid classification scheme," *Decis. Support Syst.*, vol. 57, pp. 245–257, January 2014.

[34] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing*, vol. 10 of *EMNLP '02*, (Stroudsburg, PA, USA), pp. 79–86, Association for Computational Linguistics, 2002.

[35] J. Wiebe and E. Riloff, "Creating subjective and objective sentence classifiers from unannotated texts," in *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'05, pp. 486–497, Springer-Verlag, 2005.

[36] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the Workshop on Languages in Social Media*, (Stroudsburg, PA, USA), pp. 30–38, Association for Computational Linguistics, 2011.

[37] R. Valitutti, "Wordnet-affect: an affective extension of wordnet," in *In Proceedings of the 4th International Conference on Language Resources and Evaluation*, pp. 1083–1086, 2004.

[38] E. Cambria, R. Speer, C. Havasi, and A. Hussain, "Senticnet: A publicly available semantic resource for opinion mining," in *AAAI Fall Symposium: Commonsense Knowledge*, vol. FS-10-02 of *AAAI Technical Report*, AAAI, 2010.

[39] G. A. Miller, "Wordnet: A lexical database for english," *Commuications of the ACM*, vol. 38, pp. 39–41, November 1995.

[40] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller, "Wordnet: An on-line lexical database," *International Journal of Lexicography*, vol. 3, pp. 235–244, 1990.

[41] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," in *5th Conference on Language Resources and Evaluation*, pp. 412–422, 2006.

[42] "Sentic computing." `http://sentic.net/sentics/`. Accessed: 2014-06-20.

[43] S. M. Mohammad and P. D. Turney, "Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion*

in Text, (LA, California), pp. 26–34, Association for Computational Linguistics, June 2010.

[44] H. Davis and S. M. Mohammad, “Generating music from literature.” In Proceedings of the EACL Workshop on Computational Linguistics for Literature, April 2014, Gothenburg, Sweden, March 2014.

[45] A.-R. Tsai, C.-E. Wu, R.-H. Tsai, and J. Hsu, “Building a concept-level sentiment dictionary based on commonsense knowledge,” *Intelligent Systems, IEEE*, vol. 28, pp. 22–30, March 2013.

[46] “Lexipedia.” `http://www.lexipedia.com/`. Accessed: 2014-06-20.

[47] “Conceptnet 5.” `http://conceptnet5.media.mit.edu/`. Accessed: 2014-06-20.

[48] “Probase.” `http://research.microsoft.com/en-us/projects/probase/`. Accessed: 2014-06-20.

[49] M. Koppel and J. Schler, “The importance of neutral examples for learning sentiment,” in *In Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations (FINEXIN)*, 2005.

[50] “Open xerox.” `http://open.xerox.com/`. Accessed: 2014-06-20.

[51] “Conceptnet 5 - wiki.” `https://github.com/commonsense/conceptnet5/wiki`. Accessed: 2014-06-20.

[52] “Saif: Publications.” `http://www.saifmohammad.com/WebPages/ResearchInterests.html`. Accessed: 2014-06-20.

[53] “Penn treebank notation.” `http://www.cis.upenn.edu/~treebank/`. Accessed: 2014-06-20.

[54] “The free dictionary.” `http://www.thefreedictionary.com/`. Accessed: 2014-06-20.

[55] “Emopic satisfaction survey.” `https://docs.google.com/forms/d/1qzb-kkMHGItYecz-3vUYVqsm_D2vBqRx2Zk3wBSCeoE/viewform?pli=1&edit_requested=true`. Accessed: 2014-06-20.

[56] “Google docs.” `https://docs.google.com/?hl=es`. Accessed: 2014-06-22.