Deusto

Facultad de Ingeniería
Universidad de Deusto

Ingeniaritza Fakultatea
Deustuko Unibertsitatea

Grado en Ingeniería Informática
Informatikako Ingeniaritzako Gradua

Proyecto fin de grado
Gradu amaierako proiektua

Automatic detection of repressed anger from text messages

Jesús María Sesma Solance

Director: Takashi Yukawa

Nagaoka, Mayo de 2017

Abstract

The growth in the usage of social media, microblog and review platforms has resulted in a significant increase of the access to short text messages that reflect individual's opinion and feelings. Automatic detection of people's emotions has a wide range of applications such as producing systems that measure the satisfaction of customers and thus help companies to improve their products or services. This research project focuses on detecting anger, an emotion that is relative difficult to detect compared to other sentiments due to the usage of linguistics figurative language techniques, such as irony, that intends to communicate the opposite of what it is literally said. To this purpose, a review of the state of the art has been made and an experiment using not only traditional machine learning techniques, such as Neural Networks, K-Nearest Neighbor and Support Vector Machines, but also Deep learning algorithms has been conducted in an open social network like Twitter. The proposed methods define the repressed anger detection as a classification problem and to solve it, the task is divided into two subtasks that complement each other. The first focuses on explicit anger detection, while the second subtask's goal is to detect irony. The system make use of selected featured based on characteristics properties of English language and studies emotion and irony in psychology. The model is composed on features such as: frequency of words, style in written and spoken languages, intensity of adverbs and adjectives, structure of the document, use of emoticons, synonymy, ambiguity and the contrast of sentiment and negative situation.

Keywords

Emotion Analysis, Supervised Classification, One-vs-one Classification, Convolutional Neural Networks, Repressed Anger Detection

Contents

List of Figures

Chapter 2

Chapter 3

Chapter 4

Chapter 5

List of Tables

Chapter 3

Chapter 5

1. INTRODUCTION

- Introduction
    - Get the important points of the state of art referring to sentiment analysis, repressed anger detection.
    - Explain the hypothesis to be worked out in the thesis.
- Planning and Methodology [OK]
    - Explain the plan. [OK]
    - Explain the methodology. [OK]
    - Show the schedule. [OK]
- State of art
    - Explain that sentiment analysis is.
    - Explain what emotion analysis is, compared to previous point.
    - Explain previous work.
        - Previous work in anger detection.
        - Previous work in irony detection.
    - Explain the that the project will focus on resolve the issue as a text classification problem.
        - Classification Techniques
        - Fundamentals of Classification
        - General classification problem solving
        - Explain the algorithms I used to make tests in WEKA.
        - Explain Deep Learning.
            - Explain Convolutional neural networks.
- Development
    - Early development on the exploration phase.
        - WEKA text classification testing.
    - Explain that on SemEval they said that the best scoring system were using DL and is becoming trending. General Problem Solving techniques (NPL) in Deep Leaning perform better that those that focus on resolving in depth focusing on the topic.
        - Change to Deep Learning Development.
        - Explain the usage of architecture that Yoon Kim uses. (Previously explained in SoA)
        - Explain the process to detect repressed anger. (The system I made starting from dataset merging, ending with merge of 2 classification output.) Simple diagram.
            - Dataset searching and generation.
              –> Developed a tweet downloader for IDs.
            - Word2vec (needed for CNN)
              –> Model used for Word2vec.

–> Spell checking

–> Slang dictionaries (difficulty on when to process the word. Detect Slang)

–> Usage of Stopwords.

· CNN classifiers

–> hyperparams used.

· Classification output merge.

–> Due to the labels in the dataset cannot use ensemble learning and use a 2x2 matrix to match everything.

–> Develop a automatic Google forms and how to process them back to the original dataset.

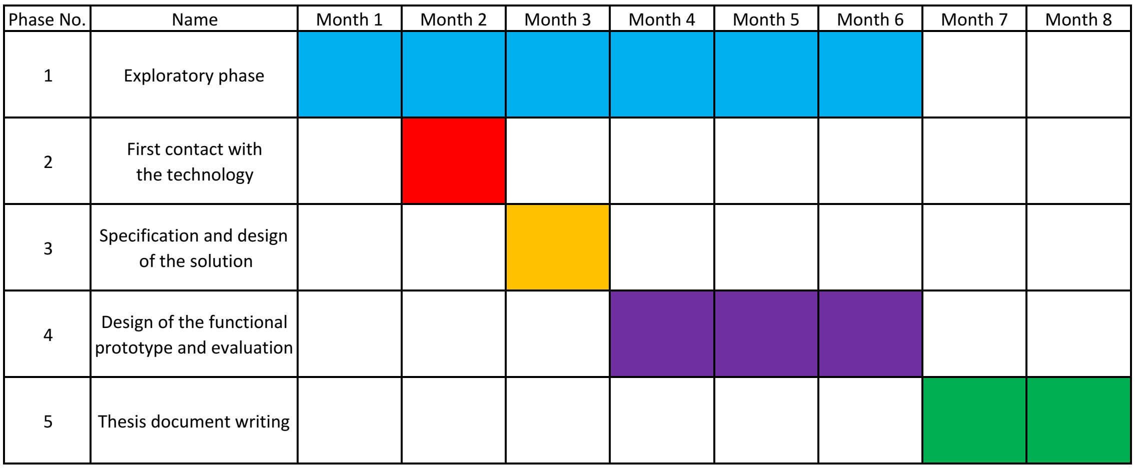
- Research Framework
  - Searching datasets, or create one, automatically.
  - Manual labeling used to final prediction process.
- Results
  - Google Model
    - Explain the final result plus each classification independently.
  - Twitter Model
    - Explain the final result plus each classification independently.
  - Spell checking
- Conclusion and Future Work
  - Different approach now that manual label data is obtained. (Semi-supervised learning)
- Self-assessment
  - writing.

2. PLANNING AND METHODOLOGY

In order to accomplish the Master's thesis during the established period of eight months, the following plan will be followed:

1. **Exploratory phase:** this period of time will consist in searching and studying the literature related to the research field in order to build a solid theoretical framework on which support the rest of the investigation. Even though this process is presented as a isolated task, it is an fundamental activity that will be executed continuously throughout the whole project.
2. **First contact with the technology:** after having reviewed the first phase of the state of art and checked the advantages or limitations of each architecture, the first empirical tests will be done. These experiments will serve to for a better understanding of the literature read in the exploratory phase and to observe how they behave with the gathered sample data.
3. **Specification and design of the solution:** in this phase, based on the knowledge gathered from the empirical tests in the last point and the relevant literature that has been constantly read, the requirements of the solution that best results can achieve for this research topic will be specified.
4. **Design of the functional prototype and evaluation:** after the requirements gathering, during this task, a prototype that justify all the steps followed this far will be developed, and evaluate its performance.
5. **Thesis document writing:** Finally all the relevant knowledge that has been required to complete this research will be gathered, the procedure to develop the system will be described and the conclusion and future work on this topic will be written down altogether, resulting in this master's thesis document, that will be refined until its laster submission and defense.

Previously defined tasks will be executed according to the next schedule diagram:



Phase No.	Name	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6	Month 7	Month 8
1	Exploratory phase	🟦	🟦	🟦	🟦	🟦	🟦		
2	First contact with the technology		🟥						
3	Specification and design of the solution			🟧					
4	Design of the functional prototype and evaluation				🟪	🟪	🟪		
5	Thesis document writing							🟩	🟩

Figure 2.1: Schedule

The figure 2.2 illustrates the methodology used during the investigation process, in which the results obtained from the prototypes might not be satisfactory or could lead to formulate new questions. To solve those doubts a redesign of the system may be needed. In addition, after reporting the results to project supervisor, new relevant information might be provided. This information has to be added to the knowledge obtained from previous state of the art review and reanalyze if the proposed solution requirements are still valid.

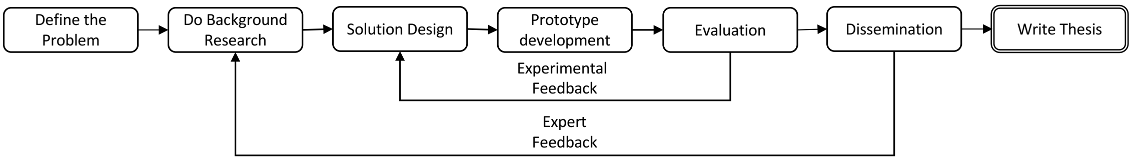


Figure 2.2: Methodology

3. STATE OF ART

The aim of this chapter is to conduct a study of the state of the art of automatic detection of repressed anger by first analyzing the current state of sentiment analysis and emotion detection, which are the fundamentals to detect anger and irony.

3.1. SENTIMENT ANALYSIS

Sentiment Analysis (SA), sometimes referred as Opinion Mining (OM), is the field of study that aims determine the attitude of the author respect to some topic. During the last decade, due mainly to the rapid increase of the social media, the research of this area, which includes linguistics and Natural Language Processing (NLP), has gain more relevance as it has a wide range of applications that can be used commercially.

Almost the human actions are influenced by opinions. A practical example of this behavior that get repeated in daily basics occurs when before making a decision we want to know other' to contrast our thoughts. In business, when an organization requires to public or consumer opinion, it conducted different types of surveys to gather this information. However, nowadays we do not precise to ask personally to people around us to discuss about a product. Thanks to microblogs, Twitter and reviews in social networks, we can broaden the amount of external opinions for decision making. Due to the amusing amount of information available, sometimes we could face some challenges, such as finding reliable sources summarizing and filtering the relevant information. The same way we consume this information, companies also search for new ways to obtain and process such information efficiently without human intervention, to achieve this automated analysis is needed[52].

3.1.1 Tasks

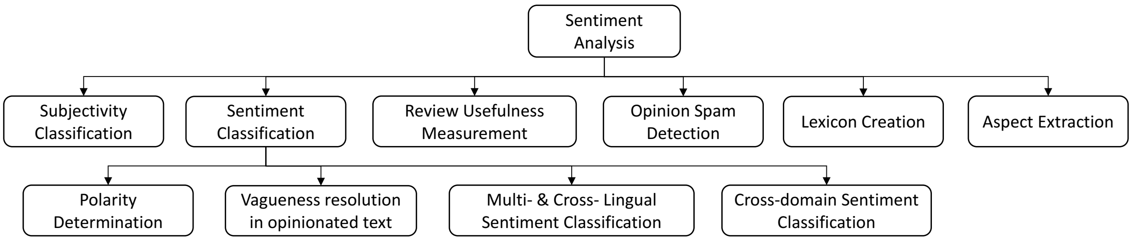

Sentiment Analysis
Subjectivity Classification
Sentiment Classification
Review Usefulness Measurement
Opinion Spam Detection
Lexicon Creation
Aspect Extraction
Polarity Determination
Vagueness resolution in opinionated text
Multi- & Cross- Lingual Sentiment Classification
Cross-domain Sentiment Classification

Figure 3.1: Sentiment analysis tasks.

Depending on the problem definition, different tasks have been defined related to sentiment analysis. According to [66], after the revision of more than three hundred papers related to this topic, the survey carried out concludes that the main tasks can be categorized as (a) subjectivity classification, (b) sentiment classification, (c) review usefulness measurement, (d) opinion spam detection, (e) lexicon creation and (f) aspect extraction, as shown in the figure 3.1.

Subjectivity classification

According to [57], subjectivity classification aims to determine the "private state" of the author of a text. The Subjectivity analysis is the process of distinguish objective language from the opinion oriented. Even though there is much less literature about this field compared to other SA task, it has proven to be more difficult than determine the measuring the polarity of a document and, thus, improvements achieved in this field will positively impact on sentiment classification.

Sentiment classification

Sentiment classification consist in determine the orientation of a sentiment of a given text into two or more classes. This classification has been performed in multiple classes, such as, binary (positive or negative), ternary (positive, neutral and negative), n-ary [60], among others.

Review usefulness measurement

Review usefulness measurement tends sometimes be confused with opinion spam detection, however there are some slight differences. Of course false review are always rated as useless, as their objective is just to boast or ridicule a product. Although, there are also bad reviews that might not necessary be spam, as they reflect the honest opinion of the author. Thus, this subtask complement the opinion spam detection.

Opinion spam detection

In order to boost sales, some companies have invested on unmoral marketing techniques that consist in publishing false reviews about their own products or from the competence. In order to detect fake reviews, researches have focus their effort on (a) analyzing the content of the review, finding hidden lies by using NLP, (b) meta-data of the review, such as start rating, author information, date of writing, among many others [40], however this information sometimes is not accessible, and (c) comparison with real-life information about the product.

Lexicon creation

A lexicon is a collection of words that determines the polarity of each one of them. Lexicons are used as an approach to detect the sentiment polarity of a given document [43].

Aspect extraction

Aspect extraction is a key to detect author's opinion regarding various characteristics of a product based on the collection of opinion words and entities and identification and extraction of the aspects of those entities [58].

3.1.2 Techniques

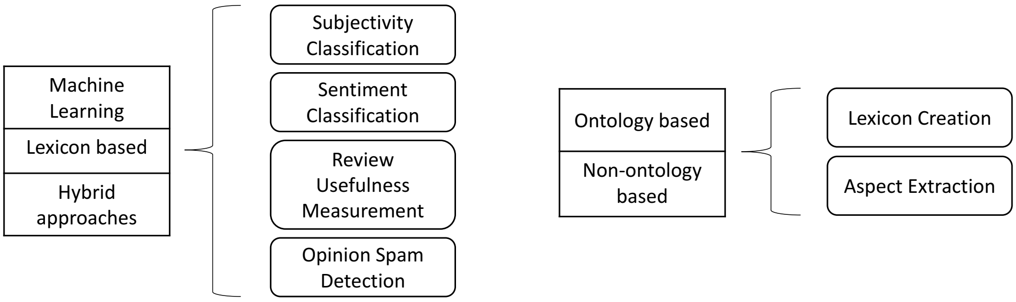


Figure 3.2: Sentiment analysis approaches for each task.

To solve all these problems in the most efficient way, multiple approaches have been conducted, such as [78]; [59] or [64]. The figure 3.2 illustrates which techniques have been used for each previously named tasks.

The figure 3.3 collects the most common techniques used for SA based the survey conducted by Walaa Medhat et al. (2014). For the purpose of this project, we will focus on making an introduction to those approaches related to sentiment classification, more precisely to Machine Learning (ML), lexicon-based and hybrid approaches.

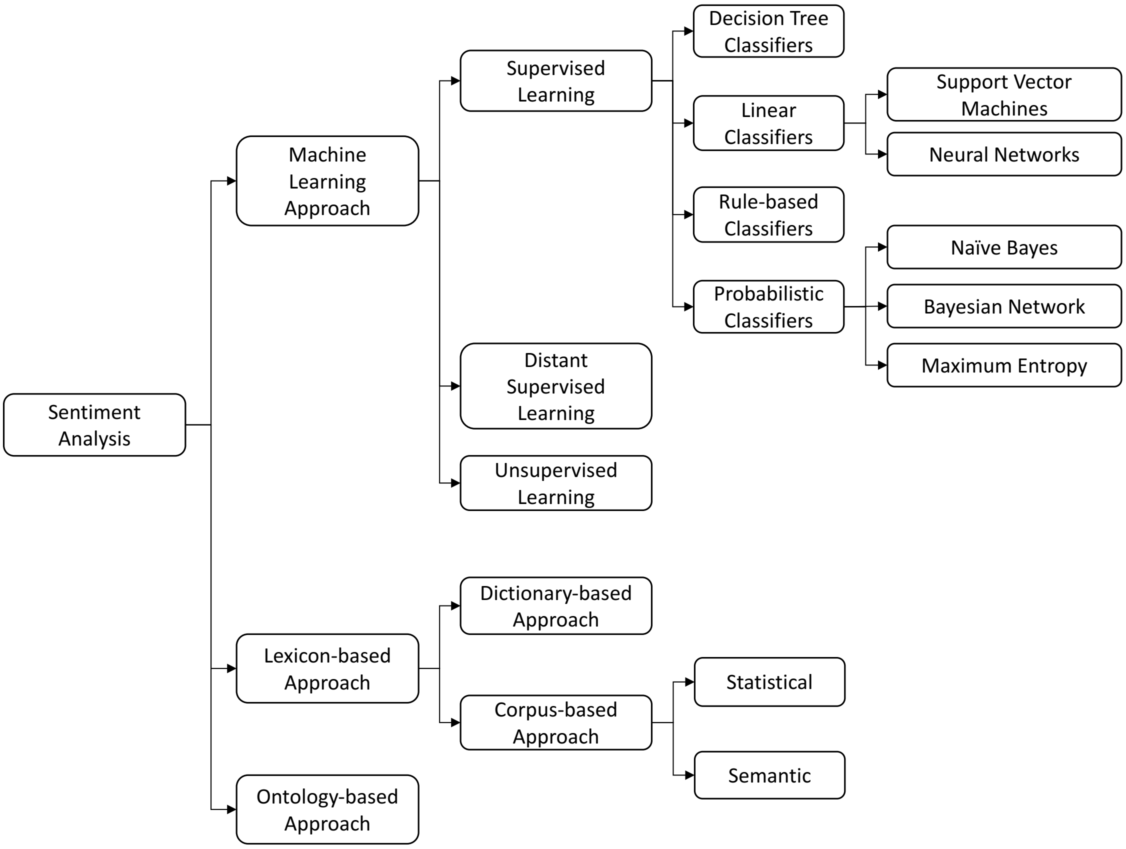


Figure 3.3: Sentiment analysis techniques[55].

Machine learning approach

Thanks to the good performance machine learning techniques have proved to achieve, more and more researches have gain interest on this technology. The classification process (see 3.4) used, consist on extracting a series of features vectors from the dataset and a to create a model that is able to identify different patters among theses features. The selection of relevant features is key as it directly affect the performance the model can score. Some good known features for model generation are based on n-grams, such as unigrams, bi-grams and tri-grams [24].

1. **Data gathering:** During this phase, first publicly available datasets are searched from related literature. Then if needed, to complement this data, an analysis of the social media that contains information regarding the research topic is conducted. After analyzed the sources from which data can be extracted, a crawler adapted to deal with those networks is used to collect the data. This data, usually mostly or partially, requires to be labeled with a predefined set of classes, a task generally done by crowd-sourcing.
2. **Pre-processing:** Most of the raw data obtained from social media contains information is not relevant or cannot be directly processed by the classifier. In this step, tasks like word spell checking, Uniform Resource Locator (URL) and hash-tags removal, keyword extraction, language filtering, tokenization, among many others are carried out.
3. **Classifier training:** With the dataset most commonly split into training, validation and test, the training and validation sets will be provided as an input to the classification algorithm as the learning process, which concludes with the generation of a model.
4. **Classification:** With the previously generated model on the learning process, the classifier can now be used to predict new data.
5. **Result evaluation:** To measure how accurate the learning algorithm performed, the output of the system is compared with dataset labels by means of a defined evaluation function and representation format.

Lexicon based approach

Sentiment analysis based on lexicon approach consist on the use of a collection of words, a dictionary, that contains a pre-tagged lexicon to classify a given text. The input document is tokenized and, thus, every token is compared with the lexicon forming the dictionary. Depending on how the lexicon is defined, the result of the comparison can be either positive or negative. If positive match is given, the total score of the text classification is increased, if negative match occurs, then the score decreases. Although this technique could be considered simplistic, some simplistic tests [38] have proven to be successful in achieving high precision.

Hybrid approach

As [77] suggests, both, machine learning and lexicon based approaches have their advantages and disadvantages. Thus, researchers have tried merging different methods to order to find more precise and efficient ways of classify sentiments. The combination of techniques are variated, such as architectures that mix lexicons to automatically label data that is used as a learning input into a Naive Bayes (NB) classifier as shown in [61]; the author of [45] uses the combination of three classifiers, enhanced emoticon classifier, improved polarity classifier and a SentiWordNet classifier to determinate the polarity of a given tweet or [65], which used multiple rule based classifiers approaches combined with SVM executed on multiple hybrid sequential executions on a semi-automatic dataset.

3.2. EMOTION DETECTION

Emotion detection has been always a important field of study in neuroscience and psychology. Several experiments have been conducted to recognize emotions from face expressions, voice and body gestures, such in [6]; [5] and [34]. However, these features are not always available and thus, the detection of emotion from text has gain relevance during the last decade.

Emotion detection from text documents is closely related to SA. As explained in 3.1.1, sentiment classification task consists in determine the polarity of a given text into two or more classes. Emotion detection also fits this description, since this field of study is conducted by classifying a given document into predefined emotion labels. The definition of these labels are based on previously presented emotion models, which consist on application psychological and neuroscientific theories to represent emotions. One of the first model defined was proposed by Paul Ekman, on 1972, composed by six basic emotions, *"happiness, sadness, fear, anger, surprise and disgust"*, and has been used in multiple face recognition systems and is the base emotion detection from text [36] as its definition has resulted to be the most semantically diverse [4].

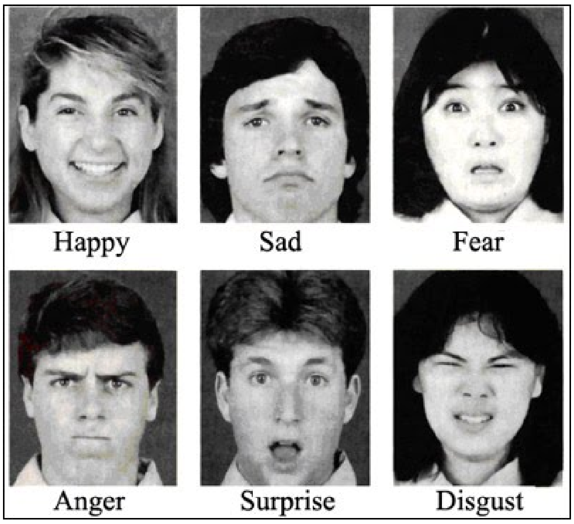


Figure 3.4: Ekamn's basic emotions, image extracted from Expressions, Emotions and Emblems, Google Sites.

Since then, more interpretations have been made regarding the representation of the emotions as shown in [13]. Ekman himself added his list 11 new emotions stating that not all of them could be represented by facial expressions. In 1980 Robert Plutchik created another bi-dimensional model (see figure 3.5), known as wheel of emotions, that defined eight basic bipolar emotions with different levels of intensification and could be combined among them. Ortony Clore and Collins (OCC) designed in 1988 a model in which emotions were driven by an agent, based on the premise that *"emotions are not themselves linguistic things, but the most readily available non phenomenal access we have to them is through language"* [8]. The model included 22 emotion categories that aimed to *"model humans in general"* [8]. OCC enables the study of emotions as classes that share the same feeling, instead of specific words, proving that this *"is a theory of emotions and not a theory of the language of emotions"* [8]. For more information, authors of [7] analyze the state of art of the emotion models and the techniques used to generate them.

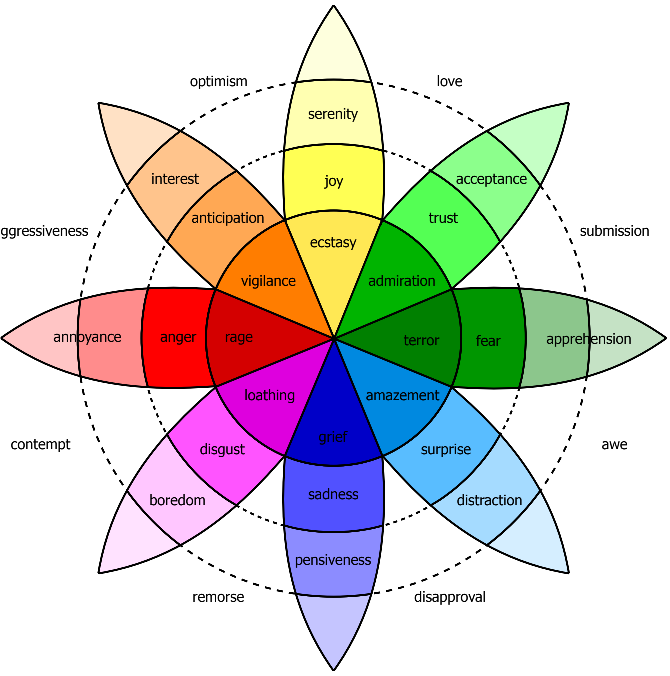


Figure 3.5: Plutchik's emotion wheel, image extracted from Wikipedia's public domain.

Emotion classification tasks shares the same principles as SA, and thus, the techniques used are mainly ML, lexicon-based and hybrid approaches previously explained in section 3.1.2.

3.3. PREVIOUS WORK

The study of anger and its management has been studied since ancient times. According to American Psychological Association (APA), anger is *"an emotion characterized by antagonism toward someone or something you feel has deliberately done you wrong"* [2]. Anger is a normal emotion all human experience. Modern psychologists have studied the health effects that may derivate from anger suppression, such as shown in [44] and [73], linking it to a more extremist and sarcastic personality. This research aims to find the answer to the question: "Can the repressed anger be detected in text?". Based on the definition of personality of a subject that has experienced repressed anger, the problem formulation has been simplified into finding negative messages influenced by anger that covered by a layer of sarcasm. The following subsections present a brief summary of the research conducted in the detection of anger and sarcasm.

3.3.1 Anger

As explained in section 3.2, the increase of the usage of social media and computational viability to process all the publicly available data has enable the detection of emotions from text. However as there is not literature that focuses on anger emotion, this section will briefly enunciate the approaches researched to classify emotions that contain anger as target label.

In 2007 during international workshop of evaluations of computational semantics analysis system, known as SemEval [71], presented an task that analyzed the affective text, in which the participants had to calculate the the valence and emotional class of news titles. Three groups participated in the emotion classification subtask: SWAT, UA and UPAR7.

The system proposed by SWAT was based on unigram trained model supervised learning. In addition to the emotion label target given this group perform a synonym expansion of these keywords by using the Roget Thesaurus. The training of the system was performed by combining the development data provided by the SemEval task organizers plus a set of 1000 headlines that was annotated from the group.

UPAR7 developed a linguistic rule-based system approach. The first step was to pre-process the data by un-capitalize the common words. Then, each word was rated separately with each emotion target label.

In order to determine the type of emotion in the headline UA group collected statistics from three web search engines to calculate the distribution of nouns, verbs, adjectives and adverbs of the headline. The emotion scores were obtained through Point-wise Mutual Information (PMI).

The F1-score obtained by SWAT, UA and UPAR7 for anger sentiment was 7.06, 16.03 and 3.02 respectively (results multiplied by 100) [75].

Authors in [74] implemented five system for emotion analysis based on a knowledge based emotion annotation and corpus based emotion annotation: WN-affect presence, LSA single word, LSA emotion synset, LSA all emotion words and NB trained on blogs. For anger annotation, the proposed system scored a F1-score of 16.77 (results multiplied by 100), outperforming the system proposed by UA in SemEval for anger annotation.

Kirk Roberts et al. developed a system composed of six SVM classifiers, one for each Ekman's sentiment basic emotion. The Features selected as input for the binary classifier where: unigrams, bigrams, trigrams, presence of interrogation or exclamation mark, Word Net synsets, WN hypernyms, Latent Dirichlet Allocation topic scores and high PMI unigrams (significant words). The anger classifier used unigrams, synsets, topis and significant words, scoring a F1-score of 0.642 [69].

Chew-Yean Yam posted at Microsoft developer Blog an example of emotion classification using Deep Learning (DL) techniques. The corpus was collected by using Amazon's Mechanical Turk to perform a Human Intelligence Task (HIT) to manually annotate the data, obtaining a total of 784,349 samples of informal short English messages classified as five classes, anger, sadness, fear, happiness and excitement. The DL architecture is described ad a Neural Network (NN) with five out nodes, three hidden layers, that contain 5, 25 and 125 nodes respectively. The loss function selected was Cross entropy with a stochastic gradient descent optimization algorithm. The learning rate was set to 0.001. The maximum training iteration was set to 100 with a greedy pre-trainer type executed though-out 25 epochs. The anger classifier in this system obtained a F1-score of 0.64958 [87].

3.3.2 Irony and Sarcasm

Sarcasm, also referred as verbal irony [30]; [29] and [28], is defined as cutting remark to express contempt or ridicule according the free dictionary [70]. Based on the literature, the boundaries of irony and sarcasm are fuzzy [10], as some authors consider irony as global term that includes sarcasm [27], [85] and [50] and others analyze the differences between both [21].

Authors in [42] conduct a survey of the approaches used to detect sarcasm. Mainly three approaches have been proposed: rule-based, statistical approaches and Deep Learning approaches.

In 2009 Paula Carvalho et al. analyzed the detection of Portuguese irony by using pattern matching rules, such as diminutive forms, interjections, verb morphology, heavy punctuation, quotation marks, laughter expressions, among others. The proposed system classified the given sentences as ironic, not ironic, undecided and ambiguous. They concluded that the best patters for irony recognition are quote and laugh patterns, obtaining an accuracy of 68.29% and 85.40% respectively [14].

[54] developed in 2014 a rule based system to classify sentences in which the presence of sarcasm was known by analyzing hashtags in Twitter. Based on the premise that if the hashtag does not agree with the rest of the tweet the sentence is classified as sarcastic and by re-tokenizing hashtashs to split concatenated tokens, the classifier obtained a F1-score of 91.03 (result multiplied by 100).

Two important aspects must be pointed out regarding statistical approaches: the features used and the selected learning algorithms. Aditya Joshi et al.'s survey states that even though *"multiple variety of classifiers have been experimented for sarcasm detection most of the work conducted relies on SVM"* [42].

Authors in [19] proposed a system that classified Twitter and Amazon documents by employing semi-supervised techniques. The dataset was composed of 5.9 million unique tweets that contained the #sarcasm hashtag, which were pre-processed by removing any link appearance, users mentions and hashtags and replaced them by proper identification tags. The system relied on features such as High Frequency Words (HFWs), Content Words (CWs), punctuation based features to create single entry feature vectors. The classification output was defined as a number from range 1 to 5 to measure the presence of sarcasm in the given text and performed with a K-Nearest Neighbors (KNN)-like strategy [18]. To evaluate the system 5-fold cross validation was employed. The experiment that best performed used all the listed features obtained and obtained a F1-score of 0.545 on Twitter messages and 0.827 in Amazon reviews.

In 2012, Reyes et al. introduced the following features for irony detection: n-grams, Part of Speech (POS) n-grams, funny profiling, positive/negative profiling, affective profiling and pleasantness profiling. NB, SVM and Decision Tree (DT) classifiers were evaluated by comaparing positive sets against three negative subsets, begin SVM the learning algorithm that scored the highest in two out of the three subsets with a F1-score of 0.747 and 0.891 respectively.[67]. In 2013, they explored the use of skip-gram and character n-gram-based features for detecting irony representativeness and relevance with the Toyota case study. Dividing the tweets into three levels of representativeness the proposed model obtained a 0.66 F1-score on the third level [68].

Regarding DL, is a technology that is gaining popularity among researchers studying sarcasm and irony. Silvio Amir et al. in 2016 proposed a convolutional network-based system that learned content and context from user embeddings, achieving a 2% improve in absolute accuracy compared to the Bamman and Smith's 2015 baseline system [1].

Authors in [26] proposed a system that combined Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM) and Deep Neural Networks (DNN). They compare the approached system against recursive SVM and other datasets stating that their architecture obtains better F1-score than previous DL solutions, obtaining 0.901.

3.4. CLASSIFICATION TECHNIQUES

The aim of this section has two purposes. The first one, is to make an introduction of the basic concepts of classification, which is essential for the detection of repressed anger as the solution proposed has been defined as a classification problem. The second one, is to explain how the algorithms used in this study work.

3.4.1 Fundamentals of Classification

According to [81], classification can be defined as the task of predicting an outcome from a given input. This outcome is produced by the process of mapping a group of characteristics present in the input to a certain category. In other words, it consists in assigning objects (the input) to one of several predefined classes (the outcome) [62]. Examples of classification can be found in everyday life, such as e-mail spam detection, news classifiers, Optical Character Recognition (OCR), animal kingdom classification (see Figure 3.6), among many others.

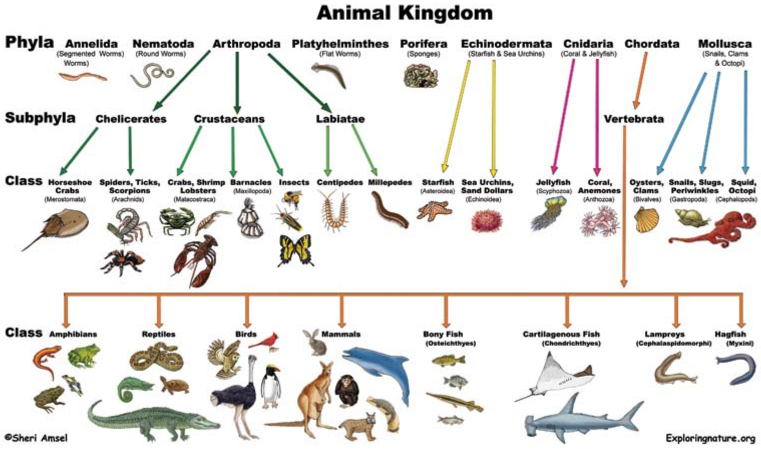

Animal Kingdom
Phyla
Annelida (Segmented Worms)
Nematoda (Round Worms)
Arthropoda
Platyhelminthes (Flat Worms)
Porifera (Sponges)
Echinodermata (Starfish & Sea Urchins)
Cnidaria (Coral & Jellyfish)
Chordata
Mollusca (Snails, Clams & Octopi)
Subphyla
Chelicerates
Crustaceans
Labiatae
Vertebrata
Class
Horseshoe Crabs (Merostomata)
Spiders, Ticks, Scorpions (Arachnids)
Crabs, Shrimp Lobsters (Malacostraca)
Barnacles (Maxillopoda)
Insects
Centipedes
Millepedes
Starfish (Asteroidea)
Sea Urchins, Sand Dollars (Echinoidea)
Jellyfish (Scyphozoa)
Coral, Anemones (Anthozoa)
Oysters, Clams (Bivalves)
Snails, Slugs, Periwinkles (Gastropoda)
Squid, Octopi (Cephalopods)
Class
Amphibians
Reptiles
Birds
Mammals
Bony Fish (Osteichthyes)
Cartilagenous Fish (Chondrichthyes)
Lampreys (Cephalaspidomorphi)
Hagfish (Myxini)
©Sheri Amsel
Exploringnature.org

Figure 3.6: Classification of animals. The image is extracted from Exploring Nature.

The input data for a classification task is composed by a collection of records, the dataset. In the same time, each record, also known as an instance, is composed by a set attributes. From all these attributes there is one considered special, which is called the target attribute or the class label. Regular attributes can be both discrete or continuous values. For the values signed for the class label, however, they must be discrete. This characteristic is what distinguishes classification form regression. Table 3.1 shows a sample dataset for animal classification into the following categories: amphibian, bird, fish or mammal.

Common Name	Hair	Feathers	Eggs	Milk	Aquatic	Legs	Class Label
antelope	Yes	No	No	No	No	4	mammal
catfish	No	No	Yes	No	Yes	0	fish
dolphin	No	No	No	Yes	Yes	0	mammal
dove	No	Yes	Yes	No	No	2	bird
duck	No	Yes	Yes	No	Yes	2	bird
elephant	Yes	Yes	No	Yes	No	4	mammal
flamingo	Yes	Yes	Yes	No	No	2	bird
frog	No	No	Yes	No	Yes	4	amphibian
fruit bat	Yes	No	No	Yes	No	2	mammal
gull	No	Yes	Yes	No	Yes	2	bird
herring	No	No	Yes	No	Yes	0	fish
kiwi	No	No	Yes	No	No	2	bird
lark	No	Yes	Yes	No	No	2	bird
lynx	Yes	No	No	Yes	No	4	mammal
mole	Yes	No	No	Yes	No	4	mammal
mongoose	Yes	No	No	Yes	No	4	mammal
newt	No	No	Yes	No	Yes	4	amphibian

Table 3.1: Animal kingdom dataset.

Tan Pang-Ning et al. propose a more mathematical definition of classification stating that it is the process of learning a target function f, also known as classification model, that maps each attribute set x to one of the predefined class labels y.

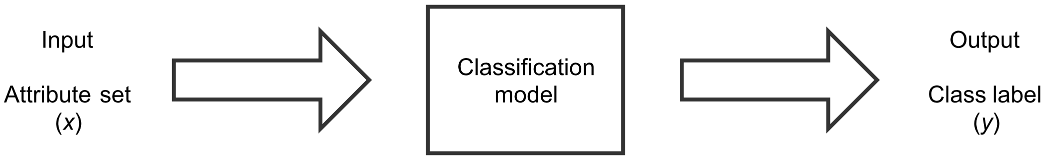


Figure 3.7: Classification as a task of mapping a set attributes x into its fitting class label y.

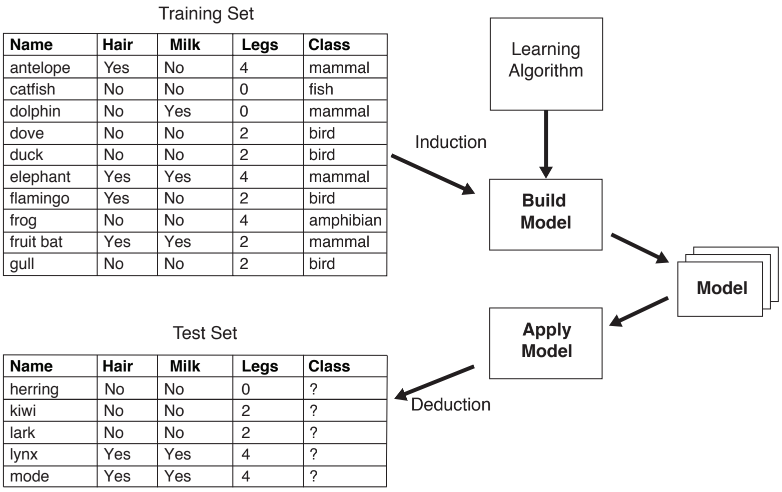
A classification model is useful for the following purposes [62]:

- **Descriptive Modeling:** Since a classification model presents the main features of the data, it can serve as an explanatory tool to distinguish between instances of different categories [53].
- **Predictive Modeling:** A classification model can also be used to predict the class label of an unknown new instance. As shown in Figure 3.7, a classification model can be represented as a black box that automatically assigns a class label to an instance by providing its attribute set.

It is important to remark that classification techniques perform their best when used for predicting or describing datasets which its class label is binary or nominal, Since they no consider properties such ordinality or the implicit order among the categories, they become ineffective with ordinal class labels [22].

3.4.2 General classification problem solving

For general classification problems solving, popular techniques consists on a process that starts with building classification models from a sample dataset [86]. Each technique depends on a learning algorithm witch is in charge of generating the classification model. A good model should define the relationship between the input attribute set and its belonging category that suits the best. Therefore the model should be valid for both, the sample data used to generate the model and also for new unknown instances. Among popular classification techniques Support Vector Machine (SVM), Neural Networks (NNs), Naive Bayes or Decision Trees (DTs) can be found [25].



Training Set

Name	Hair	Milk	Legs	Class
antelope	Yes	No	4	mammal
catfish	No	No	0	fish
dolphin	No	Yes	0	mammal
dove	No	No	2	bird
duck	No	No	2	bird
elephant	Yes	Yes	4	mammal
flamingo	Yes	No	2	bird
frog	No	No	4	amphibian
fruit bat	Yes	Yes	2	mammal
gull	No	No	2	bird

Test Set

Name	Hair	Milk	Legs	Class
herring	No	No	0	?
kiwi	No	No	2	?
lark	No	No	2	?
lynx	Yes	Yes	4	?
mode	Yes	Yes	4	?

Figure 3.8: General approach for classification model building and new instance category prediction.

As shown in the Figure 3.8, to solve a classification problem a sample dataset must be provided as a training set. This sample is used to build the classification model according to the learning algorithm. After the model is built, it is applied to unlabeled dataset, also called the test set, to predict the categories of each instances of the records. To measure how good the model is, there is only need to count the number of instances have been correctly and incorrectly classified from the test set. Usually, to represent system's performance values, a confusion matrix is used [35].

		Predicted Class	
		$Class = Yes$	$Class = No$
Actual Class	$Class = Yes$	a	b
	$Class = No$	c	d

Table 3.2: Confusion matrix of a binary classification.

As a example, Table 3.2 represents the confusion matrix of a binary classification problem, in which the possible classes that can be assigned to a given instance are *Yes* or *No*. The values presented in this confusion matrix represents the counts of instances correctly and incorrectly classified as a means of model performance evaluation, being the diagonal of the table the instances classified correctly, the true positives (TP) and negatives (TN). The rest of the table corresponds to the number of instances that have been classified incorrectly, the false positives (FP) and false negatives (FN). The definition of a cell done by the value of the column and the row in which is positioned, read as the number instances of X classified as Y, where X and Y are the determined value of the row and column respectively. For instance, the cell a is interpreted as the counts of Yes that have been classified as *Yes*, the TP.

Although the confusion matrix gives all the relevant information to determine how well the system has performed, sometimes is convenient to provide this information into a single value that summarizes the content of all the table. To do so, multiple performance metrics have been defined and one of those is the accuracy defined as:

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

Accuracy represents the percentage of correctly classified instances. However, using only accuracy may not be enough, as for example when the dataset is imbalanced or it has a majority of elements for a determined class. Of instance, on a naive classifier that always determined an instance as the element with the maximum number of appearances in the dataset, the accuracy of classifier results in a high value although, the model does not have predictive capacity. This is considered the accuracy paradox [20] and thus, to avoid the scoring a misleading measurement, it is recommendable to combine its usage with other performance metrics such as precision and recall.

Translating the precision from information retrieval context it would answer to the question *how many classified items are relevant for the current query?* and thus, the precision is defined as:

$$Precision = \frac{TP}{TP + FP}$$

In the same context recall, in the other hand, answers to the question *how many relevant items are classified properly?* and such, defined as:

$$Recall = \frac{TP}{TP + FN}$$

Finally, a combination of both measures, precision and recall, as a weighted average is called F-Measure or F1-Score. It allows to have a single performance metric to evaluate the classifier. Although, multiple definitions of the metric that give more relevance to the recall over the precision or vice versa exist, the general definition can be specified by the following equation:

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

3.4.3 Deep Learning

TODO [12]

Introduction to Sentiment Analysis. Relevant International Workshop found: SemEval Datasets repositories: Twitter as corpus, Amazon products reviews Dataset labeling: CrowdFlower, Amazon Mechanical Turk. Word embedding: Word2vec, GloVe. Text classification features: N-grams based on Sentiment lexicon Learning Methods: Supervised, Unsupervised, Distant Deep Learning techniques: Convolutional NN, Recurrent NN, Deep Autoencoders Classifiers: Bag of Words, Naive Bayes, Logistic Regression, SVM, NN, Clustering Accuracy evaluation: Precision, Recall, F1-Score, Cross-validation Trends: Best scoring teams used Deep Learning techniques.

Next week work: Deep reading and prepare a Playground with small sample data.

Read fundamental theory about classifiers such as SVN, Bayesian, KNN, Decision Tree. Urge test what has been learned in order to set the basic knowledge. Download and test WEKAs basic functionalities for educational and research purpose. (Includes a No-Code-GUI, which makes it user friendly for initial testings) Use scikit-learn for following iterations of the classifier (More modern algorithms, better memory management, Python development environment...) Further reading about Emotion classification techniques and investigate about repressed anger in English texts. Multiclass angry classification?

Future work: Deep Learning

Depending on the generated dataset (size and labeled/unlabeled) the project could opt for a DL based classifier. Could score better results in the benchmarking. To gather large amount of data for each class is compulsory solve the overfitting problem. (Detection and prevention by using Cross-validation) If the dataset is not completely labeled weakly (distant) supervised learning could be applied. Theano and Keras have proved a DL frameworks to perform well.

4. PROPOSED SOLUTION

The aim of this chapter is to explain in detail the steps followed to conduct the experiment first introduced in section **??**.

As explained in section 3.3, the suppression of anger can derivate in the development on a more extremist and sarcastic personality. To recognize repressed anger from text messages, this research has based on the hypothesis that repressed anger messages contain both anger and ironic features and thus, focus on the detection of these features independently on Twitter documents.

Many research have been done to detect theses to aspects, in general, we could say that the most explored approach used for emotion classification and irony detection are based on SVM learning algorithm. However, recently introduced technology has been gaining popularity among researchers: DL. In fact, during the task four presented in the international workshop of evaluations of computational semantics, SemEval 2016, the organizers stated as follow:

"A general trend that emerges from SemEval-2016 Task 4 is that most teams who were ranked at the top in the various subtasks used Deep Learning, including convolutional NN, recurrent NN, and (general purpose or task-specific) word embeddings. In many cases, the use of these techniques allowed the teams using them to obtain good scores even without tuning their system to the specifics of the subtask at hand [...]. Conversely, several teams that have indeed tunned their system to the specifics of the subtask at hand, but have not used Deep Learning techniques, have performed less satisfactorily. This is a further confirmation of the power of Deep Learning techniques for tweet sentiment analysis" [60].

DL has caused a great impact on SemEval and recent researches and therefore, based on these statements the decision to implement a system that make used of this technique was made.

Since DL is a class of ML, the procedure developed in the proposed solution (see figure 4.1) follows the steps of a classical Machine Learning approach (section 3.1.2): data collection, pre-processing, learning process, prediction and evaluation. In the following sections a general perspective of the proposed solution will introduced followed up by an explanation of the four steps in detail, while evaluation results will be presented of the following chapter.

4.1. DATA COLLECTION

The objective of subtask is to gather already annotated tweets that contains emotions, specially anger, and irony. During this step, multiple research paper that would provide public available dataset or would provide the methodology on how to build one were analyzed, such as the explained in [37] and [76]. In the following paragraphs describe the public datasets gathered and created for this project.

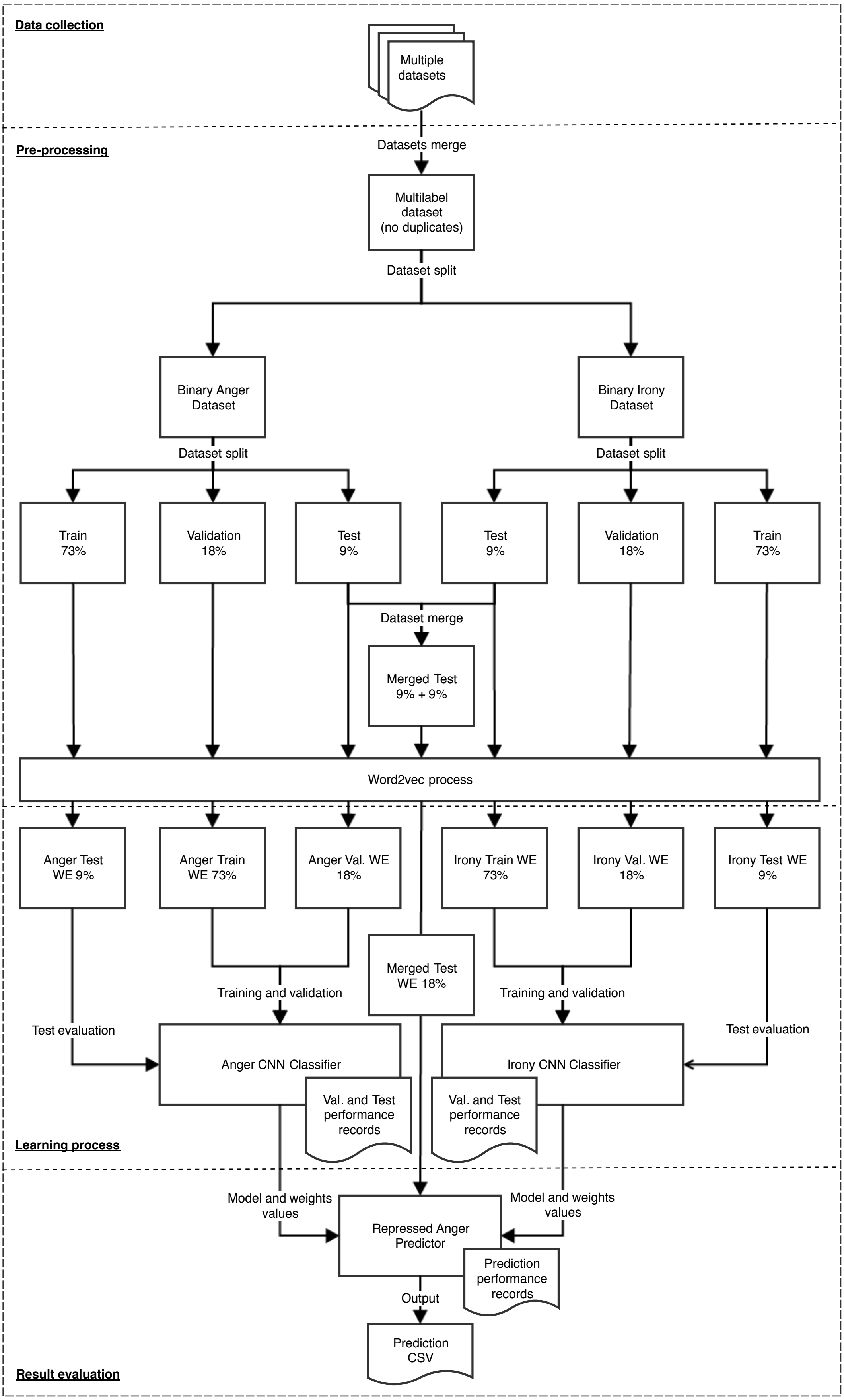


Figure 4.1: Proposed solution procedure.

4.1.1 Wang's SMILE Twitter emotion dataset

At first, only Wang et al.'s SMILE Twitter emotion dataset was found, containing a total of 3,085 tweets classified as: anger, disgust, happy, no-code, not-relevant, sad and surprise [82]. By simply taking into account the number of tweets offered in this dataset as small, and probably not enough, to perform DL based classification. Figure 4.2 shows the distribution of tweets related to each annotated class. By analyzing the number of tweets related to anger the amount of tweets are reduced to 57. Moreover, the tweets labeled as not-relevant and no-code, classes that contains no value to the purpose of the research, represents around 58% of the whole dataset. However, even if not being enough, the remaining 42% would be stored to posteriorly merged to other datasets.

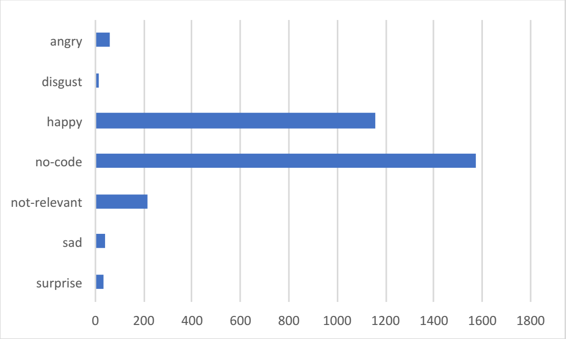


Figure 4.2: Wang's SMILE Twitter emotion dataset distribution.

4.1.2 CrowdFlower's Data for Everyone, Sentiment Analysis: Emotion in Text dataset

CrowdFlower, an HIT web solution, offers publicly available datasets under its "Data for Everyone" section, where an annotated emotion dataset for sentiment analysis can be downloaded for research purposes. The dataset is composed from tweets classified as the following 13 classes: anger, boredom, empty, enthusiasm, fun, happiness, hate, love, neutral, relief, sadness, surprise and worry [16]. Figure 4.3 shows the number of tweets per annotated class, being anger and hate the most relevant for the purpose of this project representing around 3% of the dataset.

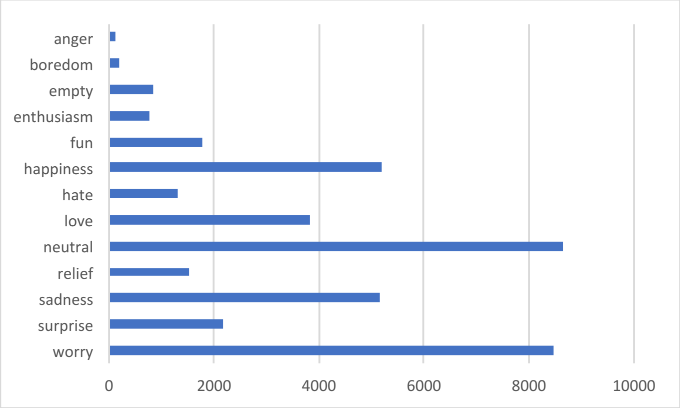


Figure 4.3: CrowdFlower's Data for Everyone, Sentiment Analysis: Emotion in Text dataset distribution.

4.1.3 Wang 2012, Harnessing Twitter "big data" dataset

In 2012 Wang et al. created a publicly available dataset from Twitter messages to analyze emotions. The dataset is composed from tweets classified in 7 categories as anger, fear, joy, love, sadness, surprise and thankfulness [83]. However, due to the term of service of Twitter, *"If you provide Content to third parties, including downloadable datasets of Content or an API that returns Content, you will only distribute or allow download of Tweet IDs and/or User IDs"* [80] and thus, this dataset does not contain the proper tweet message, but a identifier number instead. To download the content of the tweet an Twitter Python API known as Tweepy [9] was used over the dataset's "test" and "train_2_1" files. Although, since the originally the tweets were collected on 2011 to build the dataset, nowadays not all the tweets are available, as some users have removed their account or change their privacy settings. Figure 4.4 shows the number of tweets that were able to collect per class.

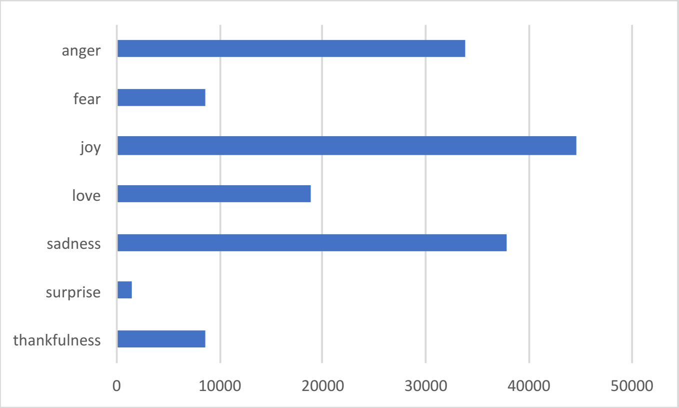


Figure 4.4: Wang 2012's test and train_2_1 datasets merged distribution.

4.1.4 Harvesting Irony dataset

To collect tweets related to irony, as no publicly available dataset was found on a reasonable time, it was decided to harvest automatically annotated tweets directly. To do so, Twitter archiver [79], an extension of Google Sheets, was used. It enables to automatically download tweets based on features such as keywords, hashtags, language, among others. By using this tool a total of 36,274 English tweets that contained hashtags such as #irony, #ironic, #sarcasm or #sarcastic where downloaded from December 12th 2016 to January 25th 2017.

4.2. PRE-PROCESSING

After having collected enough data to prevent DL overfitting issue, the next step is to pre-process the datasets, to eliminate non English tweets, possible duplicates, hashtags, user references, URLs, correct spell mistakes, filter undesired classes, among others.

The first step of pre-processing was to delete all duplicates. To do so, all the data was merged into a single dataset file from which all the tweets that contain the same identification number (real duplicate instances) or tweet sharing same text message (possible re-tweets) were deleted. After

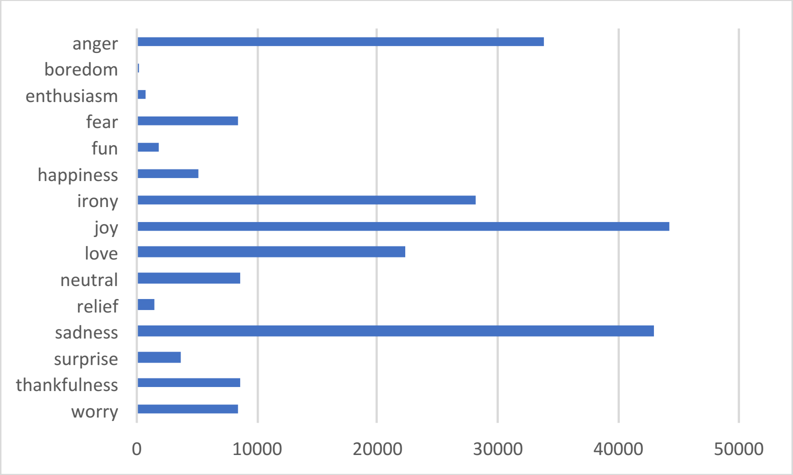
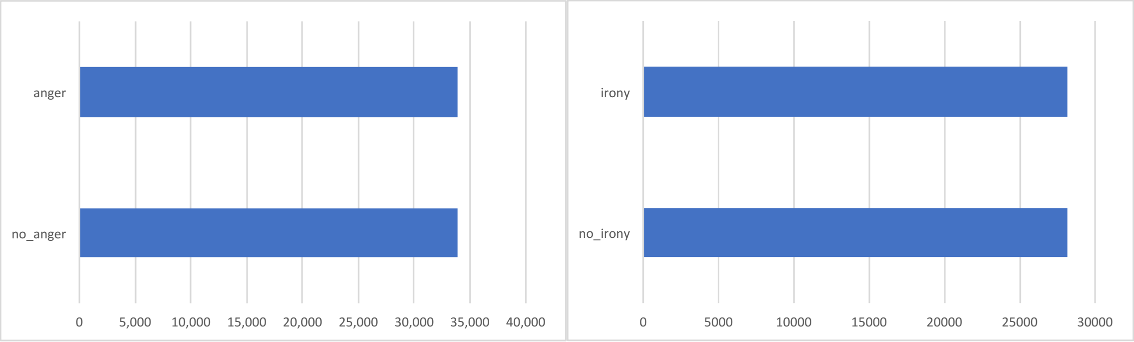


Figure 4.5: Merged multi label dataset distribution.

delete non English tweets and filter ambiguous target labels from the file, the final dataset was composed by a total of 218,619 tweets grouped into 15 categories classified as: anger, boredom, fear, fun, happiness, irony, joy, love, neutral, relief, sadness, surprise, thankfulness and worry. Figure 4.5 shows the distribution of each label target.

Because anger and irony are analyzed independently in this project, to isolate each classifier two datasets were built: binary anger dataset and binary irony dataset. As their name suggest, each dataset is composed by tweets grouped into two classes, the pairs "anger", "no_anger" and "irony", "no_irony" label targets. To prevent the learning process to be biased into one class or the other, the data was under-sampled to match the number of instances appearances of the anger and irony tweets in the previously merged multi-label dataset to created balanced datasets. Thus, binary anger dataset is composed by a total of 67,748 tweets, 33,874 for "anger" annotated tweets and the same amount for "no_anger" related tweets (see Figure 4.6a). Regarding to binary irony dataset, it is formed by 28,132 tweets for irony and the same amount for "no_irony", making a total 56,264 tweets (see Figure 4.6b).



(a) Balanced binary anger dataset

(b) Balanced binary irony dataset

Figure 4.6: Dataset under-sampling

To prepare the datasets for the posterior learning process, each dataset is divided into train, validation and test subsamples. Normally the division would be split into 70% training, 15% validation and 15% test, however, as in the last prediction phase the test sample is produced by merging both anger and irony test files, these percentages were changed to 73%, 18% and 9% respectively, to make the merged test file size have nearly the same size as each dataset validation files. Once both dataset have been split and the final prediction file generated by merging both datasets' test files, all the resulting files enter the word2vec process to generate the word embeddings files that will later on serve as input for the CNN learning process.

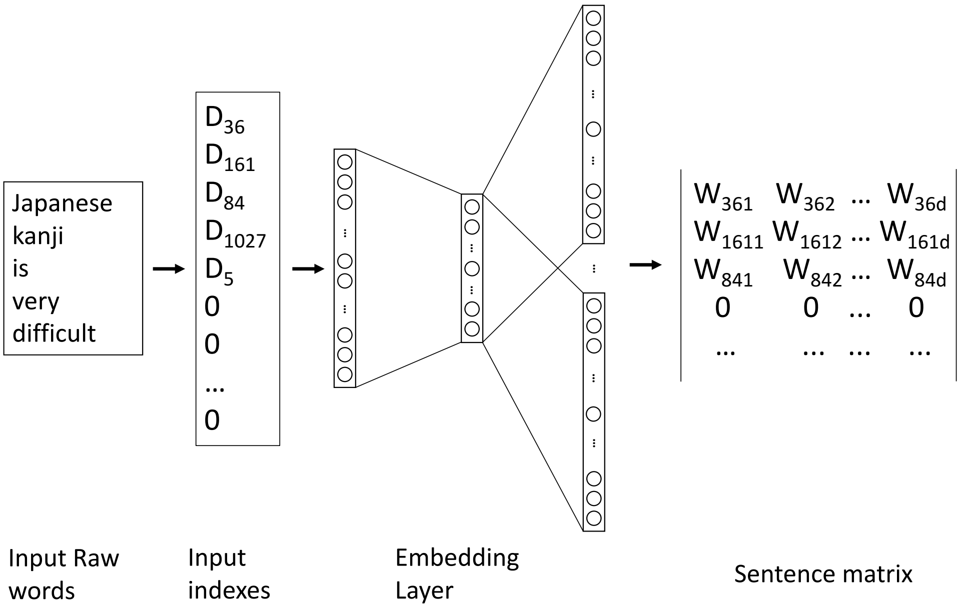


Figure 4.7: Raw word sentence to word vector matrix transformation.

During this process the pre-processing of the tweets that compose the corpus occurs and all the hashtags, user references, URLs are replaced by identification tags such as TAG, MENTION and URL, stopwords are removed from the sentences and optionally all the remaining words are checked and replaced by using Peter Norvig's spell checker [39] with an own made language model generated from calculating the word appearance of 36,173 free English HTML eBooks extracted by Kiwix harvester (2014) [49] Project Gutenberg [23], most used frequency words list obtained from TV programs and contemporary fiction books from Wiktionary [84], film series from Opensubtitles extracted by Hermit Dave [17] and British National Corpus (BNC) [46]. Since the learning process requires to introduce a sequence of words of a fixed length, we decided the value based on the length of the phrase length of our corpus, and thus, calculated the maximum sentence length of both anger (29) and irony (28) datasets and select the maximum value (29) as the fixed sequence length for the model generation. All tweets sentences composing the corpus are transformed sequences of 29 indexes by using a dictionary D that contains the mapping to indexes of millions of words. Then, by ploying a embedding layer, each of those indexes are converted into space vectors, which are initialized by using the Word2Vec algorithm [56]. The combination all the space vectors that compose the original tweet sentence creates an output matrix $M \in \mathbb{R}^{29 \times d}$, where d represents the length of the word embeddings (see figure 4.7). As not all the input sentences contain 29 words, each matrix is padded with 0s until reaching the fixed sentence length. Finally, the list of generated matrices are stored for their later use on the learning and prediction processes.

4.3. LEARNING PROCESS

The learning solution proposed consist on a hybrid approach that combines the prediction output of two deep neural network classifiers. To design the architecture of each classifier, we followed the work of Kim [47], which uses CNN for sentence classification. The input of this architecture is an sequence of words, more precisely, the list of word embedding matrices generated on the previous word2vec process. Our model is fed with each matrix and performs convolution operations with multiple filters to learn from multiple features as stated in [88]. We have experimented using 200 filters of 3 sizes (3, 4 and 5) with a ReLU activation function. The functionality of these filters can be compared with how n-grams based classification work [15]. Our model relies on what on n-grams would be considered as 3-grams, 4-grams and 5-grams. The n-grams are extracted from input matrix, in which each row represents a word from the tweet. Each filter combines a groups of 3, 4 or 5 rows obtaining all the possible n-grams from the sentence, generating as a result, a feature map. This map serve as the input of the next max-pooling layer. Its objective is to reduce the profile and bandwidth of the input matrix for computational purposes. Although multiple pooling strategies exist [11], the experiments performed in [88] shows that max-pooling strategy outperforms the rest and thus, selected this strategy to develop our model. It extracts only the feature with the highest value from the initial feature map. When all the features maps from each filter have been processed, the extracted features are then concatenated into a single feature vector to which a dropout layer of rate 0.5 is applied as a measure to prevent the model from over-fitting [72]. The model ends with a final softmax layer that calculates the probability distribution over the output vector, representing the classification into the target labels. For training purpose, a categorical cross-entropy loss function has been used with Adam optimizer [48]. The figure 4.8 illustrates designed architecture using 2 filters instead of 200 for explanatory purposes.

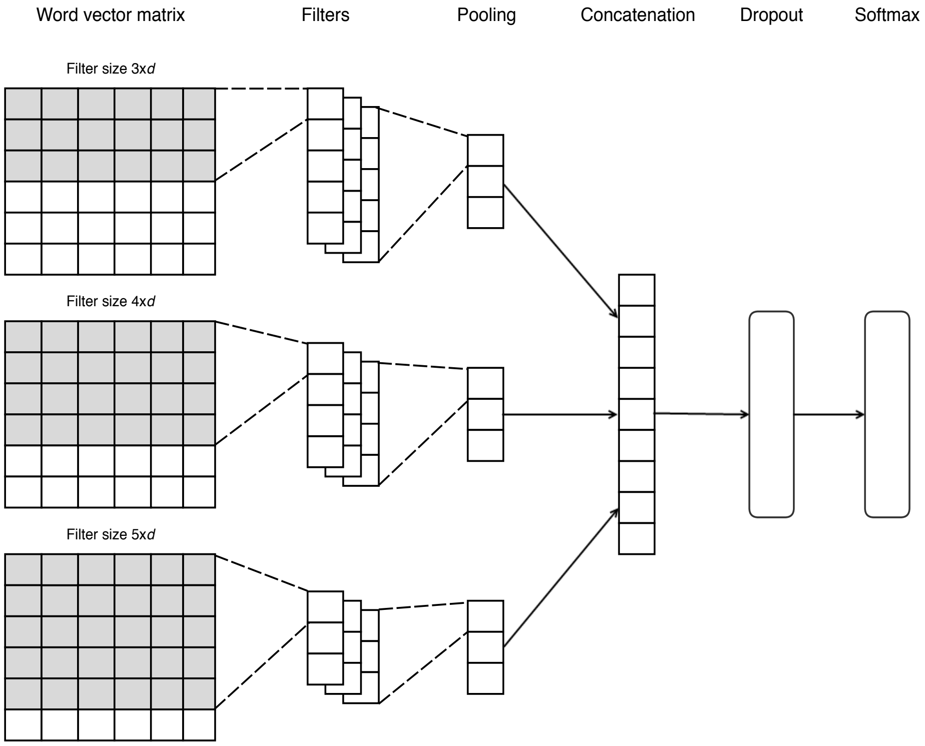
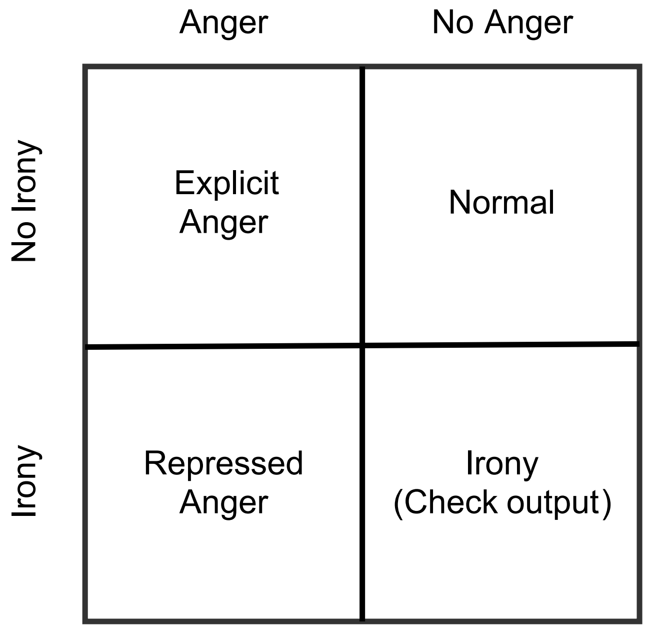


Figure 4.8: Designed architecture.

4.4. PREDICTION

Once both classifiers have been trained and validated with the corresponding word embeddings matrices files two files are generated per classifier: the model, which saves the architecture of the classifier, and the weights in which the classifier has obtained best result during the learning process. Having as a input file the combination of each classifier test sample, theses four files are used to load the anger and irony classifiers' parameters to perform the prediction of new data. Each classifier generates a binary class output prediction of a given tweet that need to be combined. Although the usage Ensemble Learning based on divide and conquer [63] approach was studied to merge classifiers, as each classifier works with its own set target labels independently this idea was discarded. Thus, the solution proposed, following the definition of repressed anger personality first presented in section 3.3, consists on merging the output of the classifiers generating four target labels according to the matrix presented in figure 4.9.



	Anger	No Anger
No Irony	Explicit Anger	Normal
Irony	Repressed Anger	Irony (Check output)

Figure 4.9: Classification prediction output's merge matrix.

5. RESULTS

The aim of this chapter is to explain the procedure followed to evaluate the system throughout the different experiments conducted, concluding with the presentation of the results of each test.

5.1. MANUAL ANNOTATION

Since the target labels that repressed anger predictor outputs does not match with the classes of either anger and irony binary datasets, a manual annotation of the merged test data must be conducted in order to be able to evaluate the performance of the proposed model. From the 11,074 tweets dedicated for the prediction phase, 1,000 were selected randomly to be part of the manual annotation phase. As the reliability of a single human classification is proven to be low [32] a HIT was conducted to prevent the annotation to be biased.

The HIT was conducted by means of online surveys that were automatically generated as forms by using Google Apps Script. To get a better sense of how difficult the task of repressed anger identification is, the contents of the tweets were presented without anger and irony related hashtags. A total of five non English native collaborators performed the annotation task. The comments received by judges reflect the uncertainty to determine which category a tweet belonged to, as sometimes either context of the tweet was missing or the topic was unknown. We considered valid all the classification which, at least, more that half of the judged agreed. From the 1,000 annotated tweets 824 fulfilled this condition as the remaining 176 was considered ambiguous and thus, discarded from the dataset. The figure 5.1 illustrates the distribution of the manually classified dataset that was used to evaluated the performance of the system.

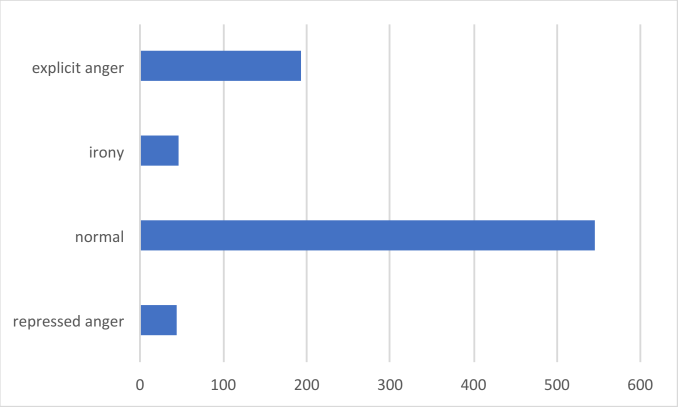


Figure 5.1: Manual annotated dataset distribution.

5.2. EXPERIMENTS

The project baseline parts from a naive classifier that determines every given instance as the the element with the maximum number of appearances in the dataset, in this case, the *normal* label. From this solution, the designed model is implemented and compared with the baseline to measure how better the proposed solution is. Finally, variations in the proposed procedure are made to try to improve the results.

Because the model designed depends on the word sequence matrices to generate the features maps, from which the model learns, and motivated by the fact that the written style contains features used on oral speech, usually containing spelling mistakes that affect on how the word sequence matrices are generated, the purpose of the modifications done in the procedure aim to improve the data pre-processing and thus, indirectly improve the performance of the model. The experiments conducted in this experiments are the following:

5.2.1 Naive classifier

As explained before, this is the baseline the project bases on to compare the improvement achieved against the proposed solution. It consist of a dummy classifier that assign to each given tweet the *normal* target label.

The results provided make used of the confusion matrix and performance metrics first introduce in the section 3.4.2, with all the values rounded to two decimal places.

	irony	normal	repressed anger	explicit anger
irony	0.0	1.0	0.0	0.0
normal	0.0	1.0	0.0	0.0
repressed anger	0.0	1.0	0.0	0.0
explicit anger	0.0	1.0	0.0	0.0

Table 5.1: Naive classifier: normalized confusion matrix

Class	Average	Precision	Recall	F1-Score
Irony	0.0	0.0	0.0	0.0
Normal	1.0	0.66	1.0	0.8
Repressed Anger	0.0	0.0	0.0	0.0
Explicit anger	0.0	0.0	0.0	0.0
Overall	0.66	0.16	0.25	0.2

Table 5.2: Naive classifier: Performance

5.2.2 Google News word2vec pre-trained model

This is original proposed solution, first introduced in chapter 4. It uses the publicly available Google News pre-trained model [33] to generate the word sequence matrices in the pre-processing phase. As with the rest of procedure variations, the performance metrics are divided into two subsection: (i) on the one hand, the evaluation of how the binary classifiers managed to determine independently if a tweet contains anger or irony; (ii) on the other hand, the evaluation of the repressed anger predictor that depends on how well the previous binary classifiers perform to merge their output.

Binary classifiers

	anger	no_anger
anger	0.77	0.23
no_anger	0.2	0.8

Table 5.3: Binary anger classifier (Google News): normalized confusion matrix

Class	Average	Precision	Recall	F1-Score
anger	0.77	0.79	0.77	0.78
no_anger	0.8	0.78	0.8	0.79
Overall	0.78	0.78	0.78	0.78

Table 5.4: Binary anger classifier (Google News): Performance

	irony	no_irony
irony	0.83	0.17
no_irony	0.14	0.86

Table 5.5: Binary irony classifier (Google News): normalized confusion matrix

Class	Average	Precision	Recall	F1-Score
irony	0.83	0.86	0.83	0.84
no_irony	0.86	0.83	0.86	0.84
Overall	0.85	0.85	0.85	0.85

Table 5.6: Binary irony classifier (Google News): Performance

Repressed anger predictor

	irony	normal	repressed anger	explicit anger
irony	0.47	0.3	0.13	0.11
normal	0.14	0.56	0.1	0.2
repressed anger	0.23	0.2	0.43	0.14
explicit anger	0.09	0.24	0.25	0.42

Table 5.7: Google News: normalized confusion matrix

Class	Average	Precision	Recall	F1-Score
Irony	0.47	0.17	0.47	0.24
Normal	0.56	0.82	0.56	0.67
Repressed Anger	0.43	0.15	0.43	0.22
Explicit anger	0.42	0.4	0.42	0.41
Overall	0.51	0.38	0.47	0.39

Table 5.8: Google News: Performance

5.2.3 Fréderic Goding word2vec pre-trained model

This experiments consist on the same idea as original solution. However, to generate the word sequence matrices, instead of using the Google-News-Vector(300) pre-trained model, which has been created by using the words from the Google News articles, the Fréderic Goding 400 feature pre-trained model [31] that was generated directly from tweets was used. As tweets are composed by words and a written style that differs from news, using a pre-trained model developed from the content generated from Twitter users could increase the chances of tweet's word appearance in the pre-trained model dictionary and thus, be able to convert more words into a vector space, instead of initialize the unknown words with Zero padding vectors.

Binary classifiers

	anger	no_anger
anger	0.83	0.17
no_anger	0.24	0.76

Table 5.9: Binary anger classifier (Fréderic Goding): normalized confusion matrix

Class	Average	Precision	Recall	F1-Score
anger	0.83	0.77	0.82	0.79
no_anger	0.76	0.82	0.76	0.79
Overall	0.8	0.8	0.8	0.8

Table 5.10: Binary anger classifier (Fréderic Goding): Performance

	irony	no_irony
irony	0.86	0.14
no_irony	0.14	0.86

Table 5.11: Binary irony classifier (Fréderic Goding): normalized confusion matrix

Class	Average	Precision	Recall	F1-Score
irony	0.86	0.87	0.86	0.86
no_irony	0.86	0.86	0.86	0.86
Overall	0.86	0.86	0.86	0.86

Table 5.12: Binary irony classifier (Fréderic Goding): Performance

Repressed anger predictor

	irony	normal	repressed anger	explicit anger
irony	0.45	0.3	0.21	0.04
normal	0.13	0.65	0.07	0.14
repressed anger	0.34	0.23	0.25	0.18
explicit anger	0.1	0.28	0.2	0.41

Table 5.13: Fréderic Goding: normalized confusion matrix

Class	Average	Precision	Recall	F1-Score
Irony	0.45	0.16	0.45	0.24
Normal	0.65	0.82	0.65	0.73
Repressed Anger	0.25	0.11	0.25	0.15
Explicit anger	0.41	0.47	0.41	0.44
Overall	0.56	0.39	0.44	0.39

Table 5.14: Fréderic Goding: Performance

5.2.4 Google News word2vec pre-trained model plus spell checker

This variation consist on adding a spell checker to those words that the Google-New-Vector(300) does not know. The usage of a spell checker would increase the chances of recognizing more words in the pre-trained model's dictionary and thus, initialize its space vector with Word2vec algorithm, helping to obtain new features that would enable the model to achieve better results. As explained in section 4.2 of the proposed solution, the spell checker used is based on Peter Norvig's solution. In his blog, he published the code and a language model, big.txt, used to select possible word candidates that correct the grammatical mistakes of the given word. Table 5.15 compares the performance obtained by both, big.txt and the self-made language model by executing the evaluation tasks that Norvig designed that checks if the proposed correct word is the expected.

Model	No. of words to correct	Average of correct words	Average of unknown words	Words processed per second
big.txt language model	270	51%	35%	37
proposed language model	270	**75%**	**5%**	**64**
big.txt language model	400	53%	32%	35
proposed language model	400	**69%**	**7%**	**63**

Table 5.15: Language model evaluation results

Binary classifiers

	anger	no_anger
anger	0.76	0.24
no_anger	0.19	0.81

Table 5.16: Binary anger classifier (Google News + spell checker): normalized confusion matrix

Class	Average	Precision	Recall	F1-Score
anger	0.76	0.79	0.75	0.79
no_anger	0.81	0.77	0.81	0.79
Overall	0.78	0.78	0.78	0.78

Table 5.17: Binary anger classifier (Google News + spell checker): Performance

	irony	no_irony
irony	0.83	0.17
no_irony	0.12	0.88

Table 5.18: Binary irony classifier (Google News + spell checker): normalized confusion matrix

Class	Average	Precision	Recall	F1-Score
irony	0.83	0.87	0.82	0.84
no_irony	0.88	0.83	0.88	0.85
Overall	0.85	0.85	0.85	0.85

Table 5.19: Binary irony classifier (Google News + spell checker): Performance

Repressed anger predictor

	irony	normal	repressed anger	explicit anger
irony	0.47	0.3	0.13	0.11
normal	0.16	0.59	0.08	0.17
repressed anger	0.3	0.18	0.43	0.09
explicit anger	0.08	0.25	0.27	0.4

Table 5.20: Google News + spell checker: normalized confusion matrix

Class	Average	Precision	Recall	F1-Score
Irony	0.47	0.16	0.47	0.24
Normal	0.59	0.82	0.59	0.69
Repressed Anger	0.43	0.16	0.43	0.23
Explicit anger	0.4	0.43	0.4	0.42
Overall	0.53	0.39	0.47	0.39

Table 5.21: Google News + spell checker: Performance

5.3. RESULT SUMMARY

This section aims to summarize the relevant information from all the scores presented throughout the chapter. Table 5.22 illustrates how each model has performed on detecting repressed anger target label, while table 5.23 compares the overall performance of each model.

Model	Average	Precision	Recall	F1-Score
Naive classifier	0.0	0.0	0.0	0.0
Google News	**0.43**	0.15	**0.43**	0.22
Fréderic Goding	0.25	0.11	0.25	0.15
Google News + spell	**0.43**	**0.16**	**0.43**	**0.23**

Table 5.22: Repressed anger performance comparison

Model	Average	Precision	Recall	F1-Score
Naive classifier	**0.66**	0.16	0.25	0.2
Google News	0.51	0.38	**0.47**	**0.39**
Fréderic Goding	0.56	**0.39**	0.44	**0.39**
Google News + spell	0.53	**0.39**	**0.47**	**0.39**

Table 5.23: Overall model performance comparison

6. CONCLUSION AND FUTURE WORK

In conclusion, in thesis we have done introduced the topic of automatic detection of repressed anger from text messages. Firstly, we described the research methodology and the planning that has been carried out its completion. Later, we have made a review of the research topics which this investigation is based on: (a) sentiment analysis and (b) emotion detection, followed by an introduction of relevant papers and previous work done, concluding the state of the art with a formal explanation about the fundamentals of the techniques that were going to be used to develop our approach.

Thus, with this base, we explained the design of our system, reviewing work done throughout all the steps that compose the procedure developed. On the next chapter, we enumerated all the experiments done to try to improve the performance of our model and compared designed approach with a naive classifier. In the results, we could see an improvement in the performance by evaluation the naive classifier and the variations of our solution with a manual imbalanced dataset based on real-life sentences extracted from users of Twitter social media. From zero chances of detecting repressed anger our first approach obtained a 0.22 F1-Score, proving the hypothesis *"Can the repressed anger be detected in text?"* to be true even though, the results obtained with this approach are low, reflecting how complex the task is.

As the tweets usually contain slang, acronyms, spelling mistakes and expressions from oral language processing all the information can become a real challenge that can threat the performance of the model in the learning phase. With this idea in mind, the motivation of the variations performed to the proposed solution aim to improve the tweet pre-processing allowing to out model to learn from a wider feature set. Therefore, an alternative Word2vec pre-trained model that was build from millions of tweets was proposed as an alternative to the wide used Google News pre-trained model. Even though that this variation sightly improved the overall F1-Score performance (less than a thousandth), the detection of repressed anger decreased from a 0.22 to 0.15, showing that even a pre-trained model of 400 features that learned from tweets, cannot compete with the 300 feature pre-trained model created by Google from thousands of millions of words from news.

By analyzing the previous results, the following experiment focuses on making the Google pre-trained model recognize more words from which new features could be learned. From previous researches that used Word2vec model state that when word does not appear into the model the following actions can be performed: (a) stemming, (b) lemmatization, (c) synonyms, (d) morphological analyzers, (e) spelling correction, (f) slang and acronyms processing and (g) retrain the Word2vec model. Since the last option was not a viable, the securest approach to implement was the spelling correction, since if the word to be processed is wrong spelled nor lemmatization, synonyms or other options would work. Thus, before other options would work the correction of the spelling should be ensured and therefore, this approach was implemented. Applying the spell checker to the Google News pre-trained model increased the overall performance of the model sightly more than with the Twitter pre-trained model (less than a thousandth). However, with this implementation the performance on repressed anger detection did not decreased, in fact increased from 0.22 to 0.23 F1-Score, becoming in the solution that sightly outperformed the rest in precision, recall and F1-Score.

Without changing the neural design of the model, the tests conducted show us how important having a proper trained Word2vec model is, affecting in how the predictor classified the tweet instances into each categories. The usage of a simple spell checker and corrector has proven that more work into pre-processing phase would lead this model to score better results. There is still lots of work to be done in this aspect. Currently the spell checker only corrects words one by one isolatedly archiving, as shown in table 5.15, a 75% and 69% of reliability on the evaluation tasks. Norvig suggests that a *"spelling corrector that scores 90% accuracy will need to use the context of the surrounding words to make a choice"*. We this variation of the spell checking implementation would be viable, as Google has released a English word frequency list that contains word sequences up to 5-grams gathered from corpus from trillions of words [51]. Once the spell checker has obtained a relatively high average of words correction, it would be viable to implement the remaining presented approaches to improve the initialization of space models in the word embedding matrices.

Once the pre-processing of the corpus is fair enough, modifications of the neural design could also have a big impact of the model's performance. As for example, [41] and [3] show that by adding a ReLU activation function layer with batch normalization could both, considerably increase model's accuracy and reduce the learning process spent time. Another factor that should be work on is the model's hyperparmeter optimization. All the proposed variations have been executed with a fixed setting of 200 filters of 3-grams, 4-grams and 5-grams size. The modification of this values with multiple combinations could lead to positively improve the performance.

It is relevant to indicate that, although the binary anger and irony classified performed relative good, with a minimum F1-Score of 0.78 and 0.85 respectively, the repressed anger predictor did not achieve that level of reliability, obtaining a little more than 0.39 F1-Score in the best case. Taking advantage of the manual annotated dataset created to evaluate this model, an alternative approach to the proposed solution could be a similar but single neural network that learn directly from the manual annotated data instead from the automatically annotated binary anger and irony datasets. By using distant or semi-supervised learning techniques it should be possible to used the small amount of annotated data as base from which determine the relevant features that defines repressed anger and then, find similitudes with the remaining majority of unlabeled data from which new features could be exploited in the training phase. This approach would enable to learn the features altogether instead of separate them into anger and irony, which might lead to solve the deficiencies found on the merging process of the designed predictor.

Finally, a more in depth psychological definition on repressed anger would also promote the definition of to new subtasks in this domain, such as in profiling the personality of individual users by exploiting more complex attributes that characterize people suffering from repressing anger. The results of this subtask could be applied on domains in which the detection of this personality disorders can help analyze the well being or health of the subjects.

7. SELF-ASSESSMENT

Personally this project started as a challenge. Throughout my career I have always chosen tasks I have never experienced before as a way to test myself, the work done in this subject has not been an exception. This project has served me as an introduction to research and the scientific world, a field that has been unknown for me until now. In general, I have learned traverse though the references of relevant manuscripts of a research topic, which as enable me to find more interesting documents that develop previous ideas and help me to understand the concepts that with a single paper I would not be able to comprehend.

One of the highest difficulties that I have found when dealing with the investigation topic is to give a formal definition of what repressed anger is and how it manifests. To do so, I had to read thorough previous studies that ended up into psychological documents that explain theories that were totally unrelated with the technical knowledge I was used to work with.

The worst part by far was to deal with the uncertainty. Even though papers that proved how a architecture performed well with their solution, after spending hundreds of hours searching annotated datasets that could apply to the research topic or implementing a solution that gathers all the data to generate a corpus, design the solution based on combining the ideas extracted from the manuscripts, implementing all parts of the process and manage to evaluate it, there was no way to preview if all the effort would result into scoring good results.

The good this was that with this project, I had the opportunity to dive into artificial, specially into a branch I had almost no knowledge about: machine learning. Parting with a few vague test with Neural Networks and Naive Bayes I started get familiarized with some well-known classifiers while I was reviewing the state of the art, such as Random Forests, KNN, SVM, N-grams based classification, among others, as I studied their mathematical foundations and made empirical testings by using the tools provided by the Waikato Environment for Knowledge Analysis (WEKA). In the end, all theses experiments were discarded as possible baseline due to time constrains and focused on the most promising classifier: Convolutional Neural Networks, a technique based on the recently popular Deep Learning.

I am glad that I the end I able to base the proposed solution on this thesis on DL, since is a technology that I was very interested in. This project has enable me to have an introduction to DL and try to understand the magic that hides beneath and makes it so powerful. It is a precious knowledge that I would like to keep developing in the future.

Moving on to a more technical aspects of the development, this project [TALK ABOUT THE HARDWARE, GPU ACCELERATION, OS INSTALATION, ENVIRONMENT SETTING, DEBUGING AND TESTING, LONG EXECUTION TIMES, ETC. THAT HAVE MADE ME REMEMBER SOME CONTENT STUDIED DURING MY BACHELLOR.]

[CONCLUDE WITH THE DESIRE TO CONTINUE WITH RESEARH, AS I HAVE TO WORK

ON THE PDH PROPOSAL]

Bibliography

[1] Silvio Amir, Byron C Wallace, Hao Lyu and Paula Carvalho Mário J Silva. ‘Modelling context with user embeddings for sarcasm detection in social media’. In: *arXiv preprint arXiv:1607.00976* (2016).

[2] *Anger*. 2000. URL: `http://www.apa.org/topics/anger/` (visited on 17/09/2016).

[3] Jimmy Lei Ba, Jamie Ryan Kiros and Geoffrey E Hinton. ‘Layer normalization’. In: *arXiv preprint arXiv:1607.06450* (2016).

[4] Eugene Y Bann and Joanna J Bryson. ‘The conceptualisation of emotion qualia: Semantic clustering of emotional Tweets’. In: *Computational Models of Cognitive Processes: Proceedings of the 13th Neural Computation and Psychology Workshop, San Sebastian, Spain, 12-14 July 2012*. Volume 21. World Scientific. 2013, page 249.

[5] Tanja Bänziger, Didier Grandjean and Klaus R Scherer. ‘Emotion recognition from expressions in face, voice, and body: the Multimodal Emotion Recognition Test (MERT).’ In: *Emotion* 9.5 (2009), page 691.

[6] John N Bassili. ‘Emotion recognition: the role of facial movement and the relative importance of upper and lower areas of the face.’ In: *Journal of personality and social psychology* 37.11 (1979), page 2049.

[7] Haji Binali and Vidyasagar Potdar. ‘Emotion detection state of the art’. In: *Proceedings of the CUBE International Information Technology Conference*. ACM. 2012, pages 501–507.

[8] Haji Binali, Chen Wu and Vidyasagar Potdar. ‘Computational approaches for emotion detection in text’. In: *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*. IEEE. 2010, pages 172–177.

[9] bliti. *An easy-to-use Python library for accessing the Twitter API*. URL: `http://www.tweepy.org` (visited on 12/10/2016).

[10] Cristina Bosco, Viviana Patti and Andrea Bolioli. ‘Developing corpora for sentiment analysis: The case of irony and senti-tut’. In: *IEEE Intelligent Systems* 28.2 (2013), pages 55–63.

[11] Y-Lan Boureau, Jean Ponce and Yann LeCun. ‘A theoretical analysis of feature pooling in visual recognition’. In: *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010, pages 111–118.

[12] Denny Britz. *UNDERSTANDING CONVOLUTIONAL NEURAL NETWORKS FOR NLP*. 2015. URL: `http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/` (visited on 02/12/2016).

[13] Erik Cambria, Andrew Livingstone and Amir Hussain. ‘The hourglass of emotions’. In: *Cognitive behavioural systems*. Springer, 2012, pages 144–157.

[14] Paula Carvalho, Lus Sarmento, Mário J Silva and Eugénio De Oliveira. 'Clues for detecting irony in user-generated contents: oh...!! it's so easy;-'. In: *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*. ACM. 2009, pages 53–56.

[15] William B Cavnar, John M Trenkle et al. 'N-gram-based text categorization'. In: *Ann Arbor MI* 48113.2 (1994), pages 161–175.

[16] *CrowdFlower: Data for Everyone*. URL: `https://www.crowdflower.com/data-for-everyone/` (visited on 06/03/2016).

[17] Hermit Dave. *Invoke IT Limited: Frequency Word Lists*. URL: `https://invokeit.wordpress.com/frequency-word-lists/` (visited on 28/11/2016).

[18] Dmitry Davidov, Oren Tsur and Ari Rappoport. 'Enhanced sentiment learning using twitter hashtags and smileys'. In: *Proceedings of the 23rd international conference on computational linguistics: posters*. Association for Computational Linguistics. 2010, pages 241–249.

[19] Dmitry Davidov, Oren Tsur and Ari Rappoport. 'Semi-supervised recognition of sarcastic sentences in twitter and amazon'. In: *Proceedings of the fourteenth conference on computational natural language learning*. Association for Computational Linguistics. 2010, pages 107–116.

[20] Alan Descoins. *Why accuracy alone is a bad measure for classification tasks, and what we can do about it*. 2013. URL: `https://tryolabs.com/blog/2013/03/25/why-accuracy-alone-bad-measure-classification-tasks-and-what-we-can-do-about-it/` (visited on 28/09/2016).

[21] Elena Filatova. 'Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing.' In: *LREC*. Citeseer. 2012, pages 392–398.

[22] Eibe Frank and Mark Hall. 'A simple approach to ordinal classification'. In: *European Conference on Machine Learning*. Springer. 2001, pages 145–156.

[23] *Free ebooks by Project Gutenberg*. URL: `http://www.gutenberg.org` (visited on 01/12/2016).

[24] Johannes Fürnkranz. 'A study using n-gram features for text categorization'. In: *Austrian Research Institute for Artifical Intelligence* 3.1998 (1998), pages 1–10.

[25] GV Garje, Apoorva Inamdar, Apeksha Bhansali, Saif Ali Khan and Harsha Mahajan. 'SENTIMENT ANALYSIS: CLASSIFICATION AND SEARCHING TECHNIQUES'. In: (2016).

[26] Aniruddha Ghosh and Tony Veale. 'Fracking sarcasm using neural network'. In: *Proceedings of NAACL-HLT*. 2016, pages 161–169.

[27] Raymond W Gibbs and Jennifer O'Brien. 'Psychological aspects of irony understanding'. In: *Journal of pragmatics* 16.6 (1991), pages 523–530.

[28] Rachel Giora, Ari Drucker, Ofer Fein and Itamar Mendelson. 'Default sarcastic interpretations: On the priority of nonsalient interpretations'. In: *Discourse Processes* 52.3 (2015), pages 173–200.

[29] Rachel Giora, Shir Givoni and Ofer Fein. 'Defaultness reigns: the case of sarcasm'. In: *Metaphor and Symbol* 30.4 (2015), pages 290–313.

[30] Rachel Giora, Elad Livnat, Ofer Fein, Anat Barnea, Rakefet Zeiman and Iddo Berger. 'Negation generates nonliteral interpretations by default'. In: *Metaphor and Symbol* 28.2 (2013), pages 89–115.

[31] Fréderic Godin, Baptist Vandersmissen, Wesley De Neve and Rik Van de Walle. 'Multimedia lab acl w-nut ner shared task: named entity recognition for twitter microposts using distributed word representations'. In: *ACL-IJCNLP* 2015 (2015), pages 146–153.

[32] Roberto González-Ibánez, Smaranda Muresan and Nina Wacholder. 'Identifying sarcasm in Twitter: a closer look'. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2.* Association for Computational Linguistics. 2011, pages 581–586.

[33] *Google Code: word2vec.* 2013. URL: `https://code.google.com/archive/p/word2vec/#Pre-trained_word_and_phrase_vectors` (visited on 23/11/2016).

[34] Hatice Gunes and Massimo Piccardi. 'Bi-modal emotion recognition from expressive face and body gestures'. In: *Journal of Network and Computer Applications* 30.4 (2007), pages 1334–1345.

[35] Howard Hamilton. *Confusion Matrix.* 2000. URL: `http://www2.cs.uregina.ca/~dbd/cs831/notes/confusion_matrix/confusion_matrix.html` (visited on 21/10/2016).

[36] Steven Handel. *Classification of Emotions.* 2011. (Visited on 16/10/2016).

[37] Maryam Hasan, Elke Rundensteiner and Emmanuel Agu. 'Emotex: Detecting emotions in twitter messages'. In: *Academy of Science and Engineering (ASE)* (2014).

[38] Vasileios Hatzivassiloglou and Janyce M Wiebe. 'Effects of adjective orientation and gradability on sentence subjectivity'. In: *Proceedings of the 18th conference on Computational linguistics-Volume 1.* Association for Computational Linguistics. 2000, pages 299–305.

[39] *How to Write a Spelling Corrector.* URL: `http://norvig.com/spell-correct.html` (visited on 27/11/2016).

[40] Nan Hu, Ling Liu and Vallabh Sambamurthy. 'Fraud detection in online consumer reviews'. In: *Decision Support Systems* 50.3 (2011), pages 614–626.

[41] Sergey Ioffe and Christian Szegedy. 'Batch normalization: Accelerating deep network training by reducing internal covariate shift'. In: *arXiv preprint arXiv:1502.03167* (2015).

[42] Aditya Joshi, Pushpak Bhattacharyya and Mark James Carman. 'Automatic sarcasm detection: A survey'. In: *arXiv preprint arXiv:1602.03426* (2016).

[43] Nobuhiro Kaji and Masaru Kitsuregawa. 'Building Lexicon for Sentiment Analysis from Massive Collection of HTML Documents.' In: *EMNLP-CoNLL.* 2007, pages 1075–1083.

[44] Simon Kemp and Kenneth Thomas Strongman. 'Anger theory and management: A historical analysis'. In: *The American journal of psychology* (1995), pages 397–417.

[45] Farhan Hassan Khan, Saba Bashir and Usman Qamar. 'TOM: Twitter opinion mining framework using hybrid classification scheme'. In: *Decision Support Systems* 57 (2014), pages 245–257.

[46] Adam Kilgarriff. *BNC database and word frequency lists.* 1995. URL: http://www.kilgarriff.co.uk/bnc-readme.html (visited on 27/11/2016).

[47] Yoon Kim. 'Convolutional neural networks for sentence classification'. In: *arXiv preprint arXiv:1408.5882* (2014).

[48] Diederik Kingma and Jimmy Ba. 'Adam: A method for stochastic optimization'. In: *arXiv preprint arXiv:1412.6980* (2014).

[49] *Kiwix: Project Gutenberg.* URL: http://wiki.kiwix.org/wiki/Project_Gutenberg (visited on 01/12/2016).

[50] Roger J Kreuz and Richard M Roberts. 'The empirical study of figurative language in literature'. In: *Poetics* 22.1-2 (1993), pages 151–169.

[51] *Linguistic Data Consortium: Web 1T 5-gram Version 1.* 2006. URL: https://catalog.ldc.upenn.edu/LDC2006T13 (visited on 28/11/2016).

[52] Bing Liu. 'Sentiment analysis and opinion mining'. In: *Synthesis lectures on human language technologies* 5.1 (2012), pages 1–167.

[53] David Madigan. *Descriptive Modeling.* 2002.

[54] Diana Maynard and Mark A Greenwood. 'Who cares about Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis.' In: *LREC.* 2014, pages 4238–4243.

[55] Walaa Medhat, Ahmed Hassan and Hoda Korashy. 'Sentiment analysis algorithms and applications: A survey'. In: *Ain Shams Engineering Journal* 5.4 (2014), pages 1093–1113.

[56] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean. 'Efficient estimation of word representations in vector space'. In: *arXiv preprint arXiv:1301.3781* (2013).

[57] Andrés Montoyo, Patricio MartNez-Barco and Alexandra Balahur. *Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments.* 2012.

[58] Arjun Mukherjee and Bing Liu. 'Aspect extraction through semi-supervised modeling'. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1.* Association for Computational Linguistics. 2012, pages 339–348.

[59] Tony Mullen and Nigel Collier. 'Sentiment Analysis using Support Vector Machines with Diverse Information Sources.' In: *EMNLP.* Volume 4. 2004, pages 412–418.

[60] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani and Veselin Stoyanov. 'SemEval-2016 task 4: Sentiment analysis in Twitter'. In: *Proceedings of SemEval* (2016), pages 1–18.

[61] Alexander Pak and Patrick Paroubek. 'Twitter based system: Using Twitter for disambiguating sentiment ambiguous adjectives'. In: *Proceedings of the 5th International Workshop on Semantic Evaluation.* Association for Computational Linguistics. 2010, pages 436–439.

[62] Tan Pang-Ning, Michael Steinbach, Vipin Kumar et al. 'Introduction to data mining'. In: *Library of congress.* Volume 74. 2006.

[63] Robi Polikar. *Ensemble learning.* 2009. (Visited on 04/02/2017).

[64] Hadi Pouransari and Saman Ghili. *Deep learning for sentiment analysis of movie reviews.* 2014.

[65] Rudy Prabowo and Mike Thelwall. 'Sentiment analysis: A combined approach'. In: *Journal of Informetrics* 3.2 (2009), pages 143–157.

[66] Kumar Ravi and Vadlamani Ravi. 'A survey on opinion mining and sentiment analysis: tasks, approaches and applications'. In: *Knowledge-Based Systems* 89 (2015), pages 14–46.

[67] Antonio Reyes and Paolo Rosso. 'Making objective decisions from subjective data: Detecting irony in customer reviews'. In: *Decision Support Systems* 53.4 (2012), pages 754–760.

[68] Antonio Reyes, Paolo Rosso and Tony Veale. 'A multidimensional approach for detecting irony in twitter'. In: *Language resources and evaluation* 47.1 (2013), pages 239–268.

[69] Kirk Roberts, Michael A Roach, Joseph Johnson, Josh Guthrie and Sanda M Harabagiu. 'EmpaTweet: Annotating and Detecting Emotions on Twitter.' In: *LREC*. Citeseer. 2012, pages 3806–3813.

[70] *Sarcasm*. URL: `http://www.thefreedictionary.com/sarcasm` (visited on 27/10/2016).

[71] *SemEval Portal*. URL: `http://www.aclweb.org/aclwiki/index.php?title=SemEval_Portal` (visited on 03/09/2016).

[72] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov. 'Dropout: a simple way to prevent neural networks from overfitting.' In: *Journal of Machine Learning Research* 15.1 (2014), pages 1929–1958.

[73] Mihaela-Luminita Staicu and Mihaela Cutov. 'Anger and health risk behaviors'. In: *Journal of medicine and life* 3.4 (2010), page 372.

[74] Carlo Strapparava and Rada Mihalcea. 'Learning to identify emotions in text'. In: *Proceedings of the 2008 ACM symposium on Applied computing*. ACM. 2008, pages 1556–1560.

[75] Carlo Strapparava and Rada Mihalcea. 'Semeval-2007 task 14: Affective text'. In: *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics. 2007, pages 70–74.

[76] Emilio Sulis, Delia Irazú Hernández Faras, Paolo Rosso, Viviana Patti and Giancarlo Ruffo. 'Figurative messages and affect in Twitter: Differences between# irony,# sarcasm and# not'. In: *Knowledge-Based Systems* 108 (2016), pages 132–143.

[77] Harsh Thakkar and Dhiren Patel. 'Approaches for sentiment analysis on twitter: A state-of-art study'. In: *arXiv preprint arXiv:1512.01043* (2015).

[78] Abinash Tripathy, Ankit Agrawal and Santanu Kumar Rath. 'Classification of sentiment reviews using n-gram machine learning approach'. In: *Expert Systems with Applications* 57 (2016), pages 117–126.

[79] *Twitter Archiver*. 2015. URL: `https://www.labnol.org/internet/save-twitter-hashtag-tweets/6505/`.

[80] *Twitter Developer Documentation: Developer Agreement & Policy*. URL: `https://dev.twitter.com/overview/terms/agreement-and-policy` (visited on 21/10/2016).

[81] Fabricio Voznika and Leonardo Viana. *Data Mining Classification*. 2007.

[82] Bo Wang, Maria Liakata, Arkaitz Zubiaga, Rob Procter and Eric Jensen. 'SMILE: Twitter emotion classification using domain adaptation'. In: *CEUR Workshop Proceedings.* Volume 1619. Sun SITE Central Europe. 2016, pages 15–21.

[83] Wenbo Wang, Lu Chen, Krishnaprasad Thirunarayan and Amit P Sheth. 'Harnessing twitter" big data" for automatic emotion identification'. In: *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom).* IEEE. 2012, pages 587–592.

[84] *Wiktionary:Frequency lists.* URL: `https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists` (visited on 28/11/2016).

[85] Deirdre Wilson. 'The pragmatics of verbal irony: Echo or pretence?' In: *Lingua* 116.10 (2006), pages 1722–1743.

[86] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2005.

[87] Chew-Yean Yam. *Emotion Detection and Recognition from Text Using Deep Learning.* 2015. URL: `https://www.microsoft.com/developerblog/real-life-code/2015/11/30/Emotion-Detection-and-Recognition-from-Text-using-Deep-Learning.html` (visited on 12/01/2017).

[88] Ye Zhang and Byron Wallace. 'A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification'. In: *arXiv preprint arXiv:1510.03820* (2015).

Acronyms

APA American Psychological Association. 10

API Application Program Interface. 22

BNC British National Corpus. 24

CNN Convolutional Neural Networks. 13, 24, 25, 37

CW Content Word. 12

DL Deep Learning. 11–13, 19, 21, 22, 37

DNN Deep Neural Networks. 13

DT Decision Tree. 12, 15

FN Flase Positive. 15

FP False Positive. 15

HFW High Frequency Word. 12

HIT Human Intelligence Task. 11, 21, 27

HTML Hypertext Markup Language. 24

KNN K-Nearest Neighbors. 12, 37

LDA Latent Dirichlet Allocation. 11

LSA Latent Semantic Analysis. 11

LSTM Long Short Term Memory. 13

ML Machine Learning. 7, 10, 19

NB Naive Bayes. 8, 11, 12

NLP Natural Language Processing. 5, 6

NN Neural Network. 11, 15, 19

OCC Ortony Clore and Collins. 9, 10

OCR Optical Character Recognition. 13

OM Opinion Mining. 5

PMI Point-wise Mutual Information. 11

POS Part of Speech. 12

SA Sentiment Analysis. 5–7, 9, 10

SVM Support Vector Machine. 9, 11–13, 15, 19, 37

TN True Negative. 15

TP True Positive. 15

URL Uniform Resource Locator. 8, 22, 24

WEKA Waikato Environment for Knowledge Analysis. 37

WN Word Net. 11