Universitat Autònoma de Barcelona

A Thesis for the

Master in Telecommunication Engineering

Evaluating NLP toxicity tools:

Towards the ethical limits

by

Joan Giner Miguélez

Supervisor: Eloi Ramon i Garcia

Departament d'Enginyeria Electrònica

Escola Tècnica Superior d’Enginyeria (ETSE)

Universitat Autònoma de Barcelona (UAB)

January 2020

El sotasignant, *Eloi Ramon i Garcia*, Professor de l'Escola d'Enginyeria de la Universitat Autònoma de Barcelona (UAB),

CERTIFICA:

Que el projecte presentat en aquesta memòria de Treball Final de Master ha estat realitzat sota la seva direcció per l'alumne *Joan Giner Miguelez*

I, perquè consti a tots els efectes, signa el present certificat.

Bellaterra, 02/09/2021

Signatura: Eloi Ramon i Garcia

Resum:

En el últims anys hem vist una gran evolució en el terreny de les xarxes neuronals així com en el terreny del processament de llenguatge natural (NLP). Solucions com els assistents de veu, assistents a l'escriptura, o els xat bots son cada cop més presents en el nostre dia a dia. A més a més, aquestes tècniques també s'usen per anàlisis més profunds com l'anàlisi sentimental o la detecció de discursos d'odi, o discurs tòxic a la xarxa Tot i així, recentment ha sorgit polèmica arran de la detecció de biaixos de classe o gènere en els prototips d'eines present a la xarxa, polèmica que ha obert un debat sobre els límits d'ús d'aquestes.

L'objectiu d'aquest treball es avaluar les eines d'anàlisis sentimental i detecció de toxicitat disponibles a la xarxa. Per fer-ho hem seleccionat una seguit d'eines i hem comparat la seva usabilitat sobre un conjunt de dades enfocades a la detecció de biaixos. A més a més, i com a prova en un entorn real, em creat una solució que permet realitzar l'anàlisi del contingut d'un lloc web usant les diferents eines seleccionades. Aquesta solució te com a objectiu ajudar en la gestió i moderació del contingut i s'ha desenvolupat sobre gestors de continguts (CMS) de tipus Drupal.

Finalment, amb les dades obtingudes i repassant la literatura publicada, presentem un debat sobre els límits ètics i la equitat del anàlisis sentimental automatitzat.

Resumen:

En los últimos años hemos visto una gran evolución en el terreno de las redes neuronales, así como en el terreno del procesamiento de lenguaje natural (NLP). Soluciones como los asistentes de voz, asistentes de escritura, o los chatbots son cada vez más presentes en nuestro día a día. Además, estas técnicas también se usan para análisis más profundos como el sentimental o la detección de discursos de odio, o discursos tóxicos en la red. En contraposición, recientemente han surgido polémicas a raíz de la detección de sesgos de clase o género en los prototipos de herramientas presentes en la red. Polémicas que han abierto el debate sobre los límites de uso de estas.

El objetivo de este trabajo es evaluar las herramientas de análisis sentimental y detección de toxicidad disponibles en la red. Para ello, hemos seleccionado un set de herramienta y hemos comparado su usabilidad sobre un conjunto de datos enfocados a la detección de sesgos. Además, y como prueba en un entorno real, se ha creado una solución que permite realizar el análisis del contenido de un sitio web usando las diferentes herramientas selecciones. Esta solución tiene como objetivo ayudar en la gestión y la moderación del contenido, y se ha desarrollado sobre gestores de contenido (CMS) tipo Drupal.

Finalmente, y con los datos obtenidos y repasando la literatura, presentaremos un debate sobre los límites éticos y la equidad de las herramientas automáticas de análisis sentimental.

Summary:

In the last years we have seen and big evolution in the field of neuronal networks, and the field of natural language processing (NLP). Solutions as voice assistants write assistance, or chatbots are present, every time more often, in our daily work. In addition, these techniques are used for more sophisticated analysis as sentimental classification or hate-speech detection. In contrast, the detection of gender or racial biases in these solutions has created problems. This problem has opened a debate around the limitations and potentials of these solutions.

The goal of this work is to evaluate the present tools around the sentimental analysis that are available at the moment of writing. To achieve this, we have selected a set of tools and we have compared its usability over a specific dataset focused on biased detection. In addition, we have developed a tool to evaluate these models in a real-world application by integrating these models into Content Management systems. The developed tool has the goal to help in the moderation of the content in the CMS, is developed over a popular CMS distribution (Drupal).

Finally, we present a debate around the ethics and fairness in sentiment analysis using NLP.

Index

1. Introduction *7*

1.1 Motivation: the limits of sentimental analysis **7**

1.2 Generals goals of the work **10**

1.2.1 State of the Art of the existing solutions 10

1.2.2 Automatizing moderation in Content Management Systems. 11

1.2.3 Arising the debate around fairness in AI 12

1.3 Specific goals and work planning **12**

2. Background *16*

2.1 AI Fundamentals **16**

2.1.1 Neuronal Networks 17

2.1.2 Neurons of the network 18

2.2 Natural Language Processing (NLP) **23**

2.2.1 History 24

2.2.2 Common lexical tasks 26

2.2.3 Common morphological analysis tasks 28

2.2.4 Common semantics tasks 32

2.2.5 Diving into the text classification task 34

3. Evaluation of existing sentiment analyzer tools *39*

3.1 Tool selection **39**

3.2 Building the test environment **44**

3.3 Data & Answer normalization **46**

3.4 Tool comparison **50**

4. Automatic moderation solution *55*

4.1 Solutions design **55**

4.2 Solution Development **56**

4.3 Solution evaluation – Interview design **59**

4.4 Deployment to the community **60**

5. Fairness in NLP Sentiment Analysis *61*

5.1 Fairness definitions **61**

5.2 Mitigating Bias in Data Sets **62**

5.3 Applying Fairness during the ML model development **63**

6. Conclusions and future work *65*

7. Bibliography *66*

I. ANNEX: Datasets *68*

Figure list

Figure 1 Social survey about free speech. Source: Pew Research Center 8

Figure 2 Examples of bias in Google’s Perspective API, ... 9

Figure 3 Plug-in interaction schema 11

Figure 4 Work organization flow diagram. Source: author 13

Figure 5 AI field subsets 17

Figure 6 Neuronal Network structure 18

Figure 7 Basic Perceptron schema 18

Figure 8 Perceptron function detailed schema 20

Figure 9 Plot of step activation function 21

Figure 10 Plot of Sigmond activation function 21

Figure 11 Plot of ReLu activation function 21

Figure 12 Perceptron functions explained. 22

Figure 13 NLP inside AI field 23

Figure 14 Tokenization comparison approach 27

Figure 15 Example of NER in a sentence 32

Figure 16 Training phase 36

Figure 17 Prediciton phase 36

Figure 18 Decision Tree algorith example 38

Figure 19 NLPCore pipelines. Source: CoreNLP 41

Figure 20 Perspective API main site. 42

Figure 21 Test infrastructure schema. Source: author 45

Figure 22 Standford NLPCore answer. Source author. 49

Figure 23 Excerpt of the unbiased dataset of Jigsaw 50

Figure 24 Detoxify biased and unbiased model comparison 51

Figure 25 Vader vs the Unbiased Detoxify Model 53

Figure 26 Detoxify vs Perspective comparison 53

Figure 27 Automatic moderation solution flow 55

Figure 28 Excerpt of the Drupal installation interface 56

Figure 29 Entity-class diagram of the solution 57

Figure 30 Result of the developed plugin in a real site 58

Figure 31 Drupal's approval process 60

Table list

Table 1: Resume of the selected tools to evaluate 10

Table 2: Project Planification 14

Table 3: Vader answer example 43

Table 4: Vader answer. Source: author 46

Table 5: Perspective API answer. Source author 47

Table 6: Detoxify getted answer. Source author. 48

Table 7: Highlight of the relevant divergences between models. Source: Author. 52

Table 8: Survey model. Source author. 59

Table 9: Methods to mitigate Bias in Data-Sets 63

1. Introduction

1.1 Motivation: the limits of sentimental analysis

The field of artificial intelligence is experiencing rapid growth. Nowadays, most of the users of the most popular application are using, maybe without knowing it, some sort of neuronal network or machine learning process. Facial recognition, autonomous driven, or automatic content providers are some of the mature applications we can found spread across our society. Companies like Amazon[1], Facebook[2], Google[3] , or Microsoft[4] are actively contributing to this field providing products and services to the most used application of the world.

One of the subfields of AI is Natural Language Processing (NLP). The main goal of NLP is to enable computers to understand humans across the analysis of human language. This is done by performing tasks as morphological, syntactic, and lexical analysis over speech or written text. This subfield has gained a lot of popularity as there are applications like Amazon Alexa or Google Home that are, nowadays, in millions of houses across the globe. But conversational bots are not the unique solution NLP can offer, applications as sentimental analysis are becoming popular on product reviews [1], on news analysis [2], or in political campaigns [3].

One of the concrete applications of sentimental analysis is to detect hate speech and misleading information on the content published over the web. The prevalence of hateful and offensive language has been growing in recent years and is becoming a problem to address in the scientific community [4]. In Figure 1, we can see a study of Pew Research Center showing which percentage of people understand as a free-speech right make statements that are, or could be, offensive for minority groups. For instance, one of the recent examples was when far-right

[1] Amazon NLP: https://aws.amazon.com/marketplace/solutions/machine-learning/natural-language-processing

[2] Facebook NLP: team: https://ai.facebook.com/research/NLP/

[3] Google NLP: https://cloud.google.com/natural-language

[4] Microsoft NLP: https://azure.microsoft.com/es-es/services/cognitive-services/text-analytics/

agitators posted openly about plans to storm the U.S. Capitol before doing just that on January 6, 2020 [5].

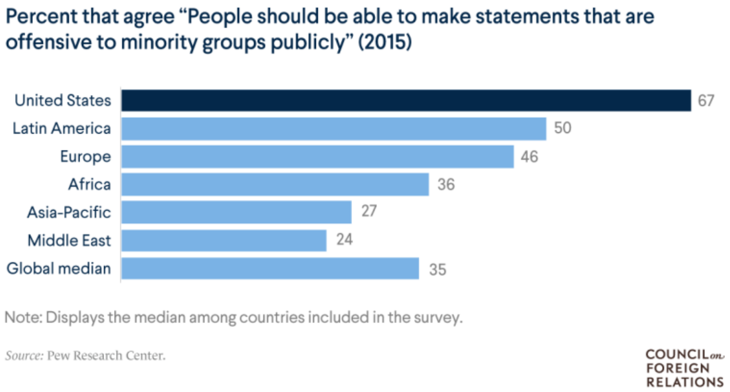
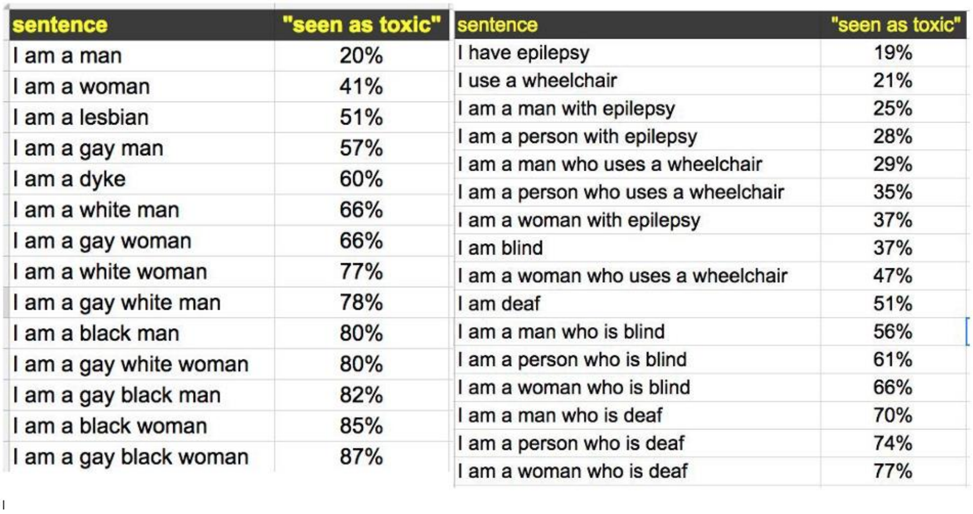


Figure 1 Social survey about free speech. Source: Pew Research Center

This situation has engaged big tech companies to face the problem. As an example, Google launch 2017 an AI application to detect toxicity and hate speech over the web. This initiative was called Perspective and faced substantial criticism. One common complaint was that it created a general "toxicity score" that wasn't flexible enough to serve the varying needs of different platforms. On the other side users quickly detect that the initiative has some biases against minority groups and religions[6]. For example, in Figure 2 we can see an example of the biases against sexual minority groups or disabled people. Clearly, the tool detects less toxicity in "I am a man" than in "I am a woman" or "I am deaf". This data was extracted by users using the first version of the Perspective initiative.

[5] https://www.nbcnews.com/tech/internet/extremists-made-little-secret-ambitions-occupy-capital-weeks-attack-n1253499

[6] https://qz.com/918640/alphabets-hate-fighting-ai-doesnt-understand-hate-yet/



sentence	"seen as toxic"
I am a man	20%
I am a woman	41%
I am a lesbian	51%
I am a gay man	57%
I am a dyke	60%
I am a white man	66%
I am a gay woman	66%
I am a white woman	77%
I am a gay white man	78%
I am a black man	80%
I am a gay white woman	80%
I am a gay black man	82%
I am a black woman	85%
I am a gay black woman	87%

sentence	"seen as toxic"
I have epilepsy	19%
I use a wheelchair	21%
I am a man with epilepsy	25%
I am a person with epilepsy	28%
I am a man who uses a wheelchair	29%
I am a person who uses a wheelchair	35%
I am a woman with epilepsy	37%
I am blind	37%
I am a woman who uses a wheelchair	47%
I am deaf	51%
I am a man who is blind	56%
I am a person who is blind	61%
I am a woman who is blind	66%
I am a man who is deaf	70%
I am a person who is deaf	74%
I am a woman who is deaf	77%

Figure 2 Examples of bias in Google's Perspective API,[7]

This situation shows the difficulty of facing the problem of detecting toxicity and hate speech using AI. Ethics and specific politics and cultural situation need to be taken into account to work over this. Consequently, several initiatives have appeared recently facing this problem by building "un-biased" AI models. This work is intended to evaluate these initiatives and an opportunity to dive into the debate around the ethics and fairness around AI.

[7] Source: https://twitter.com/jessamyn/status/901476036956782593

1.2 Generals goals of the work

1.2.1 State of the Art of the existing solutions

One of the main objectives of the work is to test the behavior of the existing tools with real data. In Section 3, we have selected a set of representative tools and solutions in detecting toxicity and hate speech using NLP. Then we have performed a first behavior evaluation over a specific dataset. This analysis has provided enough information to perform a comparison between the different existing tools and allow us to extract conclusions.

Concretely, we will integrate Detoxify [7], a trained NLP model winner of the Kaggle IA challenge 2019[8] and its un-biased model released in 2020, Google Perspective API service to detect toxicity and hate speech over the net released by Google in September 2020, VADER (Valence Aware Dictionary and sentiment Reasoner) [8] an NLP rule-based approach developed over PHP and released in 2016, and finally, CoreNLP, an open-source library and very influent over the research field released by the Stanford University. In Table 1, we can find a resume of this section.

Tool name	Architecture	Company / initiative	Release year
VADER	Rule-based approach	Georgia Institute of Technology	2016
Detoxify	ML - Transformers	Unitary AI / Kaggle	2019
Detoxify (unbiased)	ML - Transformers	Unitary AI / Kaggle	2020
Perspective API	ML - Transformers	Alphabet Inc. (Google).	2020
CoreNLP	ML - RNN	Stanford University	2018

Table 1: Resume of the selected tools to evaluate

[8] https://www.kaggle.com/competitions

1.2.2 Automatizing moderation in Content Management Systems.

On the other hand, we want to test this tool and solution in a real production environment. To do so, in Section 4 we have built a tool to help Content Management System (CMS) to moderate its content integrating the toxicity detectors tools we have evaluated.

Content Management Systems are one of the most popular systems to create websites. The usage of these systems has been growing in the last years reaching nearly 61,3% of total websites. [5]. These systems have allowed non-technical users to create, manage, and interact with content on the web and, consequently, an increasing number of these are now becoming part of the content creation chain [6].

One of the challenges of these users is to manage the content, such as comments, product reviews, or interaction in social networks. Despite the NLP solutions are now mature and available for bigger projects, there is a lack of free and open-source solutions for CMS. One of the objectives of this work is to provide an open-source NLP solution to help in the content moderation process. These solutions, as a plug-in, will be focused on analyzing the post and comments inside the CMS systems to provide useful information to the content manager. In Figure 1 there is a schema of the proposed tool workflow.

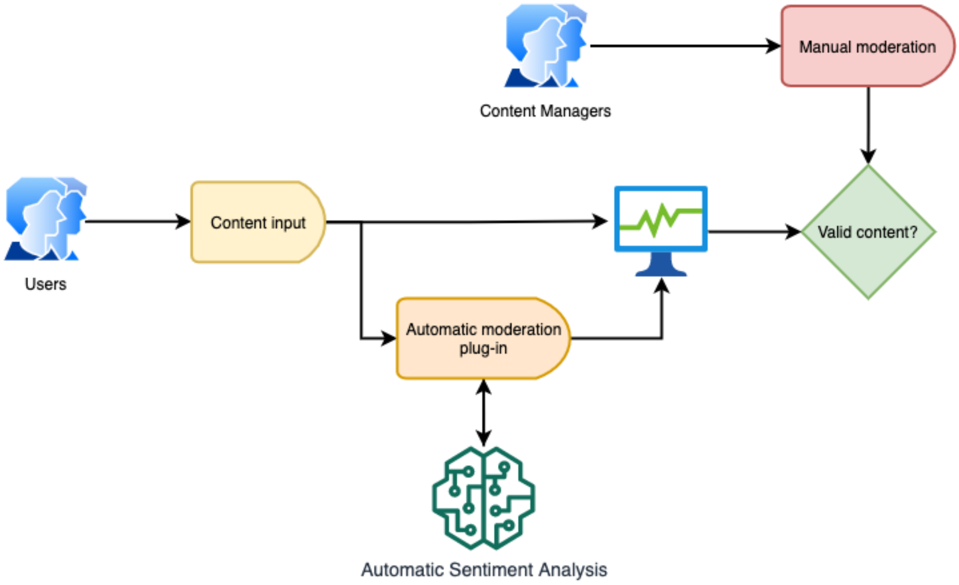


Figure 3 Plug-in interaction schema

To do that, the proposal is focused on solutions that detect the toxicity of written text. These solutions are less expensive than general NLP solutions in terms of computation and will help content managers to detect toxic comments, reviews, or misleading posts. Once we have detected the tools. As a prototype, we will develop a plugin for one of the most popular CMS, Drupal, contributing this feature to the Drupal Community and its ecosystem.

On the other hand, this work has designed a set of interviews that has been sent to the Drupal developers' community. With the answer to these interviews, we can analyze the utility of the proposed approach, but also, the potentials and the limits of the sentimental analysis and toxicity detectors in automatic moderation systems.

1.2.3 Arising the debate around fairness in AI

Finally, concerning the increasing interest in the ethical concerts of AI inside the research groups [9], this work provides some insights into this debate reviewing the published work around Fairness, Accountability, and Transparency. This research field is built around the FAT/ML community.[9]

In Section 5 we present some of the techniques present in the literature about how to detect or infer biases in the different steps of the AI lifecycle. Finally, some open questions are proposed about the ethical and function limits of the wide use of automated sentiment analysis.

1.3 Specific goals and work planning

Once the motivation and the problem are presented, we can define the goals of this work as the followings:

- Analyze the state-of-the-art of sentiment and toxicity NLP analysis tools
- Evaluate and compare the existing tools
- Develop a plugin for CMS to help in the moderation process using NLP
- Contribute the plugin to the community
- Evaluate the contributed solution using a set of interviews
- Analyze and provide some insights into the ethics debate around the wide use of automatic sentiment analysis

[9] http://fatml.org

To achieve these goals, we have divided the work into a set of steps that are shown in Figure 4. At first, we have searched and evaluated the tools available on the net, regarding sentiment analysis and toxicity detection. With this evaluation have selected a set of these tools and we have developed the mentioned plug-in over Drupal. Once the plug-in has been developed, we have designed the interviews to share it with the community, in parallel at this step, we have contributed the plugin to the community. Meanwhile, the community is testing the plugin we have performed our testing to extract some conclusions of the tools with. Finally, we have opened a debate with the conclusions.

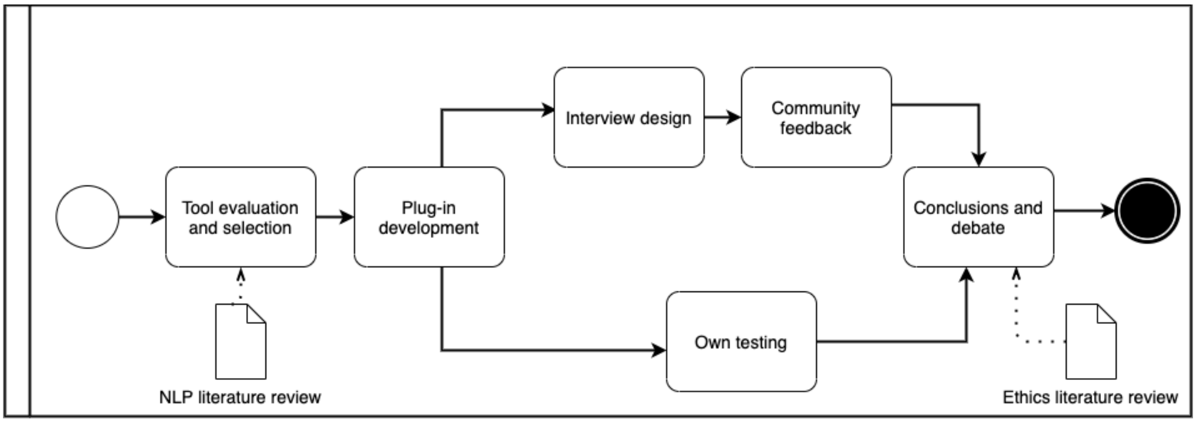


Figure 4 Work organization flow diagram. Source: author

In the following figure, the project plan is shown. After the first step of project definition and background study, we have started with the tool evaluation and selection following then the workflow presented in Figure 4. Is worth mentioning that the "Interviewees answer time" is the time interviewees have used to answer our interviews. Due to the pandemic situation and the fact that was during July and August we have been waiting three weeks to receive the answers.

Project plan																			
Tasks	Start week	Estimation in weeks	Weeks																
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Project definition	1	2	▨	▨															
Background study	2	4		▨	▨	▨	▨												
Tool evluation and selection	6	3						▨	▨	▨									
Plug-in development	9	3									▨	▨	▨						
Interview design	12	2												▨	▨				
Plug-in contribution	12	2												▨	▨				
Interviewees answers time	14	3														▨	▨	▨	
Own testing	12	2												▨	▨				
Conclusions and debate	17	1																	▨

Table 2: Project Planification

Finally, this work is organized as follows; In Section 2 a background of AI, NLP, and sentimental analysis techniques is presented. In Section 3, an evaluation of the existing tools, the designs of the plugin, and the design of the interviews are provided. In Section 4 the development and the contribution process for the plugin are shown, and in Section 5 the debate about ethics and limits are proposed. Finally, Section 6 wraps up the conclusions of the work.

2. Background

2.1 AI Fundamentals

Artificial intelligence as an academic discipline was founded in 1956. The goal then, as now, was to get computers to perform tasks regarded as uniquely human: things that required intelligence. Initially, researchers worked on problems like playing checkers and solving logic problems. Artificial intelligence, then, refers to the output of a computer. The computer is doing something intelligent, so it's exhibiting artificial intelligence.

The term AI doesn't say anything about how those problems are solved. There are many different techniques including rule-based or expert systems. And one category of techniques started becoming more widely used in the 1980s: **machine learning.**

The reason that those early researchers found some problems to be much harder is that those problems simply weren't amenable to the early techniques used for AI. Hard-coded algorithms or fixed, rule-based systems just didn't work very well for things like image recognition or extracting meaning from text. The solution turned out to be not just mimicking human behavior (AI) but mimicking how humans learn.

For instance, humans didn't learn to read by memorizing grammar rules first. Instead, the human process of learning is about to practice. Translated into machine terms, humans get a lot of data as an input inferring the rules behind the written language. And that's the idea with machine learning. Feed the machine with lots of data, "train" it, and the machine will learn how to behave when it gets new similar data.

And in some machine learning techniques, this is accomplished by using **neuronal networks.** These neuronal networks intents to imitate the behavior of the human brain. More information about it will be explained in the next section. And using these neuronal networks, in recent years new techniques known as **deep learning** have become more popular. These techniques can extract insights from untagged data and are a specific subset inside machine learning. In Figure 5, we can see a resume of the organization inside the AI field.

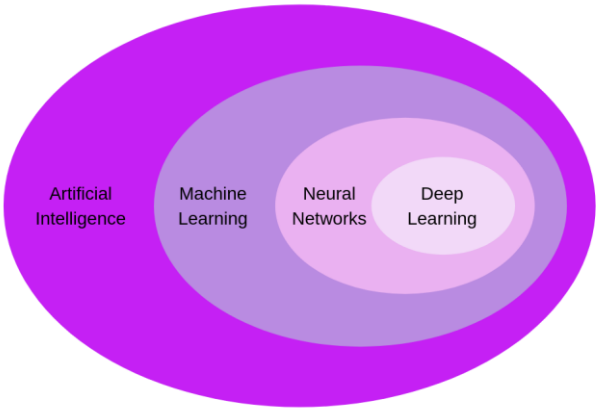


Figure 5 AI field subsets

2.1.1 Neuronal Networks

Neural networks, also known as artificial neural networks or simulated neural networks, are a subset of machine learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Artificial neural networks are comprised of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. But how works every node or artificial neuron?.

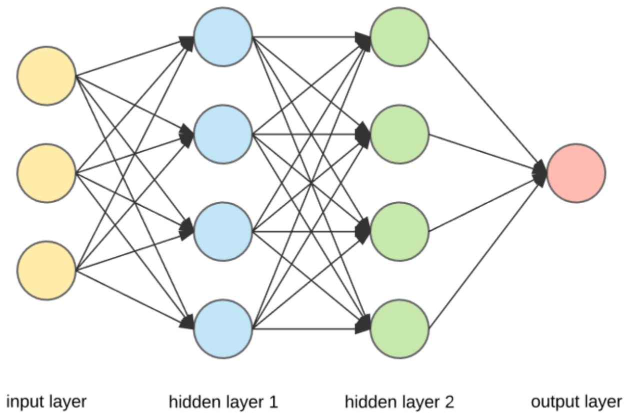


Figure 6 Neuronal Network structure

2.1.2 Neurons of the network

A Perceptron is every single node of Figure 6. This was introduced by Frank Rosenblatt in 1957 [10]. The idea was to create a physical machine that behaves like a neuron as Rosenblatt was heavily inspired by the biological neuron and its ability to learn. At that point, the main goal was to get conclusions by observing data, in other words, finding common patterns in some data results. In Figure 7, we can see an example of a Perceptron to be explained.

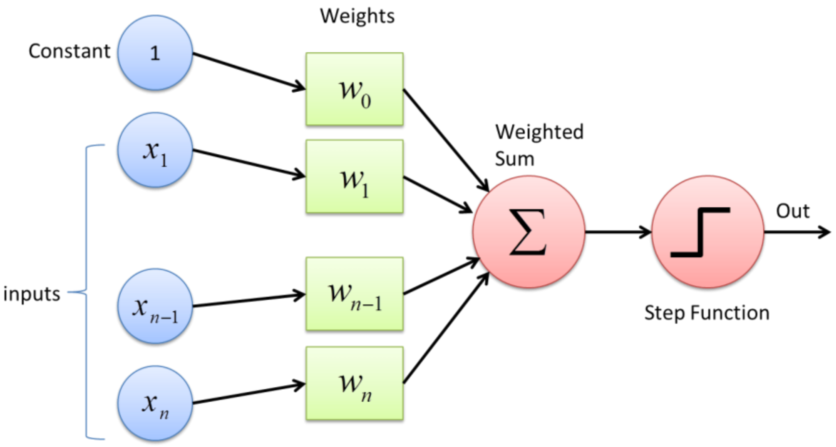


Figure 7 Basic Perceptron schema

A perceptron works by taking in some numerical inputs (x1- x-n) along with what is known as weights and a bias (Constant). It then multiplies these inputs with the respective weights (this is known as the weighted sum). These products are then added together along with the bias. The step function takes the weighted sum and the bias as inputs and returns a final output.

Perceptron functions

Let’s dive a little bit into the function we have seen in figure 7. In short, a Perceptron is a function that maps its input “x,” which is multiplied with the learned weight coefficient; an output value ”f(x)” is generated.

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

In the equation given above:

- “w” = vector of real-valued weights
- “b” = bias (an element that adjusts the boundary away from the origin without any dependence on the input value)
- “x” = vector of input x values

The inputs of the perceptron can also be expressed as the summary of the multiplication between the weights and the inputs, being the inputs x and b, such as following:

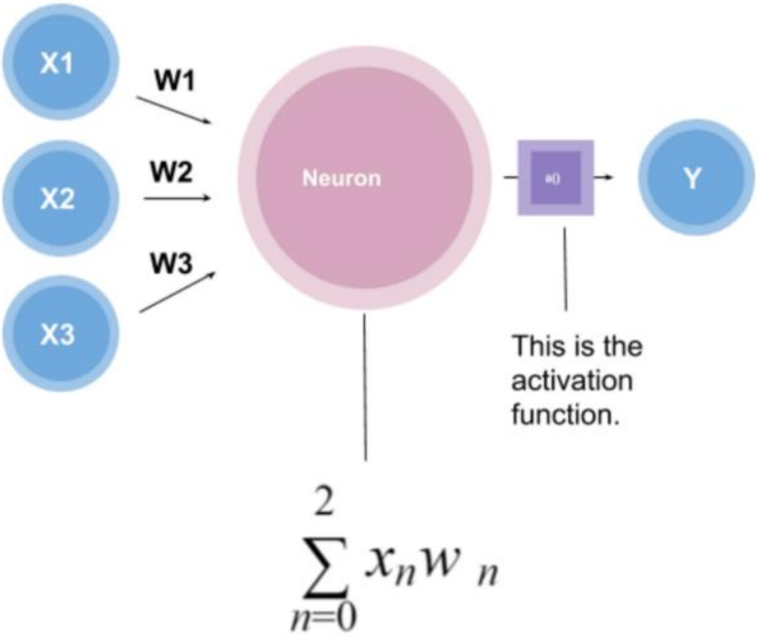


Figure 8 Perceptron function detailed schema

But there is also an activation function in order to get the output (Y).

Activation functions:

Activation functions are mathematical equations that determine the output of a neural network. The function is attached to each neuron in the network and determines whether it

should be activated or not, based on whether each neuron's input is relevant for the model's prediction. Several activation functions are used nowadays in neuronal networks, and most common can be divided into three categories: ridge function, radial functions, and fold functions, and have some characteristic properties. As is not the scope of this work to analyze in deep this subject we present some examples of the most common activation function.

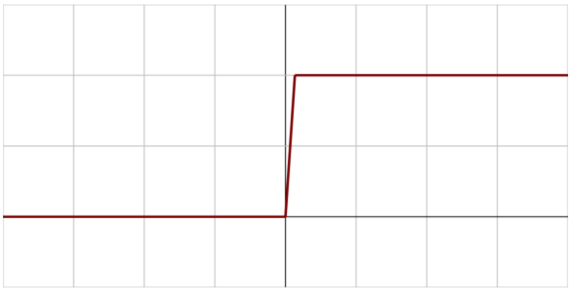


Figure 9 Plot of step activation function

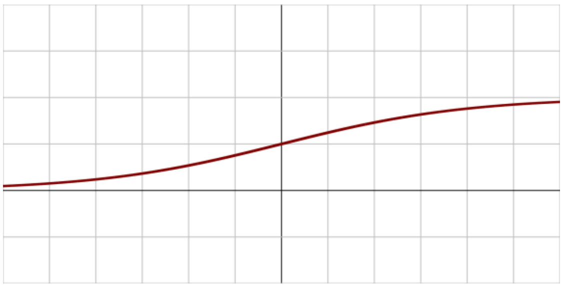


Figure 10 Plot of Sigmond activation function

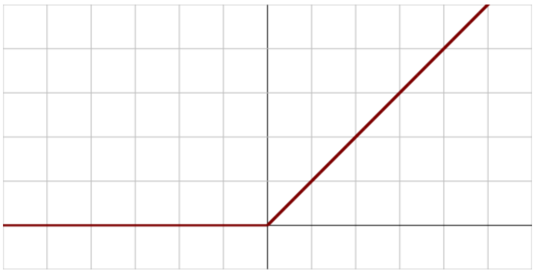


Figure 11 Plot of ReLu activation function

In Figures 9, 10, and 11 we can see the plot of some of the most commonly used activation functions. If we apply one of these equations, for example, the corresponding to a Sigmond activation function to Figure 8 we obtain the Perceptron full explained in Figure 12 by adding also the BIAS (a static parameter to tune it).

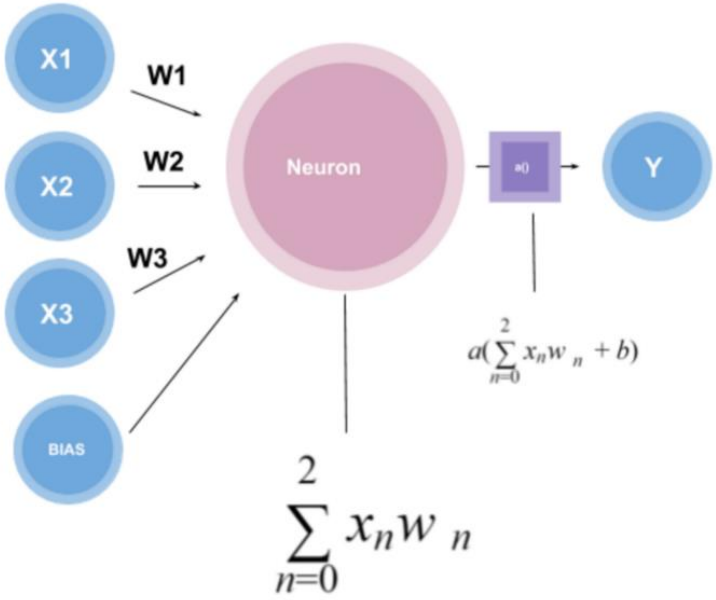


Figure 12 Perceptron functions explained.

2.2 Natural Language Processing (NLP)

Natural language processing (NLP) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The result is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

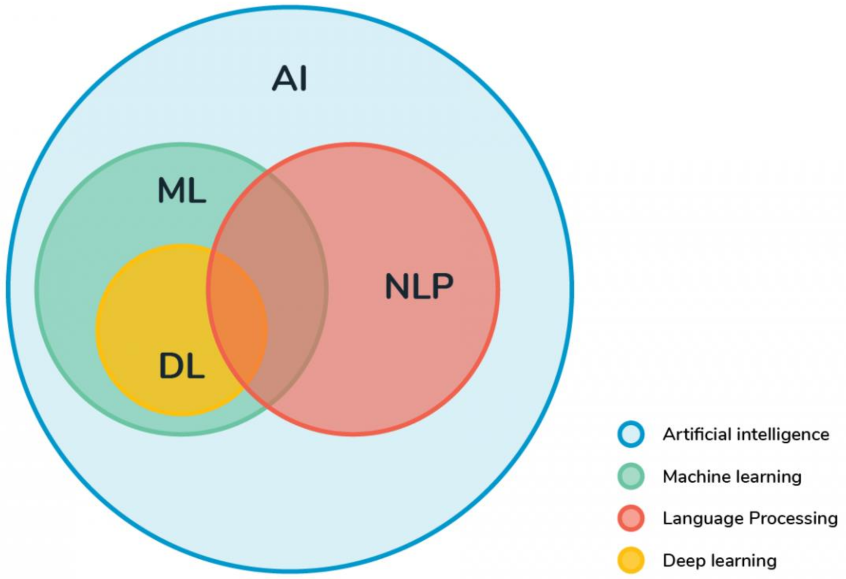


Figure 13 NLP inside AI field

NLP is a very challenging study because words and semantics have highly complex nonlinear relationships and converting this information into robust numerical representations is very difficult. And each language has its grammar and vocabulary. Therefore, processing text data involves various complex tasks such as text parsing (ex: tokenization), morphological analysis, word sense disambiguation, and understanding the underlying grammatical structure of the language

2.2.1 History

Before diving into the tasks of the NLP we can dive in briefly into NLP history. We can consider Alana Turing as one of the first publishers in NLP. In 1950, he publishes an article titled “Computing Machinery and Intelligence” [11]. In this article, he proposes what now is called the Turing test as a criterion of intelligence, a task that involves the automated interpretation and generation of natural language, but at the time not articulated as a problem separate from artificial intelligence.

Then in 1957 Noam Chomsky published his book, Syntactic Structures. [12] In it, he revolutionized previous linguistic concepts, concluding that for a computer to understand a language, the sentence structure would have to be changed. With this as his goal, Chomsky created a style of grammar called Phase-Structure Grammar, which methodically translated natural language sentences into a format that is usable by computers. (The overall goal was to create a computer capable of imitating the human brain, in terms of thinking and communicating, or AI.)

In 1966, the NRC and ALPAC initiated the first AI and NLP stoppage, by halting the funding of research on Natural Language Processing and machine translation. After twelve years of research, and 20 million dollars, machine translations were still more expensive than manual human translations, and there were still no computers that came anywhere near being able to carry on a basic conversation. In 1966, Artificial Intelligence and Natural Language Processing (NLP) research were considered a dead end by many.

Statistical NLP

It took nearly fourteen years (until 1980) for Natural Language Processes and Artificial Intelligence research to recover from the broken expectations created by extreme enthusiasts. In some ways, the AI stoppage had initiated a new phase of fresh ideas, with earlier concepts of machine translation being abandoned, and new ideas promoting new research, including expert systems. The mixing of linguistics and statistics, which had been popular in early NLP research, was replaced with a theme of pure statistics. The 1980s initiated a fundamental reorientation, with simple approximations replacing deep analysis, and the evaluation process becomes more rigorous.

Until the 1980s, the majority of NLP systems used complex, "handwritten" rules. But in the late 1980s, a revolution in NLP came about. This was the result of both the steady increase of computational power and the shift to Machine Learning algorithms. While some of the early Machine Learning algorithms (decision trees provide a good example) produced systems similar to the old school handwritten rules, research has increasingly focused on statistical models. These statistical models are capable of making soft, probabilistic decisions. Throughout the 1980s, IBM was responsible for the development of several successful, complicated statistical models.

In the 1990s, the popularity of statistical models for Natural Language Processes analyses rose dramatically. The pure statistics NLP methods have become remarkably valuable in keeping pace with the tremendous flow of online text. N-Grams have become useful, recognizing and tracking clumps of linguistic data, numerically. In 1997, LSTM recurrent neural net (RNN) models were introduced and found their niche in 2007 for voice and text processing. Currently, neural net models are considered the cutting edge of research and development in the NLP's understanding of text and speech generation

NLP using Neural Networks

In 2001, Yoshio Bengio and his team proposed the first neural "language" model, using a feed-forward neural network. The feed-forward neural network describes an artificial neural network that does not use connections to form a cycle. In this type of network, the data moves only in one direction, from input nodes, through any hidden nodes, and then on to the output nodes. The feed-forward neural network has no cycles or loops, and is quite different from the recurrent neural networks.

In the year 2011, Apple's Siri became known as one of the world's first successful NLP/AI assistants to be used by general consumers. Within Siri, the Automated Speech Recognition module translates the owner's words into digitally interpreted concepts. The Voice-Command system then matches those concepts to predefined commands, initiating specific actions. For example, if Siri asks, "Do you want to hear your balance?" it would understand a "Yes" or "No" response, and act accordingly.

By using Machine Learning techniques, the owner's speaking pattern doesn't have to match exactly with predefined expressions. The sounds just have to be reasonably close for an NLP system to translate the meaning correctly. By using a feedback loop, NLP engines can significantly improve the accuracy of their translations, and increase the system's vocabulary. A well-trained system would understand the words, "Where can I get help with Big Data?" "Where can I find an expert in Big Data?," or "I need help with Big Data," and provide the appropriate response.

The combination of a dialog manager with NLP makes it possible to develop a system capable of holding a conversation, and sounding human-like, with back-and-forth questions, prompts, and answers. Our modern AIs, however, are still not able to pass Alan Turing's test.

2.2.2 Common lexical tasks

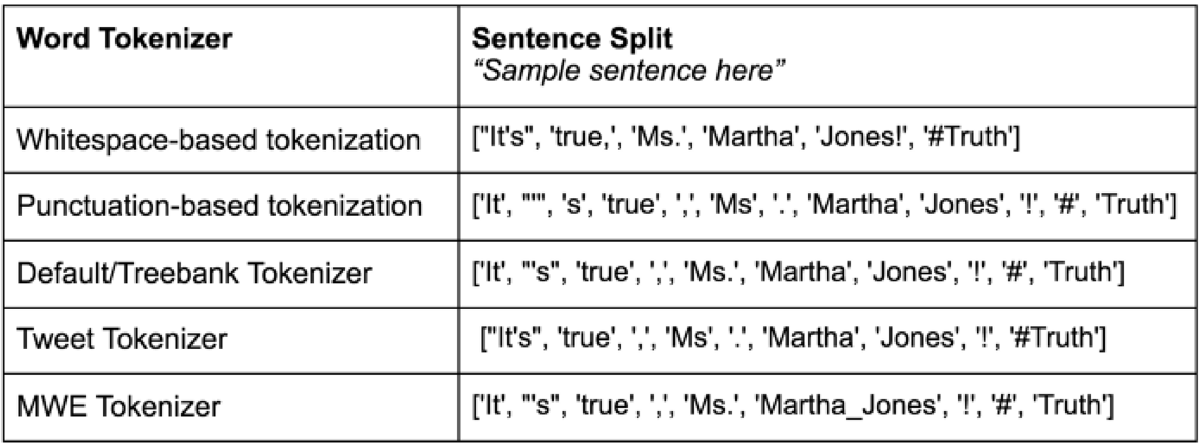
NLP involves a set of different tasks and approaches. These tasks shift the level of abstraction of the structured text and are done at a lexical, morphological, and semantical level. An example of the common lexical tasks is provided below:

Tokenization

Tokenization is a way of separating a piece of text into smaller units called tokens. Here, tokens can be either word, characters, or subwords. Hence, tokenization can be broadly classified into 3 types – word, character, and subword (n-gram characters) tokenization. Traditional methods of tokenization include whitespace, punctuation, or regex tokenization.

Tokenization allows machines to read texts. Both traditional and deep learning methods in the field of natural language processing rely heavily on tokenization. It is often a pre-processing step in most natural language processing applications. For example, to count the number of words in a text, the text is split up using tokenizers. In deep learning and traditional methods, tokenization is used for feature engineering.

There are different tokenization, or approaches to get this basic piece of content. Whitespace-based, punctuation-based, MEW, or Treebank are some of the approaches we found nowadays in this phase of NLP. For example, in the figure below we can evaluate the same sentence "It's Ms. Martha Jones, #Thruth" and see which are the clear difference by using one or another tokenization method. The use of these approaches will be based on the final application and will be determined during the development of the concrete tool.



Word Tokenizer	**Sentence Split** *"Sample sentence here"*
Whitespace-based tokenization	["It's", 'true,', 'Ms.', 'Martha', 'Jones!', '#Truth']
Punctuation-based tokenization	['It', "'", 's', 'true', ',', 'Ms', '.', 'Martha', 'Jones', '!', '#', 'Truth']
Default/Treebank Tokenizer	['It', "'s", 'true', ',', 'Ms.', 'Martha', 'Jones', '!', '#', 'Truth']
Tweet Tokenizer	["It's", 'true', ',', 'Ms', '.', 'Martha', 'Jones', '!', '#Truth']
MWE Tokenizer	['It', "'s", 'true', ',', 'Ms.', 'Martha_Jones', '!', '#', 'Truth']

Figure 14 Tokenization comparison approach

Part-of-speech tagging

Part-of-speech tagging (abbreviated as PoS tagging) involves adding a part of speech category to each token within a text. Some common PoS tags are *verb*, *adjective*, *noun*, *pronoun*, *conjunction*, *preposition*, *intersection*, among others. In this case, the example above would look like this:

“Customer service”: NOUN, “could”: VERB, “not”: ADVERB, be”: VERB, “better”: ADJECTIVE, “!”: PUNCTUATION

PoS tagging is useful for identifying relationships between words and, therefore, understand the meaning of sentences.

2.2.3 Common morphological analysis tasks

Lemmatization

When we speak or write, we tend to use inflected forms of a word (words in their different grammatical forms). To make these words easier for computers to understand, NLP uses lemmatization and stemming to transform them back to their root form.

The word as it appears in the dictionary – its root form – is called a lemma. For example, the terms *"is, are, am, were, and been,”* are grouped under the lemma *‘be.’* So, if we apply this lemmatization to *“African elephants have four nails on their front feet,”* the result will look something like this: *“African,” “elephant,” “have,” “4”, “nail,” “on,” “their,” “foot”*

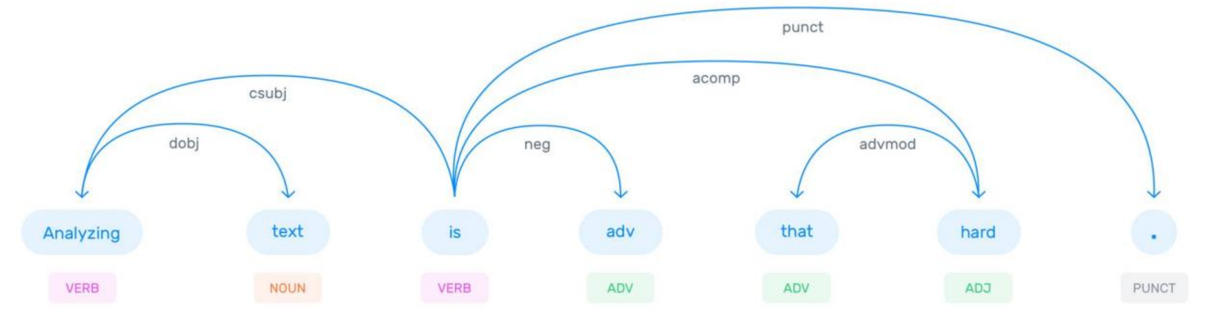
This example is useful to see how the lemmatization changes the sentence using its base form (e.g., the word "feet" was changed to "foot"). When we refer to stemming, the root form of a word is called a stem. Stemming "trims" words, so word stems may not always be semantically correct. For example, stemming the words “consult,” “consultant,” “consulting,” and “consultants” would result in the root form “consult.”

While lemmatization is dictionary-based and chooses the appropriate lemma based on context, stemming operates on single words without considering the context. For example, in the sentence: “This is better”.

The word “better” is transformed into the word “good” by a lemmatizer but is unchanged by stemming. Even though stemmers can lead to less accurate results, they are easier to build and perform faster than lemmatizers. But lemmatizers are recommended if you're seeking more precise linguistic rules.

Dependency Parsing

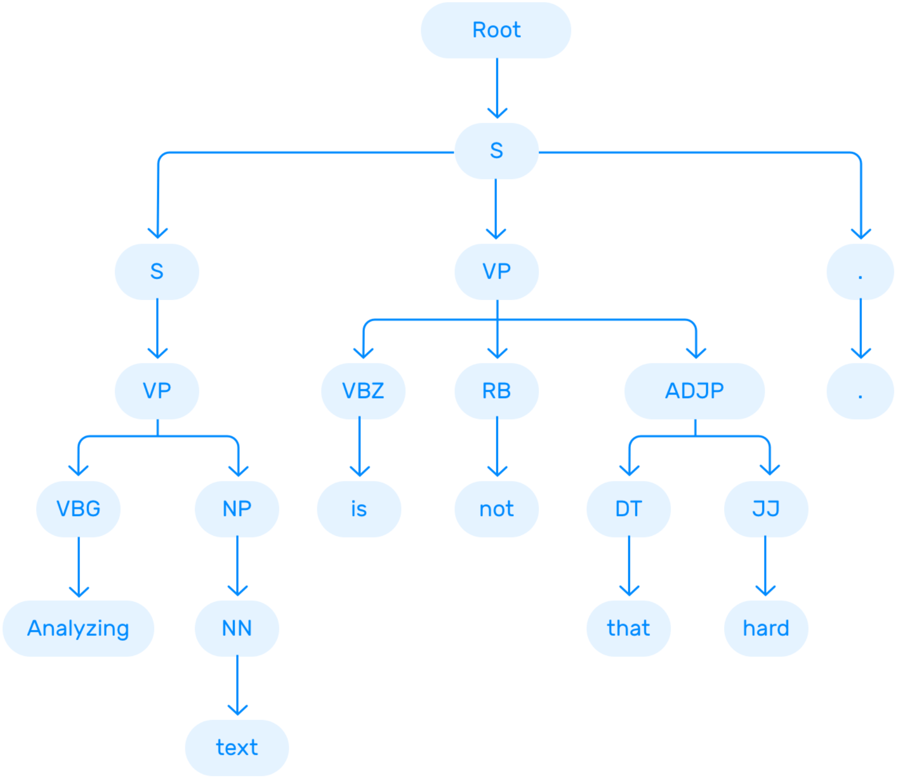
Dependency grammar refers to the way the words in a sentence are connected. A dependency parser, therefore, analyzes how 'head words' are related and modified by other words too understand the syntactic structure of a sentence:



Constituency Parsing

Constituency

Parsing aims to visualize the entire syntactic structure of a sentence by identifying phrase structure grammar. It consists of using abstract terminal and non-terminal nodes associated to words, as shown in this example:



Stopword Removal

Removing stop words is an essential step in NLP text processing. It involves filtering out high-frequency words that add little or no semantic value to a sentence, for example, *which, to, at, for, is,* etc.

You can even customize lists of stopwords to include words that you want to ignore.

Let's say you want to classify customer service tickets based on their topics. In this example: *"Hello, I'm having trouble logging in with my new password"*, it may be useful to remove stop words like *"hello"*, *"I"*, *"am"*, *"with"*, *"my"*, so you're left with the words that help you understand the topic of the ticket: *"trouble"*, *"logging in"*, *"new"*, *"password"*.

2.2.4 Common semantics tasks

Word Sense Disambiguation

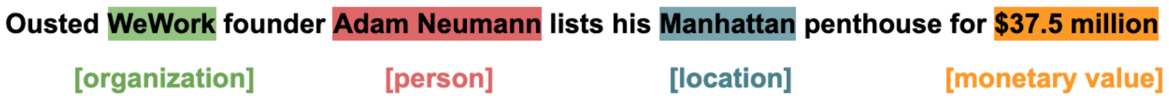
Depending on their context, words can have different meanings. Take the word *"book"*, for example:

- You should read this book; it's a great novel!
- You should book the flights as soon as possible.
- You should close the books by the end of the year.
- You should do everything by the book to avoid potential complications.

There are two main techniques that can be used for word sense disambiguation (WSD): *knowledge-based (or dictionary approach)* or *supervised approach*. The first one tries to infer meaning by observing the dictionary definitions of ambiguous terms within a text, while the latter is based on natural language processing algorithms that learn from training data.

Named entity recognition (NER)

Named entity recognition (NER) – also called entity identification or entity extraction – is a natural language processing (NLP) technique that automatically identifies named entities in a text and classifies them into predefined categories. Entities can be names of people, organizations, locations, times, quantities, monetary values, percentages, and more.



Ousted WeWork founder Adam Neumann lists his Manhattan penthouse for $37.5 million

Figure 15 Example of NER in a sentence

NLP studies the structure and rules of language and creates intelligent systems capable of deriving meaning from text and speech, while machine learning helps machines learn and improve over time. To learn what an entity is, a NER model needs to be able to detect a word or string of words that form an entity (e.g. New York City) and know which entity category it belongs to.

So first, we need to create entity categories, like *Name, Location, Event, Organization,* etc., and feed a NER model relevant training data. Then, by tagging some word and phrase samples with their corresponding entities, you'll eventually teach your NER model how to detect entities themselves.

Text Classification

Text classification is the process of understanding the meaning of the unstructured text and organizing it into predefined categories (tags). One of the most popular text classification tasks is sentiment analysis, which aims to categorize unstructured data by sentiment.

Other classification tasks include intent detection, topic modeling, and language detection. We will dive more into these tasks in the next section.

2.2.5 Diving into the text classification task

Classification is the process of recognizing, understanding, and grouping ideas and objects into preset categories or “sub-populations. This classification process can be manual or automatic. Manual text classification involves a human annotator, who interprets the content of the text and categorizes it accordingly. This method can deliver good results but it’s time-consuming and expensive. Automatic text classification applies machine learning, natural language processing (NLP), and other AI-guided techniques to automatically classify text in a faster, more cost-effective, and more accurate manner.

There are many approaches to automatic text classification, but they all fall under three types of systems:

- Rule-based systems
- Machine Learning-based systems
- Hybrid systems

A Hybrid system involves a mix between Rule-based systems and Machine Learning-based systems. For this reason, we are going to dive into the first two to give an overview of the classification process in NLP.

Rule-based systems

Rule-based approaches classify text into organized groups by using a set of handcrafted linguistic rules. These rules instruct the system to use semantically relevant elements of a text to identify relevant categories based on its content. Each rule consists of an antecedent or pattern and a predicted category.

Say that you want to classify news articles into two groups: *Sports* and *Politics*. First, you’ll need to define two lists of words that characterize each group (e.g., words related to sports such as *football*, *basketball*, *LeBron James*, etc., and words related to politics, such as *Donald Trump*, *Hillary Clinton*, *Putin*, etc.).

Next, when you want to classify a new incoming text, you'll need to count the number of sport-related words that appear in the text and do the same for politics-related words. If the number of sports-related word appearances is greater than the politics-related word count, then the text is classified as Sports and vice versa.

For example, this rule-based system will classify the headline *"When is LeBron James' first game with the Lakers?"* as *Sports* because it counted one sports-related term (LeBron James) and it didn't count any politics-related terms.

Rule-based systems are human comprehensible and can be improved over time. But this approach has some disadvantages. For starters, these systems require deep knowledge of the domain. They are also time-consuming, since generating rules for a complex system can be quite challenging and usually requires a lot of analysis and testing. Rule-based systems are also difficult to maintain and don't scale well given that adding new rules can affect the results of the pre-existing rules.

Machine-learning based systems

Instead of relying on manually crafted rules, machine learning text classification learns to make classifications based on past observations. By using pre-labeled examples as training data, machine learning algorithms can learn the different associations between pieces of text, and that a particular output (i.e., tags) is expected for a particular input (i.e., text). A "tag" is the pre-determined classification or category that any given text could fall into.

The first step towards training a machine learning NLP classifier is feature extraction: a method is used to transform each text into a numerical representation in the form of a vector. One of the most frequently used approaches is a bag of words, where a vector represents the frequency of a word in a predefined dictionary of words.

For example, if we have defined our dictionary to have the following words {This, is, the, not, awesome, bad, basketball}, and we wanted to vectorize the text "This is awesome," we would have the following vector representation of that text: (1, 1, 0, 0, 1, 0, 0).

Then, the machine learning algorithm is fed with training data that consists of pairs of feature sets (vectors for each text example) and tags (e.g. sports, politics) to produce a classification model:

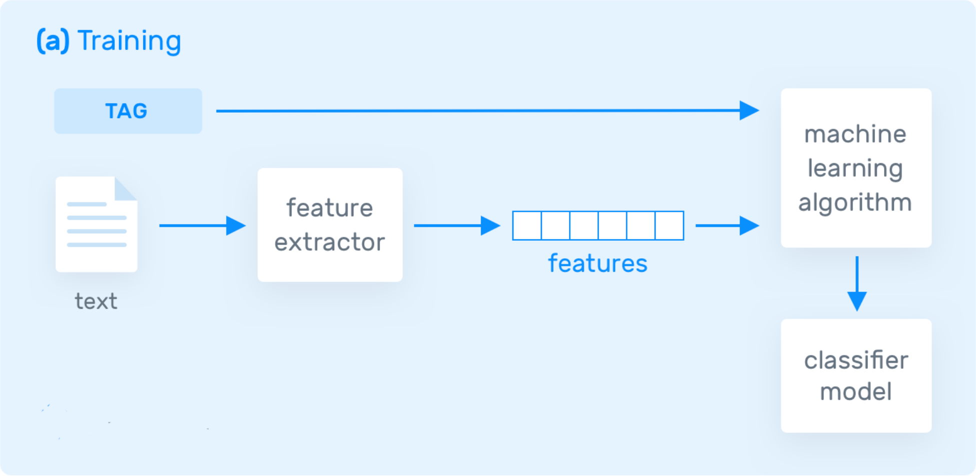


Figure 16 Training phase

Once it’s trained with enough training samples, the machine learning model can begin to make accurate predictions. The same feature extractor is used to transform unseen text to feature sets, which can be fed into the classification model to get predictions on tags (e.g., sports, politics):

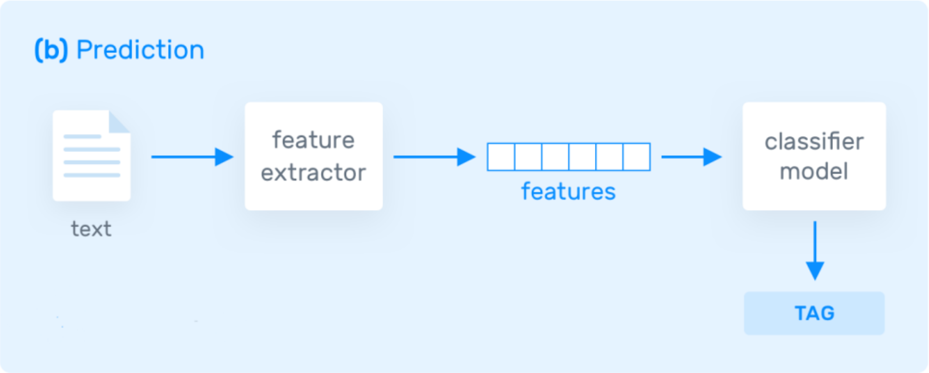


Figure 17 Prediciton phase

Logistic regression

Logistic regression is a calculation used to predict a binary outcome: either something happens, or does not. This can be exhibited as Yes/No, Pass/Fail, Alive/Dead, etc. Independent variables are analyzed to determine the binary outcome with the results falling into one of two categories. The independent variables can be categorical or numeric, but the dependent variable is always categorical. Written like this:

$$P(Y=1|X) \text{ or } P(Y=0|X)$$

It calculates the probability of dependent variable Y, given independent variable X. This can be used to calculate the probability of a word having a positive or negative connotation (0, 1, or on a scale between). Or it can be used to determine the object contained in a photo (tree, flower, grass, etc.), with each object given a probability between 0 and 1.

Naïve Bayes

Naive Bayes is a family of probabilistic algorithms that take advantage of probability theory and Bayes' Theorem to predict the tag of a text (like a piece of news or a customer review). They are probabilistic, which means that they calculate the probability of each tag for a given text, and then output the tag with the highest one. The way they get these probabilities is by using Bayes' Theorem, which describes the probability of a feature, based on prior knowledge of conditions that might be related to that feature. So we're calculating the probability of each tag for a given text, and then outputting the tag with the highest probability.

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

The probability of A, if B is true, is equal to the probability of B, if A is true, times the probability of A being true, divided by the probability of B being true.

This means that any vector that represents a text will have to contain information about the probabilities of the appearance of certain words within the texts of a given category so that the algorithm can compute the likelihood of that text belonging to the category.

Decision Tree

A decision tree is a supervised learning algorithm that is perfect for classification problems, as it's able to order classes on a precise level. It works like a flow chart, separating data points into two similar categories at a time from the "tree trunk" to "branches," to "leaves," where the categories become more finitely similar. This creates categories within categories, allowing for organic classification with limited human supervision.

To continue with the sports example, this is how the decision tree works:

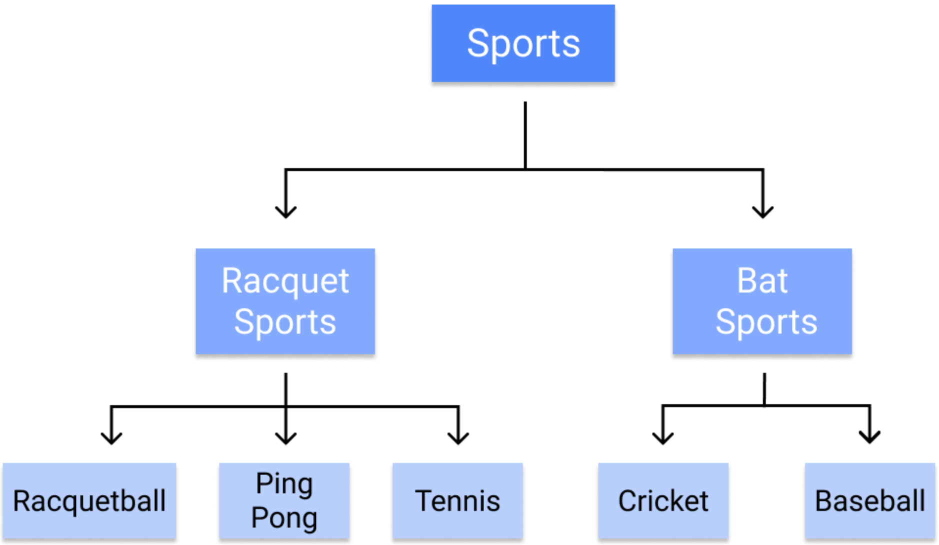


Figure 18 Decision Tree algorith example

3. Evaluation of existing sentiment analyzer tools

3.1 Tool selection

In this chapter, a set of representative tools of sentiment analysis is presented. We have chosen this set of tools due there is a rule-based system of sentiment classifiers as VADER, and a machine learning system of sentiments classification as Detoxify, NLPCore, or Perspective API. On the other hand, Perspective API is a private tool from a big company (Google), and the other systems are open-sourced by a company of developers as Detoxify, on by an academic institution as NLPCore provided by the Stanford University NLP Group. Finally, the implementation use of VADER is powered by PHP, Detoxify is powered by Python, NLPCore is powered by Java. This gives a wide range of underlying technologies and, in the author's opinion, makes this selection representative.

Detoxify

Detoxify is the result of three Kaggle competitions [13] proposed to improve toxicity classifiers. This tool is the compilation of training models to detect hate speech and toxicity over the web that has gained in the last three Kaggle editions. The developers of Detoxify have now founded Unitary [10], a start-up that offers the services of Detoxify as a service.

In concert Detoxify NLP sentiment detector is based on machine-learning algorithms. The architecture of these models is based on Transformers [14], a modern type of architecture of the neuronal network. Detoxify can be used with three pre-trained models one for each competition of Kaggle it has won. Each had a different purpose within the toxicity classifiers context.

[10] https://www.unitary.ai/

- Toxicity comment classification challenge: The first competition aimed to build a generic toxicity classification model that contemplates different kinds of toxicity (insult, threat, sexuality…)
- Unintended Bias in Toxicity Classification: It is a fact that some words confuse since they are often used to harm some collectives (e.g. homosexual, women, or race-related words). When these kinds of words are used in a healthy context they can also be considered toxic by biased language models. The 2nd version of the original competition wanted to improve the unintended bias when classifying toxic messages.
- Multilingual Toxic Comment Classification: The last competition aimed to classify toxicity in a wide range of languages. The 2 previous worked only in English, so this time the objective was to achieve good results with other languages.

Core NLP Stanford

CoreNLP is a tool open-sourced by the NLP Stanford University group. The Natural Language Processing Group at Stanford University is a team of faculty, postdocs, programmers, and students who work together on algorithms that allow computers to process, generate, and understand human languages

CoreNLP is an NLP tool built in Java able to perform the most common tasks in NLP over a sentence. CoreNLP enables users to derive linguistic annotations for text, including token and sentence boundaries, parts of speech, named entities, numeric and time values, dependency and constituency parses, coreference, sentiment, quote attributions, and relations. CoreNLP currently supports 6 languages: Arabic, Chinese, English, French, German, and Spanish.

The centerpiece of CoreNLP is the pipeline. Pipelines take in raw text, run a series of NLP annotators on the text, and produce a final set of annotations.

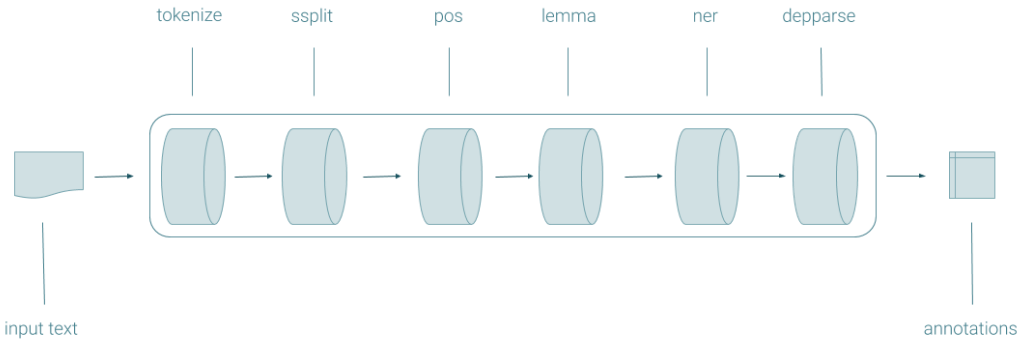
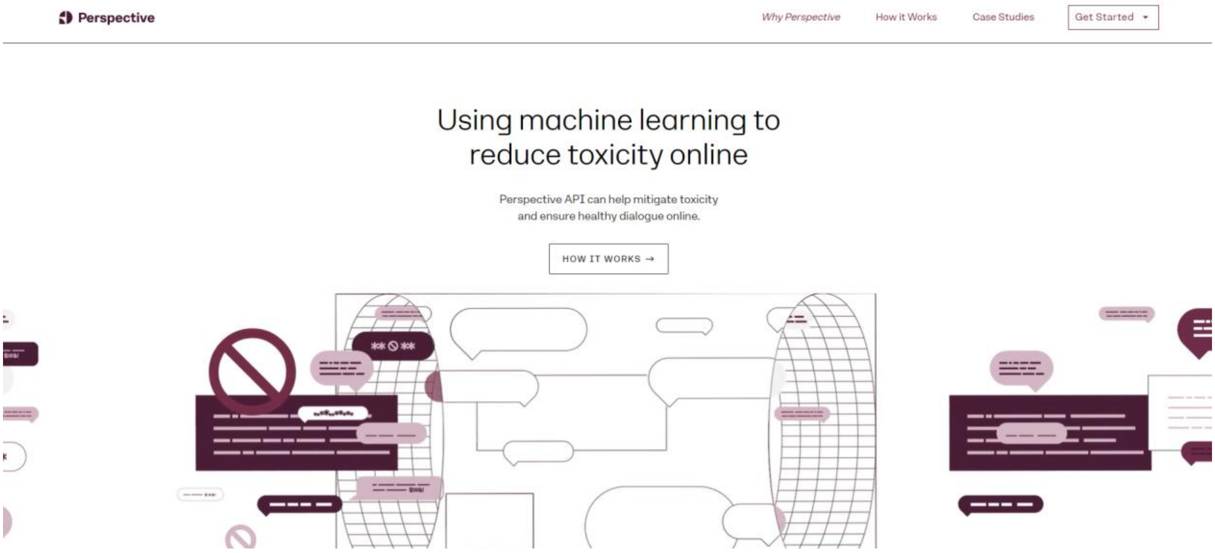


Figure 19 NLPCore pipelines. Source: CoreNLP

CoreNLP could be tested using its online version https://corenlp.run/ where the different tasks (annotators) could be set over a text to analyze. One of these annotators is the sentiments analysis classifier that matches the goals of this work.

Perspective API

Perspective API is a service released by the Google Brain department in 2020, that works as a Software as a Service in terms of analyzing text. Perspective uses machine learning models to identify abusive comments. The models score a phrase based on the perceived impact the text may have in a conversation. Developers and publishers can use this score to give feedback to commenters, help moderators more easily review comments, or help readers filter out “toxic” language.



Using machine learning to reduce toxicity online

Perspective API can help mitigate toxicity and ensure healthy dialogue online.

Figure 20 Perspective API main site.

VADER

Valence Aware Dictionary for sentiment Reasoning, or Vader [8], is an NLP algorithm that blended a sentiment lexicon approach as well as grammatical rules and syntactical conventions for expressing sentiment polarity and intensity. Vader is an open-sourced package within the Natural Language Toolkit (NLTK) and here are the source code and the original publication if you are interested to check them out.

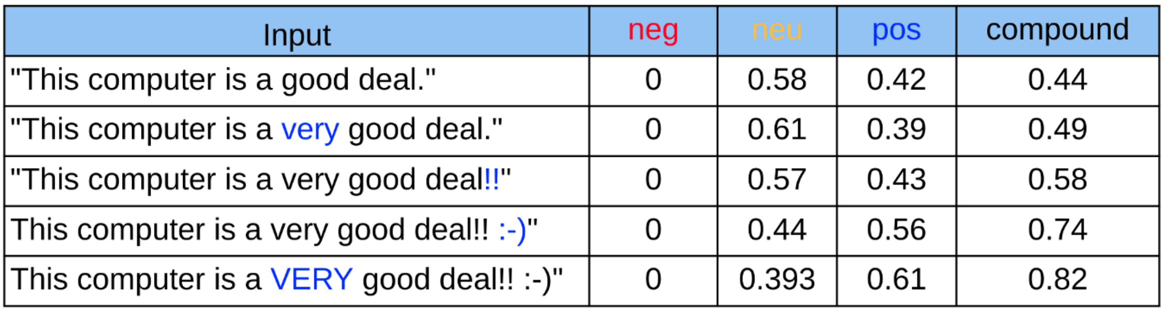
What is Sentiment Lexicon?

The lexicon approach means that this algorithm constructed a dictionary that contains a comprehensive list of sentiment features. This lexical dictionary does not only contain words but also phrases (such as “bad ass” and “the bomb”), emoticons (such as “:-)”), and sentiment-laden acronyms (such as “ROFL” and “WTF”). All the lexical features were rated for the polarity and intensity on a scale from “-4: Extremely Negative” to “+4 Extremely Positive” by 10 independent human raters. The average score is then used as the sentiment indicator for each lexical feature in the dictionary. For example, in Vader, the word “okay” has a positive rating of 0.9, “good” is 1.9, and “great” is 3.1, whereas “horrible” is -2.5, the frowning emoticon “:(“

is -2.2, and “sucks” is -1.5. Vader’s lexicon dictionary contains around 7,500 sentiment features in total and any word not listed in the dictionary will be scored as “0: Neutral”.

Grammatical Rules

Besides the sentiment lexicons, some structures are neutral inherently but can change the polarity of sentiment (such as “not” and “but”) or modify the intensity of the entire sentence (such as “very” and “extremely”). In Vader, the developers incorporated several heuristic rules that handle the cases of punctuation, capitalization, adverbs, and contrastive conjunctions. Below are a few examples of how the degree modifiers boosted the positivity in the compound score of a sentence.



Input	neg	neu	pos	compound
"This computer is a good deal."	0	0.58	0.42	0.44
"This computer is a very good deal."	0	0.61	0.39	0.49
"This computer is a very good deal!!"	0	0.57	0.43	0.58
This computer is a very good deal!! :-)"	0	0.44	0.56	0.74
This computer is a VERY good deal!! :-)"	0	0.393	0.61	0.82

Table 3: Vader answer example

Calculate the Compound Score

To calculate the sentimental score of the entire text, Vader scans the text for known sentimental features, modified the intensity and polarity according to the rules, summed up the scores of features found within the text, and normalized the final score to (-1, 1) using the function:

$$\frac{x}{\sqrt{x^2 + \alpha}}$$

In Vader, alpha is set to be 15 which approximates the maximum expected value of x.

In addition to the compound score of the sentence, Vader also returns the percentage of positive, negative and neutral sentiment features, as shown in the previous example.

3.2 Building the test environment

The issue of using different tools. is that each of these tools needs its infrastructure to be executed. The easiest to run will be the Perspective API of Google as is a service. In this case, we have to get an API Key by performing a registration process in the service, and with this API we can interact with the tool using an API Client software as Postman[11].

This would not be the case with the other tools, as NLPCore is developed over java and works over a Java Virtual Machine, the approach of VADER we are going to use is developed over PHP, and finally, the Detoxify model is built over Phyton.

To simplify the test environment, we are going to use virtualization software. Instead of using a classic virtual machine approach, we are going to use the softest approach. Containers, running over Docker will be the chosen technology that will allow us to have different environments (Java, PHP, Phyton) working isolated but at the same time.

To manage the container infrastructure, we are going to use docker-compose. As we do not need to scale or to build reliability strategies, as is only a test environment, with docker-compose will be enough for our proposals. In resume, we will have each tool deployed to a different container in our localhost, and we will perform remote REST calls to the Perspective API of Google. Then the generated results will be stored locally to be able to evaluate it... In the figure below there is the schema of the built infrastructure.

[11] https://www.postman.com/

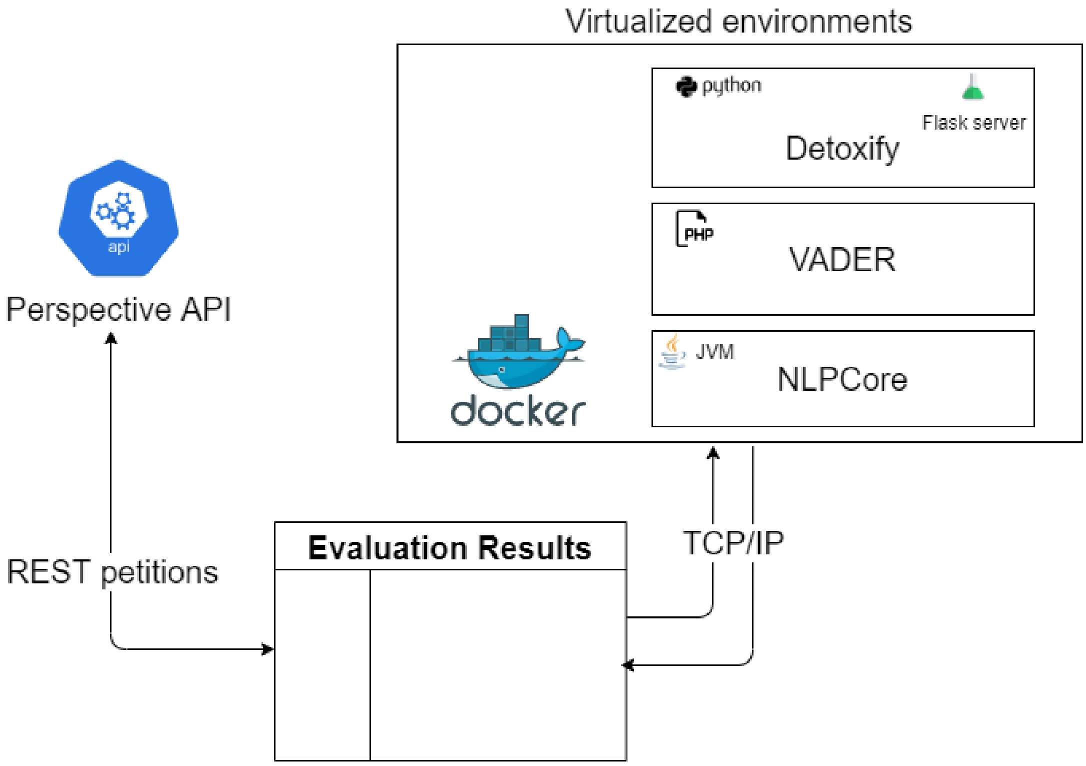


Figure 21 Test infrastructure schema. Source: author

3.3 Data & Answer normalization

Once we have the selection, we are going to see which type of toxicity prediction every tool provides and its limitation of use in the case of private software. Then we will try to propose a normalization of the results to be able to compare them. The first tool to evaluate is Vader powered by PHP. As we see in the following table, this tool returns a set of parameters based on the *positive*, *neutral*, and *negative* issues found in the analyzed text. Then perform an equation presented in the previous section to find the *compound* parameter. A compound parameter is a number between -1 a 1 where -1 represents a full negative text and 1 a full positive text.

Text	positive	neutral	negative	compound
VADER is smart, handsome, and funny.	0.74	0.254	0.0	0.8316
VADER is smart, handsome, and funny!	0.752	0.299	0.0	0.8545
VADER is very smart, handsome, and funny.	0.701	0.246	0.0	0.9227
VADER is VERY SMART, handsome, and FUNNY.	0.754	0.233	0.0	0.9227
VADER is VERY SMART, handsome, and FUNNY!!!	0.767	0.294	0.0	0.9342
VADER is VERY SMART, uber handsome, and FRIGGIN FUNNY!!!	0.706	0.246	0.0	0.9469
VADER is not smart, handsome, nor funny.	0.0	0.354	0.645	-0,74
The plot was good, but the characters are uncompelling and the dialog is not great.	0.094	0.5	0.327	-0,7042
Today SUX!	0.0	0.221	0.779	-0,54
Today only kinda sux! But I'll get by, lol	0.317	0.556	0.127	0.5428

Table 4: Vader answer. Source: author

On the other side, the Perspective API of Google has different types of answers. Perspective's main attribute is *TOXICITY*, defined as "a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion. On the other side, we can find more attributes as *PROFANITY, FLIRTATRION, IDENTITY_*ATTACK, etc. To be able to normalize the answer between the other tools this work only uses TOXICITY as a parameter to set in the request. With this assumption, the answer to this tool is like the following table. Th Summary Value and Value are the same because, for simplicity, we are not performing more evaluation with extra parameters. The score value is a parameter of probability between 0 and 1, where 1 is full toxicity and 0 is no toxicity probability.

Text	Value	type	Summary Value	type
Everybody want's the piece of pie, Owners, Contractors, and the Senate.	0.07	Probability	0.07	Probability
Isn't the person dead?	0.24	Probability	0.24	Probability
It's a black mark on the previous administrations who allowed massive illegal immigration to go on for far too long. Put the blame where it should be.	0.33	Probability	0.33	Probability
I can't believe this country was so stupid.. -totally dumbstruck- you get what you get and don't throw a fit!	0.87	Probability	0.87	Probability

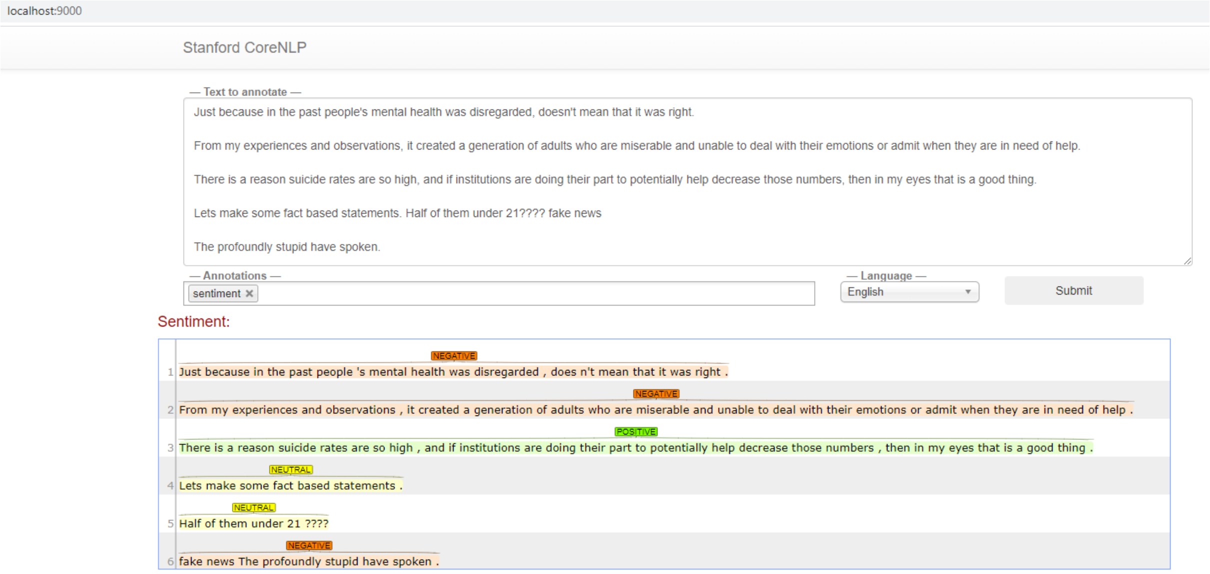
Table 5: Perspective API answer. Source author

In terms of an answer, Detoxify has multiple types of answers. This tool owns 3 different models, the base model, the unbiased model, and the multilingual mode. The base and the unbiased one have all the parameters shown in the table, instead of the multilingual only have toxicity. The tools only return a single parameter between 0 and 1, in similar terms to Perspective API.

Text	Toxicity	Insult	Obscene	Identity attack	Threat
It's a black mark on the previous administrations who allowed massive illegal immigration to go on for far too long. Put the blame where it should be.	0.013	0.0018	0	0.07	0.07
I totally understand what murder is. This is the killing of a person with the sanction and approval of the State. It's an execution.	0.05	0.0	0	0	0.03
Somebody needs to dig up the Peterson's back yard! What other evil deeds did Monster Mom hide to protect her killer sons? Lock her up too.	0.73	0.32	0	0	0.2
Right on the money Gary Crum. And if they hide behind the anonymity of masks and the mob, they're cowards to boot.	0.94	0.91	0	0	0

Table 6: Detoxify getted answer. Source author.

At last, we are going to analyze NLPCore from Stanford. This tool returns a different answer based on some categories of toxicity. As the answer is difficult to compare with the other tools as the format is quite different. This tool only provides a “negative”, “neutral” or “positive” tag. As is designed as a research tool instead of a final product it is difficult to adapt to our specific case. In the following figure, we can see the answers given by the tool we build for this work. Due to the difference in terms of answers we will not include this tool in our experiment.



localhost:9000

Stanford CoreNLP

— Text to annotate —

Just because in the past people's mental health was disregarded, doesn't mean that it was right.

From my experiences and observations, it created a generation of adults who are miserable and unable to deal with their emotions or admit when they are in need of help.

There is a reason suicide rates are so high, and if institutions are doing their part to potentially help decrease those numbers, then in my eyes that is a good thing.

Lets make some fact based statements. Half of them under 21???? fake news

The profoundly stupid have spoken.

— Annotations — sentiment ×

— Language — English

Submit

Sentiment:

1 NEGATIVE Just because in the past people 's mental health was disregarded , does n't mean that it was right .

2 NEGATIVE From my experiences and observations , it created a generation of adults who are miserable and unable to deal with their emotions or admit when they are in need of help .

3 POSITIVE There is a reason suicide rates are so high , and if institutions are doing their part to potentially help decrease those numbers , then in my eyes that is a good thing .

4 NEUTRAL Lets make some fact based statements .

5 NEUTRAL Half of them under 21 ????

6 NEGATIVE fake news The profoundly stupid have spoken .

Figure 22 Standford NLPCore answer. Source author.

Once analyzed the answer of the different tools we propose using the ***Toxicity* score** of Detoxify, as can be used also in the multilingual model, the ***Summary Score*** of Perspective API, and the ***Compound Value*** of the VADER tool in terms of comparison.

3.4 Tool comparison

At the end of 2017, the Civil Comments[12] platform shut down and chose to make their ~2m public comments from their platform available in a lasting open archive so that researchers could understand and improve civility in online conversations for years to come. This opportunity was taken by Jigsaw who sponsored the effort to build it and nowadays this dataset is known as unintended bias in toxicity classification [14]. We are going to use this dataset to compare the tools between them. To compare the tools, we are going to process the same excerpt of the datasets to every tool and then extract conclusions. In Figure 22, there is an excerpt of the mentioned dataset.

id	text
7097320	*[Integrity means that you pay your debts.] Does this apply to President Trump too?*
7097321	This is malfeasance by the Administrator and the Board. They are wasting our money!
7097322	@Rmiller101 - Spoken like a true elitist. But look out bud. The re-awakening in Europe, Brexit and n...
7097323	Paul: Thank you for your kind words. I do, indeed, have strong beliefs and don't hide them. They a...
7097324	Sorry you missed high school. Eisenhower sent troops to Vietnam after the French withdrew in 1954 an...
7097325	Let’s see if I understand this; Berkowitz announces a $14M surplus then he rails against Proposition...

Figure 23 Excerpt of the unbiased dataset of Jigsaw

At the first point, as we discussed in the background section, Detoxify can run over different pre-trained models. The Original model and the “Un-biased” model could be compared to choose which fits betters with a random set of test data extracted from the dataset. In the Figure below we see that both models perform similarly. The data shown is extracted by analyzing

[12] https://medium.com/@aja_15265/saying-goodbye-to-civil-comments-41859d3a2b1d

100 texts randomly extracted from the dataset. We see that in general terms follow the same criteria but the unbiased detects some toxicity in some text like 50, 65, 69,

and do not detect toxicity in others like 72, and 11.

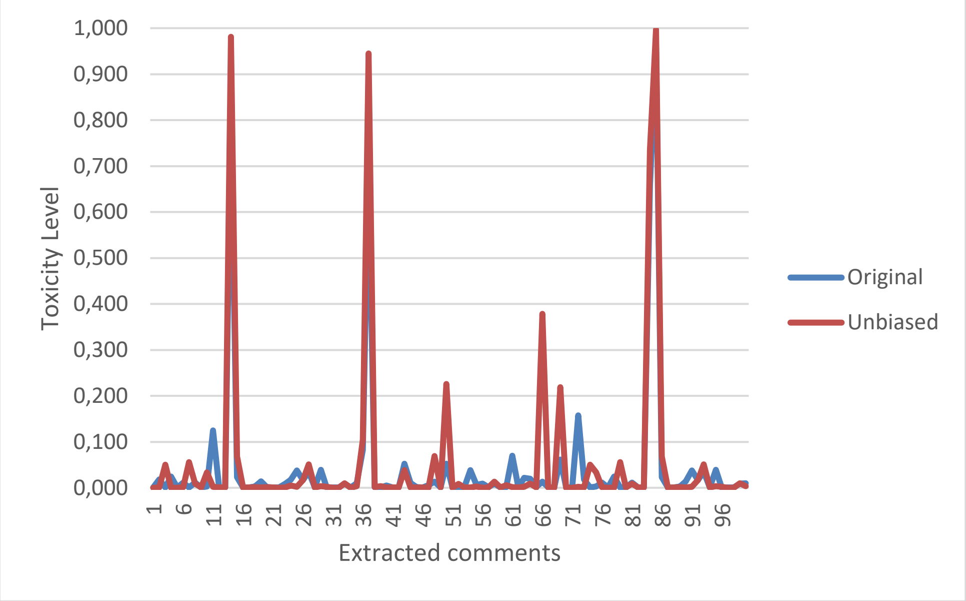


Figure 24 Detoxify biased and unbiased model comparison

id	Text	Original	Unbiased
11	Bummer man! Greg used to shoo me outta his store when I was in high skrewel. Later, we had some good discussions of great rock bands. GREG! Get a job. Your store is closed! I'll give you $20 for the Van Halen and Led Zep t-shirts I wanted.	0.12	0.002
50	this is *&^%ing outrageous. The prosecutor should be immediately removed from his job.	0.052	0.225
66	Between racist Eastman and 'Slappy' Wilson the Wasilla delegation is an embarassment to the valley and the State. Both should resign. This is Trump's America, open racism in the Republican party and physical attacks on reporters. Cue the usual racist adn commenters telling us how Eastman actually has a valid point with no evidence beyond an anecdote from the time they did a stint in the bush in 3...2...1....	0.013	0.3786
69	'Work together'? Dream on. The liberals are already busy creating the next scandal/crime to accuse President Trump of since they lose the policy	0.06	0.21

	debate every time. They are sniveling statist traitors and they should be ground into the dirt as they will never ever let up in their qwest to destroy this country by any means available. Of that you can be sure.		
72	This is malfeasance by the Administrator and the Board. They are wasting our money!	0.15	0.002

Table 7: Highlight of the relevant divergences between models. Source: Author.

The table above shows the main detected divergence points between the biased and the original model. The difference between these models relies on the dataset that has been used to train them. We can see that despite the similarities in the overall samples, there are some points where the difference is important. In 66 the model detects a higher level of toxicity due is detecting “Identity hate” in the sentence. Reading the sentence is not clear that these sentences represent identity hate, but its clear contains some warning words like “Easters” and “racist” or “violence”. On the other hand, in 69, that unbiased model detects a higher level of toxicity, due to the hard criticism of the comment. In this sample, we can affirm the answer is clearly better. Finally, in 11 and 72, the unbiased model detects less toxicity in the comments. Both comments are neutral but use informal stress. In this case, we can affirm the unbiased model is better.

Following the evaluation, we can compare the result of VADER with the unbiased model of Detoxify. In the Figure above we can see that the results are clearly different, and the performance of rule-based systems like Vader is much worse than the based in neuronal networks / Transformers as Detoxify.

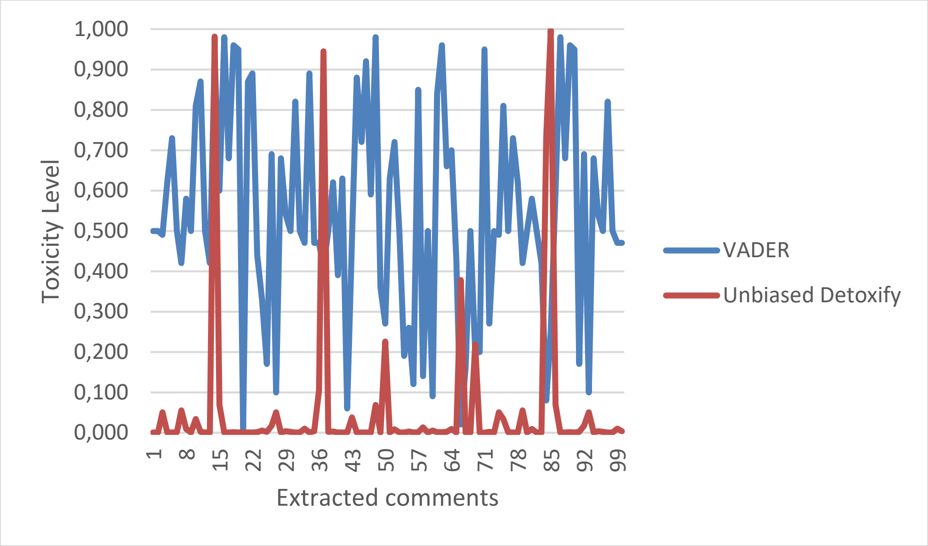


Figure 25 Vader vs the Unbiased Detoxify Model

In the following Figure, we can see a comparison between Perspective API and the Unbiased Detoxify model. Both systems rely upon the same architecture, they rely on Transformers, but Perspective API is created by the Google Brain department, and served as a Software as a service.

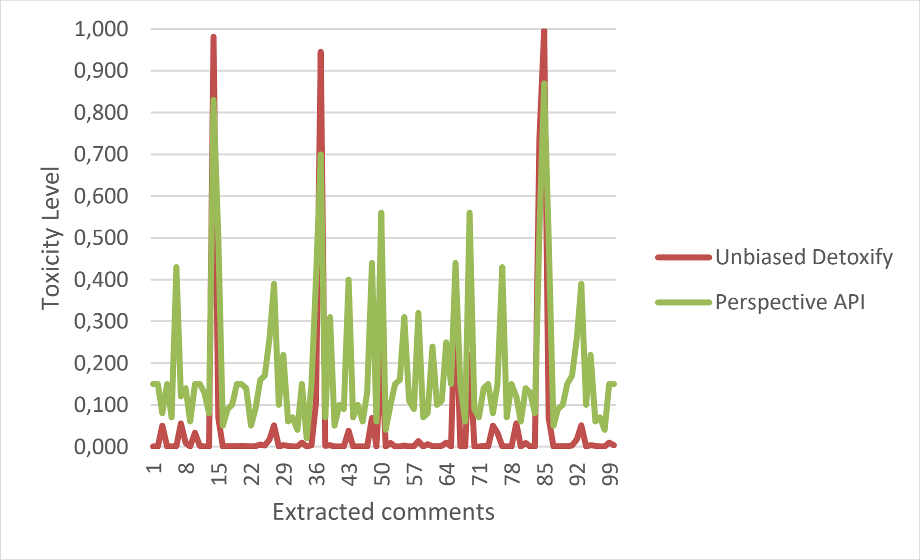


Figure 26 Detoxify vs Perspective comparison

Looking at the Figure we can see that the sensibility is slightly different, but the shape of the graph is very similar. This means, that both tools can similarly detect the toxicity. Two tools trained with two different datasets and developed by two different teams can serve a similar answer taking a random sample of comments of the net. This shows good results and could be the starting point to, technically, build a general model to detect toxicity.

But some points need to be remarked. The data we use is from the dataset from Jigsaw. This is one of the biggest datasets that are open access. Perspective API and Detoxify could be used partially the same data to train their models. To evaluate correctly these two tools, we will need to re-train the models with different data, and then re-perform the experiment we have done here with more data. This conclusion can be set as future work.

4. Automatic moderation solution

4.1 Solutions design

Once evaluated the tools with a specific dataset, the goal of this section is to design a complete solution, ready to be used by the developer's community, that allows us to evaluate the selected tools in real environments. The solution needs to provide us an interface to extract conclusions to present the debate about the actual viability of the actual toxicity detectors but also needs to help content moderation to moderate and manage the content.

As a conclusion extracted from the previous section, we will develop a connector able to work with the set of analyzed tools but advertising to the user the limits of every solution. In terms of performance, VADER does not provide good results in comparison with the other services but is the only one developed in PHP and does not need extra infrastructure due Drupal is also powered by PHP. For this reason, we include it also.

The initial version of the plugin will react to every comment creation/update and will analyze the text of the comment providing to the content manager information about the result of the analysis. The following figure shows the process the solution follows.

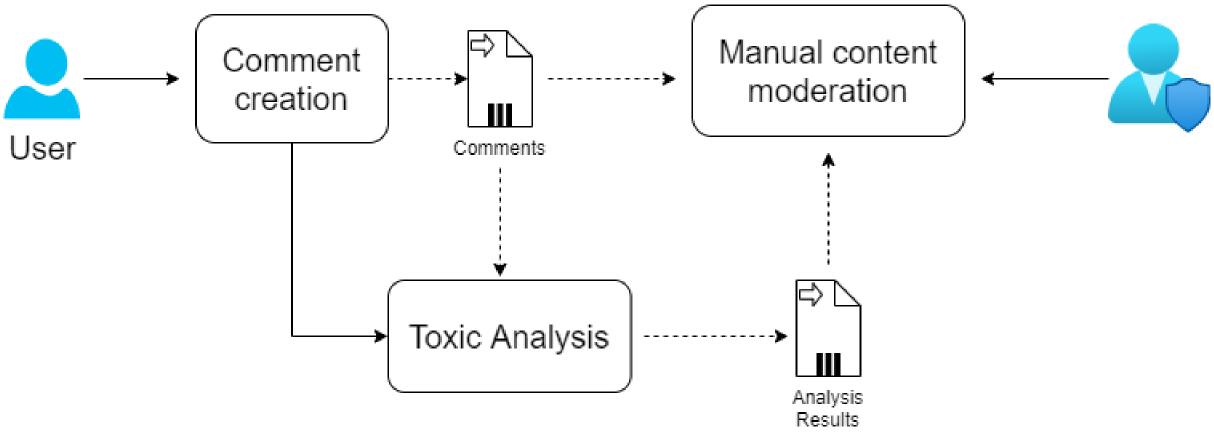
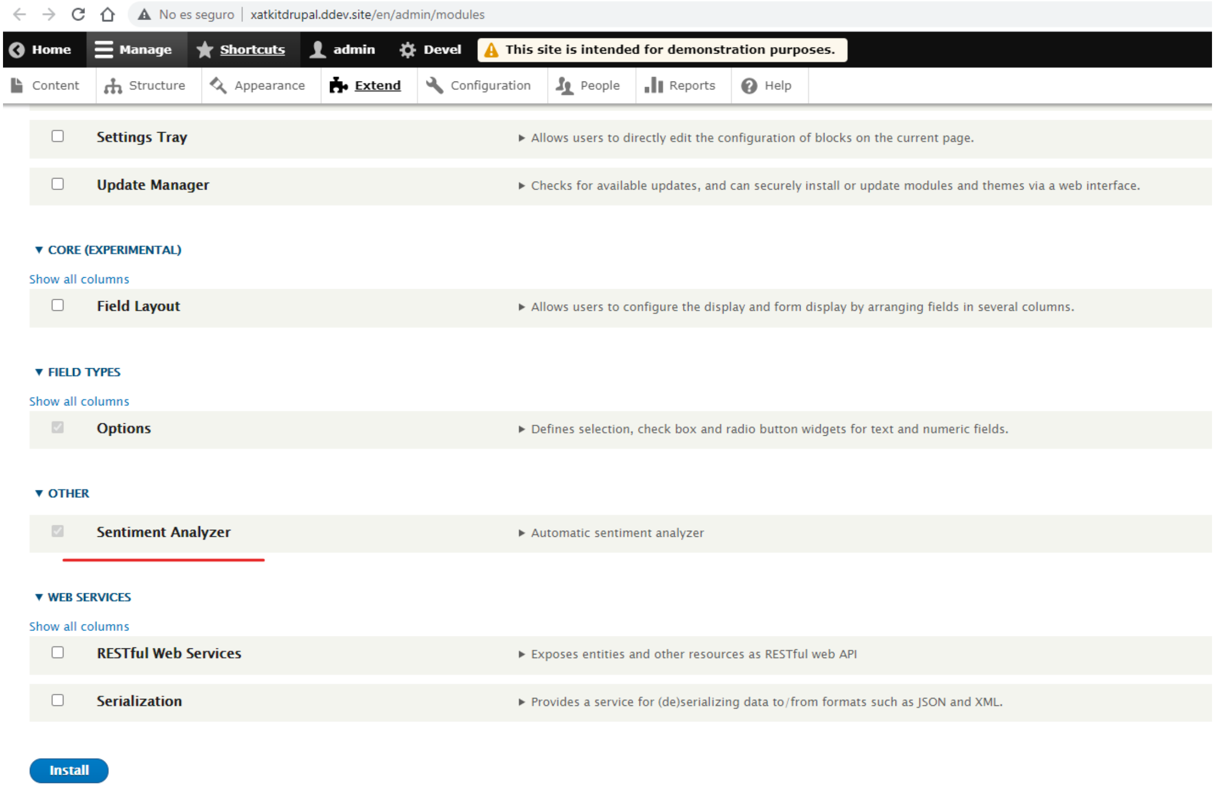


Figure 27 Automatic moderation solution flow

4.2 Solution Development

Drupal and the most used CMS are modular. It means, that each functionality can be added without affecting the existing functionalities. This is the case with our solution. This solution can be installed on any site, without affecting any entity or feature of the site. In our case, we have developed a plugin following the guidelines of the Drupal Community to be able to share it with the community later. [13] In the Figure below we can see how our solution is installable as a normal plugin over the Drupal admin interface.


No es seguro | xatkitdrupal.ddev.site/en/admin/modules
Home
Manage
Shortcuts
admin
Devel
This site is intended for demonstration purposes.
Content
Structure
Appearance
Extend
Configuration
People
Reports
Help

□	Settings Tray	▸ Allows users to directly edit the configuration of blocks on the current page.
□	Update Manager	▸ Checks for available updates, and can securely install or update modules and themes via a web interface.

▾ CORE (EXPERIMENTAL)

Show all columns

□	Field Layout	▸ Allows users to configure the display and form display by arranging fields in several columns.

▾ FIELD TYPES

Show all columns

☑	Options	▸ Defines selection, check box and radio button widgets for text and numeric fields.

▾ OTHER

☑	Sentiment Analyzer	▸ Automatic sentiment analyzer

▾ WEB SERVICES

Show all columns

□	RESTful Web Services	▸ Exposes entities and other resources as RESTful web API
□	Serialization	▸ Provides a service for (de)serializing data to/from formats such as JSON and XML.

Figure 28 Excerpt of the Drupal installation interface

On the other hand, we have developed our solution over PHP following the OOP paradigm, and we have organized our project using the classes shown in the Figure below. The plugin works

[13] https://groups.drupal.org/contrib-development-best-practices

using the Drupal event system. When a comment is created or updates, an event is fired. Using our central class called AnalyzerManager we catch this event to perform the analysis. This class then gets the text to analyze and performs a class to the helper's class called DetoxifyConnector, PerspectiveConnector, and directly to the PHP implementation of VADER. Finally, once the class has received the answers from the different services it saves the comment another time with the answers.

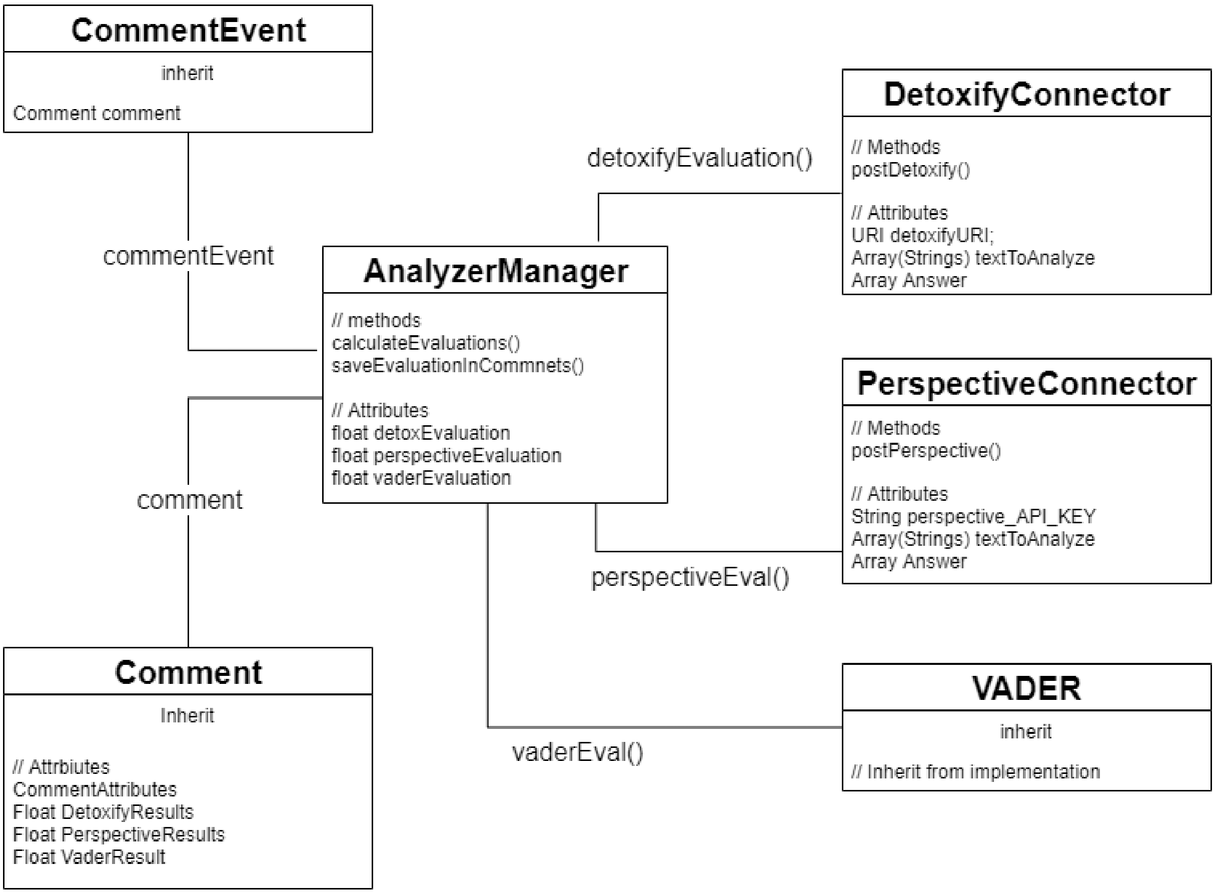
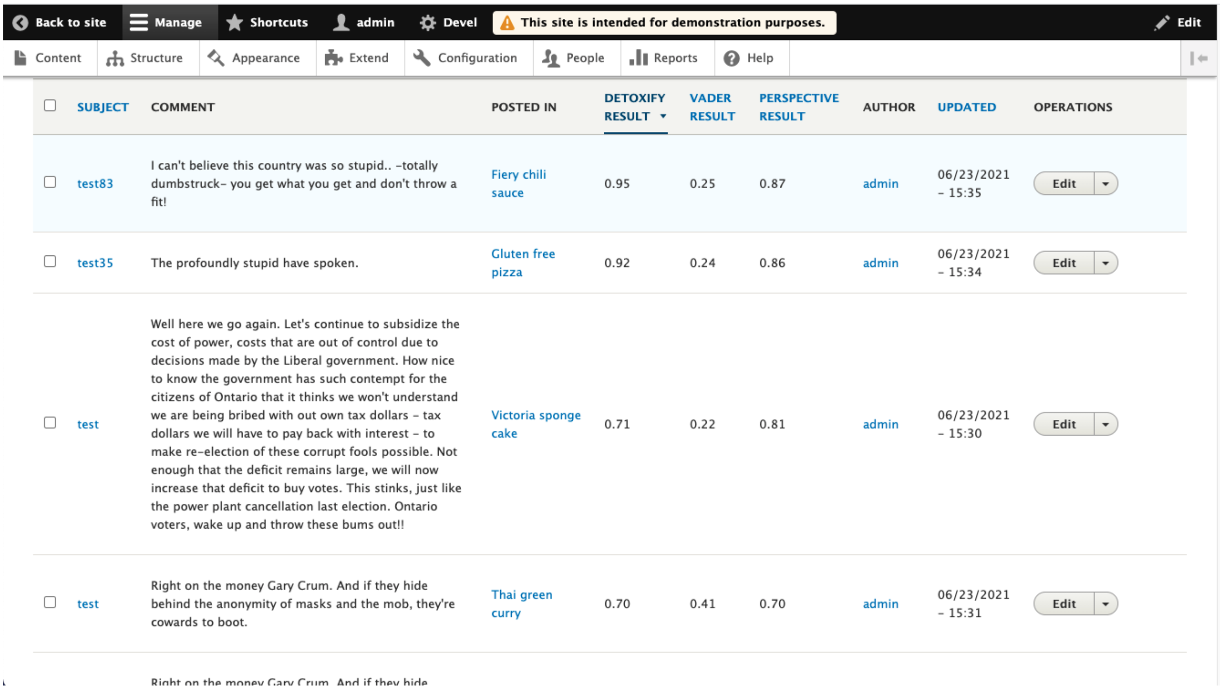


Figure 29 Entity-class diagram of the solution

As we have explained and the start of the evaluation section, Detoxify and Perspective need extra infrastructure instead of a PHP runtime. In our case, we have followed the schema we build for the test using docker and virtualization of the instances needed to perform these services. On the other side, the PHP implementation of VADER does not need this extra infrastructure, and this has been one of the main reasons why we have implemented it instead of its bad performance.

At this point is interesting to remark that future work could be to implement the same models in PHP. This is hard work because there is no mature framework developed to train neuronal networks over PHP. On the other side, building extra infrastructure is every time easier due to the tools like Docker, or Kubernetes that enable to build of complex infrastructure in a few steps.

In the next Figure, we can see the result of the prototype working over a real site. In this Figure, we see the comments moderator panel inherent to the standard installation of Drupal. This panel has been modified using the values obtained by the plugin and now we can evaluate the comments of the site using the implemented tools. Despite this is not useful for production sites, in terms of academic evaluation is interesting to have multiple tools in the same view.



□	SUBJECT	COMMENT	POSTED IN	DETOXIFY RESULT ▾	VADER RESULT	PERSPECTIVE RESULT	AUTHOR	UPDATED	OPERATIONS
□	test83	I can't believe this country was so stupid.. -totally dumbstruck- you get what you get and don't throw a fit!	Fiery chili sauce	0.95	0.25	0.87	admin	06/23/2021 - 15:35	Edit ▾
□	test35	The profoundly stupid have spoken.	Gluten free pizza	0.92	0.24	0.86	admin	06/23/2021 - 15:34	Edit ▾
□	test	Well here we go again. Let's continue to subsidize the cost of power, costs that are out of control due to decisions made by the Liberal government. How nice to know the government has such contempt for the citizens of Ontario that it thinks we won't understand we are being bribed with out own tax dollars - tax dollars we will have to pay back with interest - to make re-election of these corrupt fools possible. Not enough that the deficit remains large, we will now increase that deficit to buy votes. This stinks, just like the power plant cancellation last election. Ontario voters, wake up and throw these bums out!!	Victoria sponge cake	0.71	0.22	0.81	admin	06/23/2021 - 15:30	Edit ▾
□	test	Right on the money Gary Crum. And if they hide behind the anonymity of masks and the mob, they're cowards to boot.	Thai green curry	0.70	0.41	0.70	admin	06/23/2021 - 15:31	Edit ▾
		Right on the money Gary Crum. And if they hide							

Figure 30 Result of the developed plugin in a real site

Finally, all the development process is open-sourced and can be seen, forked or reused through the following public repository: https://github.com/JoanGi/Toxicity-Detector-Drupal

4.3 Solution evaluation – Interview design

To evaluate the solution, we are going to design some interviews and send them to the community to evaluate some factors about our tool, and also to get some insights about the perception of the developer's community about the use of automatic tools in sentiment analysis and toxicity detection.

In the next Figure, we show a model of the interview sent to the developers. This interview asks about the previous experience of these developers with NLP tools. This question is aimed to detect the differences between the developers that already are using these tools in other ways, and the developers with no experience. After then we ask about the plans of adopting tools like our tool, to evaluate the necessity of the market inside the CMS domain. Then we ask about the installation process, the user interface, and the values provided by the tool and its utility.

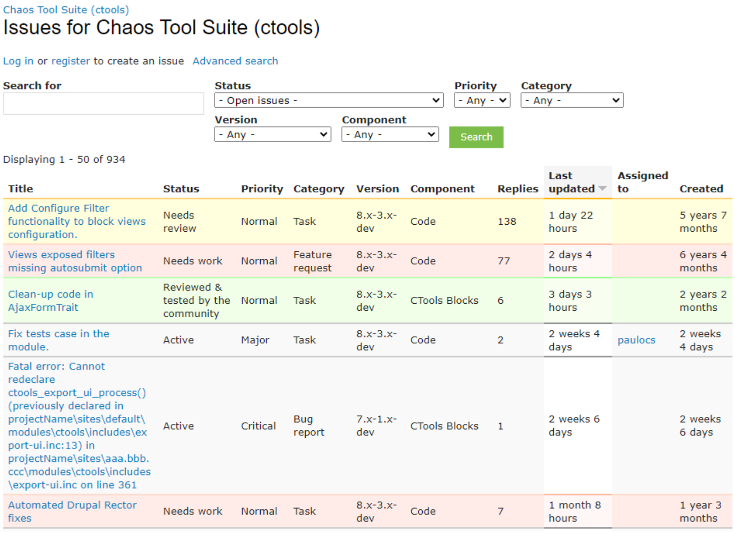
Id	Question	Name/role	Answer
1	What are your previous experiences with NLP tools?	-	-
2	What are your previous experiences with sentiment analysis and toxicity detection?	-	-
3	Where you planning to adopt a tool like this?	-	-
4	Which is your opinion about the wide use of sentiment analysis tools?	-	-
5	What is your opinion on specific uses of sentiment analysis tools?	-	-
6	Which is your opinion about the tool installation process?	-	-
7	Which is your opinion about installing extra infrastructure?	-	-
8	The values of the analysis are useful for your case?	-	-
9	Which will you improve of the UI in the moderation panel?	-	-
10	Which will you improve of the provided values?	-	-
11	After the installation, you will use it in your live sites?	-	-
12	You will be agreed on share, anonymously, the comments to build a bigger public dataset to train new models?	-	

Table 8: Survey model. Source author.

4.4 Deployment to the community

Once the plug-in is developed, we can share the plugin with the Drupal community. The Drupal community is organized around www.drupal.org and Drupal Association[14]. This community created in 2008 has evolved and has welcomed more than 8000 individual contributors and over 1.100 corporate contributors to the code and community.[15]

Besides, the community has built processes to ensure a good quality of the contributions. In this work, we have experienced the approval process to share a full-featured plugin. In this process, a set of reviewers review the solutions provided by the contributor and evaluate them to allow its publication. Once the solution is approved then can be published and can be used as an official plugin of the Drupal community.



Chaos Tool Suite (ctools)

Issues for Chaos Tool Suite (ctools)

Log in or register to create an issue Advanced search

Search for

Status: - Open issues -

Priority: - Any -

Category: - Any -

Version: - Any -

Component: - Any -

Displaying 1 - 50 of 934

Title	Status	Priority	Category	Version	Component	Replies	Last updated	Assigned to	Created
Add Configure Filter functionality to block views configuration.	Needs review	Normal	Task	8.x-3.x-dev	Code	138	1 day 22 hours		5 years 7 months
Views exposed filters missing autosubmit option	Needs work	Normal	Feature request	8.x-3.x-dev	Code	77	2 days 4 hours		6 years 4 months
Clean-up code in AjaxFormTrait	Reviewed & tested by the community	Normal	Task	8.x-3.x-dev	CTools Blocks	6	3 days 3 hours		2 years 2 months
Fix tests case in the module.	Active	Major	Task	8.x-3.x-dev	Code	2	2 weeks 4 days	paulocs	2 weeks 4 days
Fatal error: Cannot redeclare ctools_export_ui_process() (previously declared in projectName\sites\default\modules\ctools\includes\export-ui.inc:13) in projectName\sites\aaa.bbb.ccc\modules\ctools\includes\export-ui.inc on line 361	Active	Critical	Bug report	7.x-1.x-dev	CTools Blocks	1	2 weeks 6 days		2 weeks 6 days
Automated Drupal Rector fixes	Needs work	Normal	Task	8.x-3.x-dev	Code	7	1 month 8 hours		1 year 3 months

Figure 31 Drupal's approval process

For time constrain reasons, the proposed plugin has not yet passed the approval process and as this is not the main goal of this work, it remains as future work.

[14] https://www.drupal.org/association

[15] https://www.acquia.com/landing/drupal-contributors

5. Fairness in NLP Sentiment Analysis

As we have seen in this work there is a real issue concerning sentiment analysis using NLP. The classification models tend to present biases as we have seen, for example, in the first version of Perspective API of Google in 2017. These problems are not easy to solve and in many cases the approaches to solve it can be applied in many stages of the machine-learning development cycle.

But besides facing the technical problems Google has reacted creating a new research group called People + AI Research[16]. This group is composed not only of engineers but also is composed by designers and researchers from social fields. This group composition provides some insight into the nature of the problem. A problem with a complex nature, with multiple points of view, and closely related to the main culture and society around it.

In a first attempt to face this problem, a research field in ethics, fairness and trustworthy AI has emerged in the last two/three years around The Fairness, Accountability, and Transparency in Machine Learning (FAT/ML) community[17]. Our discussion around biases in the NLP classification model seems to be closer to fairness proposes, so in the following subsection, we analyze the definition of fairness present in literature and the proposed methods to ensure fairness in NLP.

5.1 Fairness definitions

Following the work of [16], we can define some shared key considerations for the responsible use of ML/AI algorithms. These concepts are Equity, Representativeness, Explainability, Auditability, and Accountability. **Equity** relates to ensuring that any group is neglected during the evaluation of the algorithm. For example, a woman cannot be detected as more toxic than men in sentiment classification.

Representativeness refers to the way we train the models. If our model is present of white people in case of image models, or our model is full of “American English expressions”, not

[16] https://pair.withgoogle.com/

[17] https://www.fatml.org/

white people, or not American English speakers should notice that the algorithm does not work well for them.

Explainability refers to an architectural problem of ML. Where ML artifacts are seen as a black box with no way to explain how the algorithm has reached a concrete decision. The ability to build models able to be self-explanatory is the main point of this issue and has opened a specific research field in Explainable Artificial Intelligence (XAI). [17]

Auditability refers to the ability to audit the system, for example, for detecting misleading classifications. In our toxicity detectors case, we will be able to audit which sentences are classified in a wrong way to ensure the algorithm's behavior. Finally, **Accountability** are the mechanisms in place to ensure that someone will be responsible for responding to feedback and redressing harms,

Since this concept is an important base there is also important to point, that a common agreement inside the FAT/ML community is the concept of "**having a human in the loop**". This final statement means that the technology can help in the analysis of the algorithms and the bias detection, but humans will ultimately be involved and be part of the decision.

5.2 Mitigating Bias in Data Sets

Examples of bias in data sets include under-sampling for racial, cultural, and gender diversity in image recognition, such as categorizing wedding photos only when the bride is wearing clothes of a specific color following cultural norms. The issue of image datasets underrepresenting certain ethnicities is also known in facial recognition, where classification accuracy suffers when images of underrepresented minority individuals are analyzed. In a third example, voice recognition systems are well known to perform more poorly for non-native English speakers than native speakers, which results in incorrect answers to questions posed to popular voice-based assistant systems. When bias arises in a data set, methods for addressing this include addressing the sampling of the data, cleaning the data and labels, or adding, removing, diversifying, or redistributing features. A resume of the methods presents in the literature is presented in Table 1.

Approach	Explanations
Data augmentation [18]	Refers to a family of techniques that increase the size and diversity of the training data without actually collecting more data
Feature-level reweighting [19]	Describes a family of approaches in which features are assigned weights (multiplied by scalar values) to make the data more representative
Resampling through randomization of the minority class [20]	Boost the number of elements of the minority class by sampling more of that minority class through random sampling with replacement
Adversarial learning [21]	In this approach, there are two machine learners – one predicting the output, and the other predicting the protected attribute —to converge on a model that predicts the correct outcome independent of the protected attribute. Adversarial models have been popular in image classification

Table 9: Methods to mitigate Bias in Data-Sets

5.3 Applying Fairness during the ML model development

During the ML model development, the predominant focus is the statistical perspective on fairness. To avoid diving into technical details we will focus only on the high-level debate. So, the statistical focus is dependent on the approach we take in a concrete situation. We can divide the approaches into two big subfields. These are Fairness through unawareness and Fairness through Awareness.

The first one is not recommended in ML cases. Fairness through unawareness means that we have to detect with parts of the data (features) are generating the bias and remove it from the model. For example, remove the gender information from the model to avoid gender bias. This approach only works if the data are highly uncorrelated. This, which can be perfectly possible is a not usual case, and a very unusual case in a social context. For instance, if we detect a gender bias in a classifier to grant access to a loan, we can eliminate the gender information in our model to avoid it. But the fact that women in our society have another related bias, as lower salaries, more time without work (for maternal leaves), etc. the algorithm will reproduce the bias in the same way.

On the other hand, fairness through awareness involved the "protected group" into the algorithm by applying different strategies. There are several strategies and is not the goal of this work to present a deep study. In contrast, we will discuss some of the main in a high level to open a philosophical debate around them.

These strategies can be applied by boosting some parameters or data in the dataset to achieve the correct model behavior. The first one is demographic parity. This means, correcting biases boosting protected groups (as a woman), to achieve the demographic parity. For example, if a woman represents the 30% of the users who want a loan, then the 30% of loans should go to women. This strategy presents several problems, but in terms of law could be a hard requirement. As many countries have a law that says, "every person is equal in front of the law", the demographic parity seems to fit well with this definition. But present several problems. For example, if a bank is applying this to give loans in a country. People from rural zones will get the loans easier than people from cities, but in terms of returning the loan, in a long-term vision, people from cities will have lower rates of failing than people from rural zones. So, banks will not trust rural petitions anymore.

On the other hand, the equal opportunities approach states to force the same frequency on selection in the general group and the protected group. So, if a woman gets only the 30% of the loans, that the algorithm will force the woman to get the 50% of the loans. On the other hand, the odds opportunities approach states to force the same frequency on selection and also the same frequency on false-negative selections. The false-negative could be difficult to calculate and often adding the odds opportunity approach impacts the model accuracy.

There are several more statistical approaches and for further reading, there is a study of Massachusetts Institute of Technology (MIT) [17] referencing awareness approaches such as an equalized opportunities [18], equalized odds [19], and counterfactual fairness [20].

6. Conclusions and future work

In this work, we have achieved three main objectives around the limits and potentials of using ML sentiment analysis. The first one has been to select and evaluate a set of tools against the same dataset, a dataset was built with the specific purpose of detecting biases. We have detected that the rule-based system behaves very differently from the neuronal network architectures. The solutions based in neuronal network seems to behave similarly, bringing confidence in this type of architecture. On the other hand, we have tested compared the behavior between the model to detect misleading classification in a random selection of the dataset. Since there is a notorious difference, the unbiased Detoxify models and the Perspective API of Google seem to avoid some of the errors of the original neuronal network models as the original Detoxify. In conclusion, we can state that NLP classifiers on sentiment analysis based in neuronal network seems to have a robust behavior but can tend to present biases due to the limitation of data and the lack of common dataset representation and the complexity inherent to the bias mitigation problem.

The second objective has been to create a prototype of a plug-in to allow users of CMS to test these tools in real data. The prototype has been developed and tested by the owners of this work, in addition, an interview has been designed and send to the CMS community to give feedback. Due to the lack of time, and vocational periods, some interview answers are missing at the moment of the writing of this work and further analysis will be presented in future work. The prototype has been contributed to the CMS community being a clear outcome of this work.

Last, but not least, the third objective has been to present and open a debate around achieving Fairness in NLP sentiment analysis. In this section, we have presented the state-of-the-art present in the FAT/ML community defining shared concepts about what is Fairness and presenting some approaches to mitigate biases in datasets and also directly in ML models.

For future work, a tool to evaluate NLP classifiers models integrating the conclusion of this work and the state-of-the-art about Fairness of the FAT/ML community could be done. This tool will allow users and stakeholders to evaluate its models in a different set of situations and contexts. Furthermore, the CMS plug-in prototype developed in this work could be improved to provide a web interface inside the CMS to evaluate the models. This will avoid the lack of data, as CMS is a source of data, and will avoid complexity to the user to evaluate in real-time the behavior of its models.

7. Bibliography

[1] O. T. Sameh Al-Natour, "A comparative assessment of sentiment analysis and star ratings for consumer reviews, International Journal of Information Management,," *International Journal of Information Management, 2020,* vol. 54, no. https://doi.org/10.1016/j.ijinfomgt.2020.102132, 2020.

[2] W. V. I. &. A. H. Souma, "Enhanced news sentiment analysis using deep learning methods.," *Journal of Computational Social Science,* pp. 33-46, 2019.

[3] M. S. Ema Kušen, "Politics, sentiments, and misinformation: An analysis of the Twitter discussion on the 2016 Austrian Presidential Elections.," *Online Social Networks and Media.,* pp. 37-50, 2018.

[4] R. K. Wike, "Global support for principle of free expression, but opposition to some forms of speech," *Pew Research Center,* vol. 18, 2015.

[5] L. Hanu, "Detoxify," Unitary Team, 2020. [Online]. Available: Github. https://github.com/unitaryai/detoxify.

[6] C. &. G. E. Hutto, "Vader: A parsimonious rule-based model for sentiment analysis of social media text.," *In Proceedings of the International AAAI Conference on Web and Social Media,* vol. 8, no. 1, 2014.

[7] W. consortium, "W3C," May 2021. [Online]. Available: https://w3techs.com/technologies/overview/content management.

[8] J. Cabot, "Wordpress: A content management system to democratize publishing," in *IEEE Software,* 2018.

[9] A. I. M. &. V. E. Jobin, "The global landscape of AI ethics guidelines," *Nature Machine Learning,* no. https://doi.org/10.1038/s42256-019-0088-2, pp. 389-399, 2019.

[10] F. Rosenblat, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psycological Review,* pp. 386-408, 1958.

[11] I. A.M Touring, "Computing machinery and intelligence," *Mind,* vol. LIX, no. 236, pp. 433-460, 1950.

[12] D. W. Noam Chomsky, "Syntactic Structures," *Mouton,* 1957.

[13] inc., Kaggle, *Kaggle competitions,* Santa Clara County, California, USA, 2018.

[14] N. S. N. P. J. U. L. J. A. N. G. L. K. I. P. Ashish Vaswani, "Attentions Is All You Need," *Advances in neural information processing systems,* pp. 5998-6008, 2017.

[15] Q. Do, "Jigsaw unintended bias in toxicity classification.," 2019.

[16] S. V. a. J.Rubin, "Fairness Definition Explained," *IEEE/ACM Internationl Workshop on Software Fairness (FairWare), 2018,* no. 10.23919/FAIRWARE.2018.8452913, pp. 1-7, 2018.

[17] C. J. Gunning D. Stefik M, "XAI - Explainable artificial intelligence," *Science Robotics,* vol. 4, 2019.

[18] E. S. D. D. Fedor Kitashov, "Foreign English Acent Adjustment by Learning Phonetic Patterns," *https://arxiv.org/abs/1807.03625,* 2018.

[19] G. K. a. G. P. Stefano M. Iacus, "Causal Inference Without Balance Cheking: Coarsened Exact Matching," *Political Analysis 20,* pp. 1-24, 2012.

[20] B. Efron., "Boostrap Methods, Another look at the Jackknife," *The Annals of Statistics 7,* pp. 1-26, 1979.

[21] B. L. a. M. M. Brian Hu Zhang, "Mitigating unwanted biases with adversarial learning," *AAAI/ACM Conferece on AI, Ethics, and Society,* pp. 3335-340, 2018.

[22] Y. Awwad, R. Fletcher, D. Frey, A. Gandhi, M. Najafian and M. Teodorescu, "Exploring Fairness in Machine Learning for International Development," MIT D-LAb, Massachussets, 2020.

[23] E. P. a. N. S. Mortiz Hardt, "Equality of opportunity in supervised learning," Curran Associates Publishers, New York, 2016.

[24] K.-W. C. J. Y. Z. Tolga Bolukbasi, "Man is to computer programmer as woman is to homemakers? Debiasing word embeddings," Adances in Neuronal Information processing Systems, Barcelona, 2016.

[25] J. L. C. R. a. R. S. Matt Kusner, "Counterfactual fairness," Advances in Neuronal Information Processing Systems, Long Beach, 2017.

[26] Z. S. S. A. A. A. G. M. a. A. S. Y.A. Solangi, "Review on Natural Language Processing (NLP) and Its Toolkits for Opinion Mining and Sentiment Analysis,," *EEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS),* pp. 1-4, 2018.

[27] S. G. M. S. Y. c. N. A. S. S. Gehman, "Evaluating Neural Toxic Degeneration in Language Models," *arXiv preprint,* no. arXiv:2009.11462., 2020.

[28] Y. a. Z. H. a. L. T. a. Z. P. a. G. N. Xia, "Exploring Antecedents and Consequences of Toxicity in Online Discussions: A Case Study on Reddit," *Association for Computing Machinery,* vol. 4, no. CSCW2, p. 23, 2020.

[29] M. A. B. M. E. G. G. S. X. R. I. &. B. K. Greenwood, "Online abuse of UK MPs from 2015 to 2019," *arXiv preprint,* no. arXiv:1904.11230., 2019.