UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF A BROWSER PLUGIN FOR INTELLIGENT TEXT ANALYTICS

MANUEL DEL POZO DÍAZ

MAYO 2022

TRABAJO DE FIN DE GRADO

Título:	Diseño y desarrollo de un plugin de Navegador para el Ánalisis Inteligente de Texto
Título (inglés):	Design and Development of a Browser Plugin for Intelligent Text Analytics
Autor:	Manuel del Pozo Díaz
Tutor:	Carlos Ángel Iglesias Fernández
Departamento:	Departamento de Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:	—
Vocal:	—
Secretario:	—
Suplente:	—

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

TRABAJO FIN DE GRADO

DESIGN AND DEVELOPMENT OF A BROWSER PLUGIN FOR INTELLIGENT TEXT ANALYTICS

Manuel del Pozo Díaz

Mayo 2022

Resumen

El concepto de Internet de las Cosas se basa en explicar cómo la conexión entre elementos físicos e Internet permite a dispositivos recoger y extraer datos, interactuando con el mundo real desde una perspectiva exterior. En el mundo actual, ser capaz de entender y analizar completamente cualquier texto en Internet es una herramienta extremadamente útil que nos permite obtener una visión general sobre determinados temas. Las aplicaciones del análisis de sentimientos son amplias y potentes. La capacidad de extraer información de los datos sociales puede proporcionar al usuario información importante.

El objetivo final de este proyecto es desarrollar una extensión de Google Chrome que permita a cualquier usuario realizar un análisis de texto en el navegador web. Con este proyecto, el usuario podrá personalizar ciertos parámetros de la extensión en la página de configuración, y seleccionar diferentes tipos de análisis de texto, cada uno de los cuales dará diferentes resultados. Para el usuario, el hecho de poder analizar un texto de forma instantánea y obtener información sobre el sentimiento o la emoción del mismo, abre toda una serie de nuevas posibilidades para entender mejor ciertos temas.

Para hacerlo fácil y comprensible para todos, la extensión ofrece los resultados mediante gráficos sencillos, que proporcionan la información y los parámetros necesarios para el usuario con una fácil visualización de los datos. Además, algunos de los servicios tienen una función añadida de estilizar dinámicamente el texto seleccionado en el navegador web con los diferentes sentimientos y emociones, lo que también ayuda a conseguir una mejor comprensión.

Finalmente, presentamos las conclusiones extraídas de este proyecto, las tecnologías que hemos utilizado durante el desarrollo, los problemas a los que nos hemos enfrentado durante el mismo y las posibles líneas de trabajo futuras.

Palabras clave: personalizar, comprensible, extensión de Chrome, análisis de sentimientos, análisis de texto

Abstract

The Internet of Things concept is based on how the connection of physical things to the Internet enable devices to collect and exchange data, interacting with real world from a distance. In today's world, being able to fully understand and analyze any text on the Internet is an extremely useful tool that allow us to gain an overview on certain topics. The applications of sentiment analysis are broad and powerful. The ability to extract insights from social data can give the user important information.

The ultimate goal of this project is to develop a Google Chrome extension that allows any user to perform a text analysis in the web browser. With this project, the user will be able to customize certain parameters of the extension in the configuration page, and select different types of text analysis, each one giving different results. For the user, being able to analyze text instantaneously and get information about the sentiment or emotion of the text, opens a whole new possibilities to better understand certain topics.

To make it easy and comprehensible for everyone, the extension gives the results using simple charts, that provide the necessary information and parameters for the user with an easy data visualization. Also, some of the services have an added feature of styling dynamically the selected text on the web browser with the different sentiment and emotions, which helps getting a better understanding too.

Finally, we present the conclusions drawn from this project, the technologies we have used during the development, the problems that we have faced during the development and possible future lines of work.

Keywords: customize, comprehensible, plugin chrome, sentiment analysis, text analysis

Agradecimientos

A mis padres, mi hermano y a mi familia, por apoyarme y animarme siempre a seguir hacia delante, sin ellos no habría llegado hasta aquí.

A toda la gente que he conocido a lo largo de este camino, y que ha estado ahí para mí, para ayudarme en los buenos y en los malos momentos, y también a los que habéis aparecido al final cuando más lo necesitaba, para darme un último empujón.

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Context

The field of sentiment and emotion classification is an exciting new research direction due to the large number of real world applications where discovering people's opinion is important for better decision-making. Sentiment and Emotion Analysis [3] (explained in Subsec.3.4.1) is the study that analyzes people's opinion and sentiment towards entities such as products, services in the text. It has always been important to know what other people think. With the rapid growth of user-generated data on the Web, people are using online review sites, blogs, forums, social networking sites, and so forth for expressing their opinion. Therefore, a necessity of analyzing and understanding these online generated data/reviews has arisen. Users can know the merits and demerits of the product from the experiences shared by people on the web, which can be useful for them in decision-making. E-commerce companies can improve their product or services on the basis of people's opinion and current trends. The automatic analysis of online contents to extract opinion requires deep understanding of natural text by the machine; capabilities of most of the existing models are known to be unsatisfactory.

An opinion lexicon [4] is a dictionary containing opinion words with their polarity value

to indicate the positive or negative sentiments or emotions, for example: “happy”, “anger”, “bad”, “fear”, and so forth. These opinion words are used in most of the existing sentiment and emotion analysis models as a key indicator of the opinion of the user. Several opinion lexicons are publicly available like SentiWordNet [5], SenticNet [6], emotion ANEW [7], etc...

With this project, we propose an approach, which selects only important features and aspects about which any opinion is expressed with the help of automatically constructed ontology. Further, polarity of an opinion word is determined with the help of contextual sentiment lexicon. Further, sentiment or emotional information with respect to each attribute of the product is aggregated according to its importance with the topic, and finally sentiment or emotion of the document is determined..

1.2 Project goals

The main purpose of this project is to develop a Google Chrome [8] extension that´s capable of analysing any text on any web page and give the results. In order to do this, the extension will have to listen for the user inputs, then communicate with the different third party APIs, and take the response and adapt the results for an easier data visualization. The most important project sub-goals are:

- Design and development of a Google Chrome extension that is able to analyse text on the web page.
- Implement a configuration page for the user to customize certain parameters.
- Implement a system to give a comprehensive visualization of the results to the user using charts and dynamic styling.
- Distribute the plugin in the Chrome Web Store for anyone to download them.

1.3 Structure of this document

In this section we provide a brief overview of the chapters included in this document. The structure is as follows:

Chapter 1 gives a brief introduction of the project and explains the context and motivation in which this project is developed. Further, it also describes the main goals to

achieve in this project.

Chapter 2 describes the technologies that this project is going to be using for design, development and implementation. It also discusses the different technologies that we could have chosen and why they weren't chosen.

Chapter 3 goes into detail with the architecture used to implement the project. It starts with and overview of it and then it describes each module and how all the modules work with each other.

Chapter 4 offers an overview on how the extension works in different use cases, explaining the main functionalities to help the user understand the overall concept.

Chapter 5 collects the conclusions obtained during the development of the project and provides a future perspective and some suggestions to implement in near future.

CHAPTER 2

Enabling Technologies

2.1 Introduction

This chapter provides a general overview of all the frameworks, libraries and technologies implemented in this project. Firstly we will introduce the scientific fields this project is based on. This includes the concepts of Natural Language Processing (NLP).

The goal is to able to analyze plain text from any web page, and for that we will need some tools, the most important being Natural Language Processing. NLP refers to the branch of computer science (and more specifically, the branch of Artificial Intelligence (AI)) concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

2.2 Natural Language Processing

NLP [9] combines computational linguistics (rule-based modeling of human language) with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data to "understand" its

full meaning, complete with the writer or speaker's intent and sentiment.

Human language is filled with ambiguities that make it incredibly difficult to write software that accurately determines the intended meaning of text or voice data: homonyms, homophones, sarcasm, idioms, metaphors, grammar and usage exceptions, variations in sentence structure... These are just a few examples of the irregularities of the human language that programmers must teach to the natural-driven applications to recognize and understand accurately from the start [10], if those applications are going to be useful. Natural Language Processing is a mixture of linguistics and artificial intelligence:

- **Artificial Intelligence:** AI is the field of computer science that focuses on making computers "intelligent". In its more basic form, AI is the combination of computer science concepts with robust data sets, enabling problem-solving. In addition, the algorithms used in AI try to make predictions or classifications based on input data. The two types of Artificial Intelligence are: Weak AI focused on performing a specific task and Strong AI, a theoretical concept where machines have a broader sense of intelligence.

- **Linguistics:** Linguistics [11] is the scientific study and modelling of language. Language can be divided into five levels:

1) *Phonology:* It is the study of sounds humans can make, that are usable in language. Each language has shared sounds and characteristic ones, making it a different case of study for each. Phonemes are the smallest units of language and they do not have meaning by themselves. The English language is composed of a total of 44 phonemes.

2) *Morphology:* it is the study of morphemes, the smallest units of language with meaning and lexers, structural variations applied to morphemes modifying their meaning. It takes into consideration single words only, it does not study the structural relationship between words in a sentence.

3) *Syntax:* sentences are structured groups of words that follow a set of rules, these are the area of study of syntax.

4) *Semantics:* whilst the other fields of linguistics are more focused on the structure of language, semantics is focused on the meaning of words and sentences. Furthermore, sentences are can be statements (true or false), questions or requests.

5) *Pragmatics:* it is similar to semantics, but it centres on what sentences and words imply in different situations.

To achieve great NLP, we will be using one of the technologies available, Senpy, described below.

2.2.1 Senpy

Senpy [1] is a framework to build sentiment and emotion analysis services. Its goal is to produce analysis services that are interchangeable and fully inter operable. All services built using Senpy share a common interface. This allows users to use them (almost) interchangeably, with the same Application Programming Interface (API) and tools, simply by pointing to a different Uniform Resource Locators (URL) or changing a parameter. The common schema also makes it easier to evaluate the performance of different algorithms and services. In fact, Senpy has a built-in evaluation API you can use to compare results with different algorithms.

Figure 2.1 shows the most relevant characteristics of the Senpy architecture. The framework consists of two main modules: Senpy core, which is the building block of the service, and Senpy plugins, which consist of the analysis algorithm. The next figure depicts a simplified version of the processes involved in an analysis with the Senpy framework. Each plugin will be composed of different machine learning algorithms for analyzing the data provided.

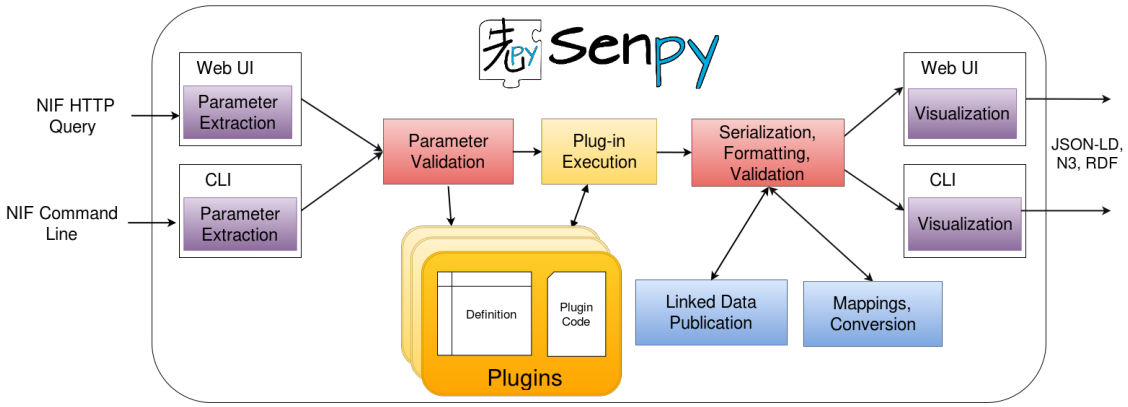

Figure 2.1: Senpy architecture [1]

To achieve this goal, Senpy uses a Linked Data principled approach, based on the NIF (NLP Interchange Format) specification [12], and open vocabularies such as Marl [13] and Onyx [14].

All this, in addition to the simplicity to merge different sentiment and emotion algorithms, makes Senpy the perfect technology to implement in this project for NLP.

When the fetch is sent, the Senpy API receives it and starts to process it. The model used in Senpy is based on NIF 2.0 [15], which defines a semantic format and API for improving interoperability among natural language processing services.

Senpy has been applied to sentiment and emotion analysis services using the following vocabularies:

- Marl [14], a vocabulary designed to annotate and describe subjective opinions expressed on the web or in information systems.
- Onyx [16], which is built one the same principles as Marl to annotate and describe emotions, and provides interoperability with Emotion Markup Language.

All Senpy plugins use the same API, which makes moving from one service to another a breeze. We can do it just by changing the URL of the service. Another good point for Senpy is that the structure and annotation schema of the response is the same. This makes it very easy to compare and migrate to different services. Service interoperability is not only useful for users. It is also key for other features such as automated evaluation.

Senpy counts with great variety of plugins and different services, plus the ability to upload your own plugins. A text analysis service must implement all the typical features, such as: extraction of parameters, validation, format conversion, visualization...

Senpy implements all the common blocks, so developers can focus on what really matters: great analysis algorithms that solve real problems. Among other things, Senpy takes care of these tasks:

- Interfacing with the user: parameter validation, error handling.
- Formatting: JSON-LD, Turtle/n-triples input and output, or simple text input
- Linked Data: Senpy results are semantically annotated, using a series of well established vocabularies, and sane default URIs.
- User interface: a web UI where users can explore your service and test different settings
- A client to interact with the service. Currently only available in Python.

2.2.2 NLP Approaches

We can find three different approaches when it comes to NLP way of resolving problems [17]:

- **Rule-based or Heuristics:** It consists of building a set of rules to do a task, requiring previous knowledge from the developer in the field of work. In most cases, they require a dictionary of words and the use of Regular Expressions or context-free grammar(CFG).

- **Machine Learning:** The distributional approach using ML is focused on comparing the similarity of words based on the statistical frequency of occurrence in other texts. For instance, the word "calculus" is more probable to appear in a text about mathematics than in one about politics. After having the statistical representation of a text, traditional machine learning algorithms can be applied. Although this approach does not rely on the meaning of words, it can be useful for text classification or checking if words are being used similarly. The Machine Learning approach, either supervised or unsupervised has three shared steps: feature selection, learning the model from the features and evaluating the model. This model is very flexible, as it treats text as arrays of data.

- **Deep Machine Learning:** These networks are composed of layers of nodes, in which the first layer is the input, the last one is the output and all the layers in between remaining hidden. Nodes are connected to other nodes and they have an associated weight and threshold, that determine the importance of a variable in the model. This method has the advantage that, unlike ML, does not need the feature selection phase as the deep machine learning does it. Some of the more common neural networks used in NLP are recurrent neural networks, convolutional, transformers and auto-encoders. Two of the main problems of using this approach are that it is easy to do over fitting when the data sets are not big enough and the high cost involved in training the networks.

2.3 Web Technologies

Ever since the computer and internet have taken over our lives, web technology is evolving and upgrading. But what is web technology? It is a mechanism by which computers communicate with each other and with the web servers with the aid of markup languages and multimedia packages.

With the help of web technology, one can build a communication platform with different functionalities and utilities using programming languages. In real-life scenarios, web technology provides higher and faster mutual device communication thereby, bringing optimal automation. Being aware of web technology definition is essential in today's world because the internet has become an indispensable part of our personal and professional lives.

We can check two of the most important aspects: web development frameworks and programming languages:

- A *web development framework* is a group of tools and resources for software developers to build and manage web services, web applications and websites. We can find a lot

of examples, such as: Angular, Ruby, MeteorJS, Express.js, Django...

- *Programming language* refers to any set of rules that converts strings, or graphical program elements (in the case of visual programming languages), to various kinds of machine code output, and they are used to implement algorithms.

2.3.1 JavaScript

JavaScript (JS) [18] is a lightweight, interpreted, just-in-time compiled programming language with first-class functions, dynamic typing and prototype-based object-orientation. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). Alongside HTML and CSS, JavaScript [18] is one of the core technologies of the World Wide Web.

JavaScript is the dominant client-side scripting language of the Web, with 97% of websites using it for this purpose. Scripts are embedded in or included from HTML documents and interact with the DOM. All major web browsers have a built-in JavaScript engine that executes the code on the user's device.

With the expanded demands for JavaScript, an easier means for programmers to develop such dynamic interfaces was needed. Thus, JavaScript libraries were developed. This libraries consist of pre-written JavaScript code that allows for easier development of JavaScript-based applications, especially for AJAX and web-centric technologies. Almost all of the JavaScript libraries are released under either a copycenter or copyleft license to ensure license-free distribution, usage and modification. Over 80% of websites use a third-party JavaScript library or web framework for their client-side scripting.

For this project, we have used one of these libraries, Chart.js [19], that allows us to visualize the data in an easier way.

2.3.2 JavaScript Libraries

For this project, Chart.js has been selected, as its the best fit for us and for the needs of the extension. But before selecting Chart.js, we've studied some more options:

2.3.2.1 Plotly

Plotly [20] is a JS library, that has been open-source since 2015, meaning anyone can use it for free. Plotly supports 20 chart types, including SVG maps, 3D charts, and statistical

graphs. It's built on top of D3.js and stack.gl.

Figure 2.2: Plotly Logo

2.3.2.2 NVD3

NVD3 [21] is another JS library built on top of D3.js. It's completely open-sourced. While performance is relatively good, the problem with this library is the lack of chart variations and low customization of the interface.

Figure 2.3: NVD3 Logo

2.3.2.3 D3.js

D3 [22] is an open-source JavaScript library released under the BSD license. It provides a tremendous amount of charts, graphs, and other methods for data visualization. D3 gives you almost everything you need to visually represent your data of any kind. There are two major concerns with D3.js: it has a steep learning curve and it is compatible only with modern browsers (IE 9+).

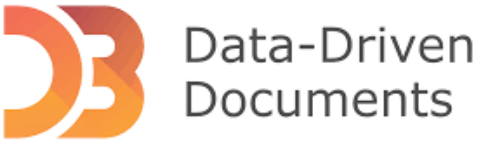

Figure 2.4: D3 logo

2.3.2.4 HighCharts

HighCharts [23] is a modern JavaScript charting library based on SVG technology. The integration with all of the major web frameworks is very simple. In all of its simplicity, Highcharts is also very much compatible with old browsers, so you can use it if you don't need to represent data using advanced charting styles.

Figure 2.5: HighCharts Logo

This solution would be an interesting one, but the complexity and the extra customization that this library adds, makes it fall of the list.

2.3.2.5 Chart.js

Chart.js [19] is a free, open-source JavaScript library for data visualization, which supports eight chart types: bar, line, area, pie (doughnut), bubble, radar, polar, and scatter. Created by London-based web developer Nick Downie in 2013, now it is maintained by the community and is the second most popular JavaScript charting library on GitHub by the number of stars after D3.js, considered significantly easier to use though less customizable than the latter. Chart.js renders in HTML5 canvas and is widely covered as one of the best data visualization libraries.

Figure 2.6: Chart.js Logo

2.3.3 Chrome extension

Extensions are software programs, built on web technologies (such as HTML, CSS, and JavaScript) that enable users to customize the Chrome browsing experience. They run in a separate, sand boxed execution environment and interact with the Chrome browser. They are made of different, but cohesive, components. Components can include backgrounds scripts, content scripts, an options page, User Interface (UI) elements and various logic files. Extensions components are created with web development technologies: HTML, CSS and JavaScript. An extensions components will depend on its functionality and may not require every option.

Extensions let you "extend" the browser by using APIs to modify browser behaviour and access web content. Extensions operate by means of an en-user UI and developer API:

- **User interface:** This provides a consistent way for users to manage their extensions.

- **Extensions APIs:** The extensions APIs allow the extension's code to access features of the browser itself: activating tabs, modifying net requests, and so on.

To create an extension, you need to assemble some resources - a manifest, JavaScript and HTML files, images and others - that constitute the extension. For development and testing, you can load the "unpacked" into Chrome using extension developer mode. Once you are happy with your extension, you can package it and distribute it to users.

Every extension has a JSON-formatted manifest file, named `manifest.json`, that provides important information, shown in Listing 2.1:

```
{
  // Required
  "manifest_version": 3,
  "name": "My Extension",
  "version": "versionString",

  // Recommended
  "action": {...},
  "default_locale": "en",
  "description": "A plain text description",
  "icons": {...},
  "permissions": ["storage", ...]

  // Optional
  "author": ...,
  "automation": ...,
  "background": {
    // Required
    "service_worker": "background.js",
    // Optional
    "type": ...
  },
  "chrome_settings_overrides": {...},
  "chrome_url_overrides": {...},
  "commands": {...},
  "content_capabilities": ...,
  "content_scripts": [{...}],
  "content_security_policy": {...},
  "converted_from_user_script": ...,
  "cross_origin_embedder_policy": {"value": "require-corp"},
  "cross_origin_opener_policy": {"value": "same-origin"},
  "current_locale": ...,
  "declarative_net_request": ...,
  "devtools_page": "devtools.html",
  "differential_fingerprint": ...,
  "event_rules": [{...}],
  "externally_connectable": {
    "matches": ["*://*.example.com/*"]
  }
}
```

Listing 2.1: manifest.json example

The directory holding the manifest file can be added as an extension in developer mode in its current state. Just go to `chrome://extensions` and click on "Load unpacked", and then select that directory.

The next step is to add the functionality. One of the basics is to add a background script. A background page is loaded when it is needed and unloaded when it goes idle. Once it has been loaded, it will stay running for as long as it is performing an action. Effective background scripts stay dormant until an event they are listening for fires, react with specified instructions, then unload. This scripts are usually used for initializing the extension, setting up listeners so they can be triggered around the events the extension depends on and to react to those events, using APIs that support event filters to restrict listeners if needed, etc... Most APIs must be registered under the `"permissions"` field in the manifest for the extension to use them.

Extensions can have different ways for a user interface, but the most common one is the popup. Just create an HTML file, name it `popup.html` and add it to the extension's directory. If you want to add some functionality to the popup too, you have to create a `popup.js` file.

Now you already have a working extension with some simple features. To improve it a little bit more, you can add the possibility for the user to have more control over the extension's functionality, adding these files: `options.html` and `options.js` ; so they can further customize their browsing experience. Once you've added these files, reload the extension and right-click on the extension's icon, and then select "Options".

2.3.4 CORS

Cross-Origin Resource Sharing (CORS) [2] is an HTTP-header based mechanism that allows a server to indicate any origins (domain, scheme, or port) other than its own from which a browser should permit loading resources. CORS also relies on a mechanism by which browsers make a "pre-flight" request to the server hosting the cross-origin resource, in order to check that the server will permit the actual request. In that pre-flight, the browser sends headers that indicate the HTTP method and headers that will be used in the actual request. The most common situation is when one domain/app/website makes a request (XMLHttpRequest or Fetch APIs) for data (usually JSON) to another domain/app/website, as shown in Figure 2.7.

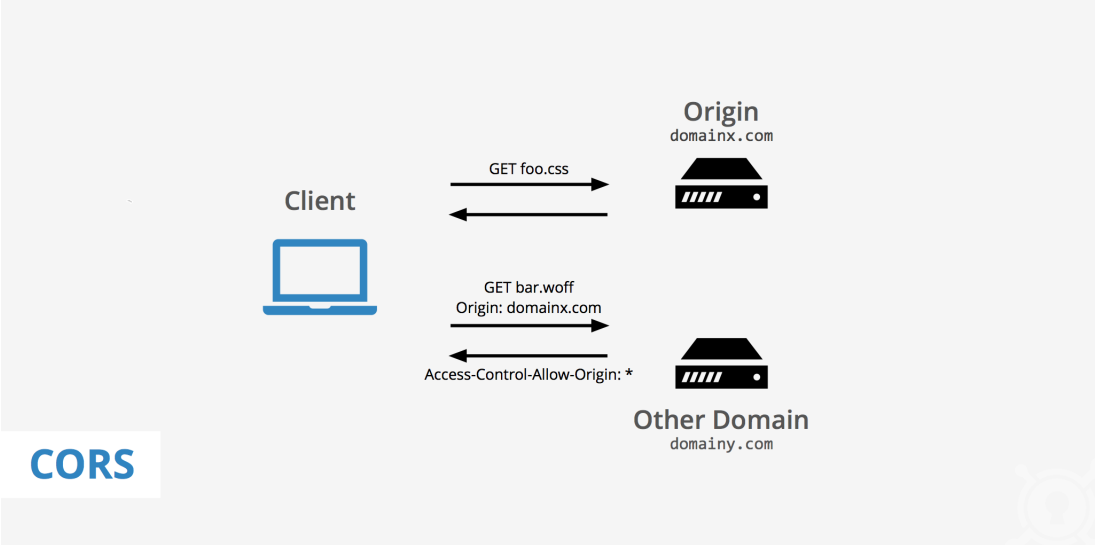

Figure 2.7: CORS example [2]

This is all for security purposes like preventing a malicious site from reading another site's data. It also prevents legitimate access to "public resources". A way to get around this is by putting the appropriate information in the 'header' and 'body' of the request that will determine what data will be exchanged between the two origins. If approved, then it goes right on through. If not, then you'll get an error message. For the majority of the development of this project, to achieve this, we have used a link provided by Just Cors, which is a proxy which adds CORS headers to the proxied request so you can simply append the requested resource to the proxy link and the right headers will be added. The Figure 2.8 shows the main page and how does it work.

Figure 2.8: JustCORS.com

CHAPTER 3

Architecture

In this chapter, we cover the design phase of this project, as well as implementation details involving its architecture. Firstly, we present an overview of the project, divided into several modules. This is intended to offer the reader a general view of this project architecture and how the different elements interact and communicate with each other. Secondly, each module will be explained separately and in much more depth.

3.1 Architecture Overview

In this section, the architecture of the project will be explained, including the design and defining the different modules. The design of the architecture has followed the Model-View-Controller (MVC) pattern [24].

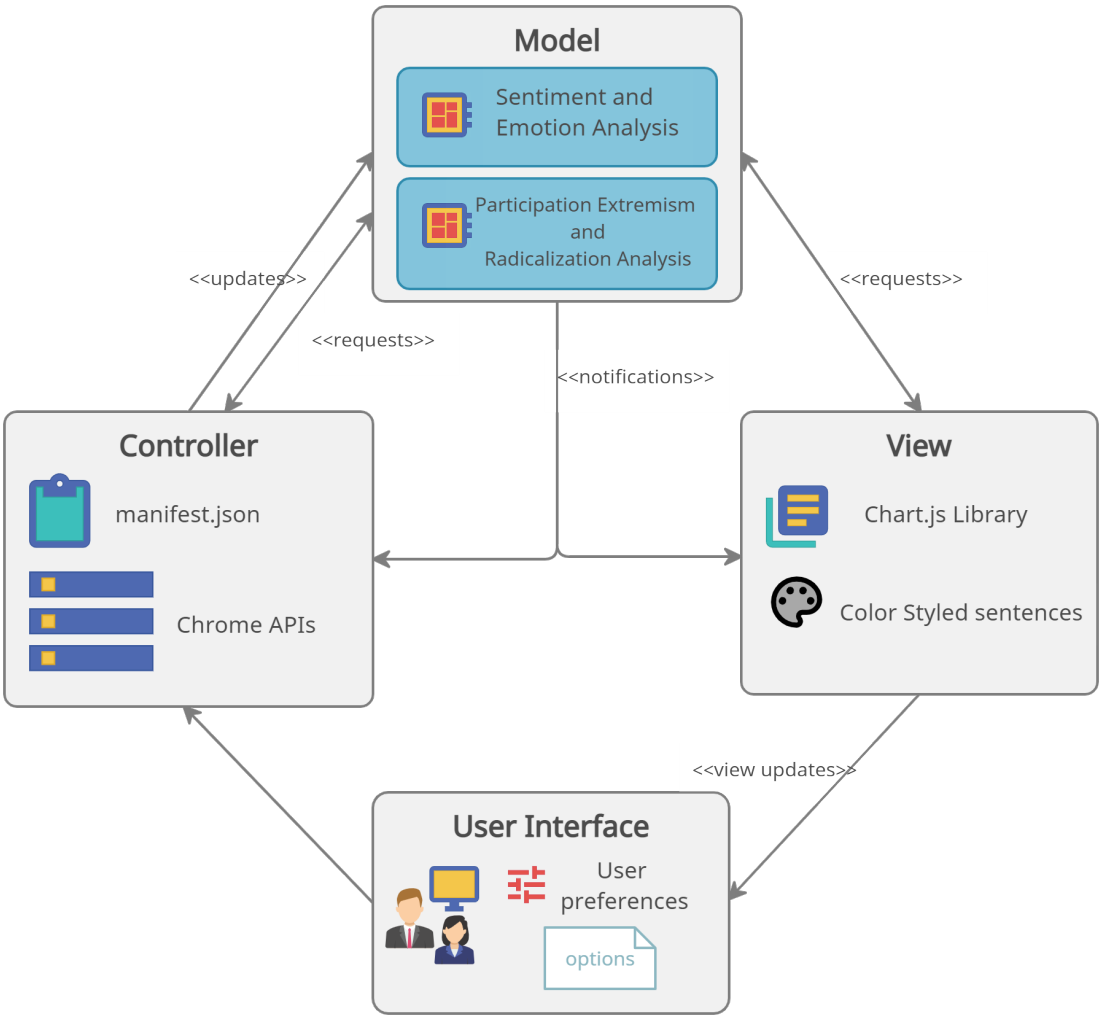

Figure 3.1: Architecture

The Figure 3.1 presents the general overview of the architecture. The architecture shows the four main modules: the User Interface, the Controller, the Model and the View. It also shows how each module interacts with each other.

- **User Interface:** This module is responsible of managing the inputs and preferences that the user selects.
- **Controller:** This module takes care of the inputs given by the User Interface and builds a request.
- **Model:** This module uses some external systems: the Sentiment and Emotion Analysis plugin and the Participation Extremism and Radicalization Analysis plugin, both from Senpy [1], explained on SUbsec. 2.2.1.

- **View:** The last module is in charge of giving a proper data visualization for the user. To accomplish this goal this module uses two different tools: a JavaScript library to build some charts and more JavaScript code to style sentences depending on the results.

3.2 User Interface

The User Interface has been designed to allow an easy management and customization for the user. It has been implemented using different technologies like HTML5, CSS3 and JavaScript.

The user has two different interfaces: the pop up (also called Splash screen) and the Configuration page.

3.2.1 Splash Screen

Figure 3.2: Splash Screen 1

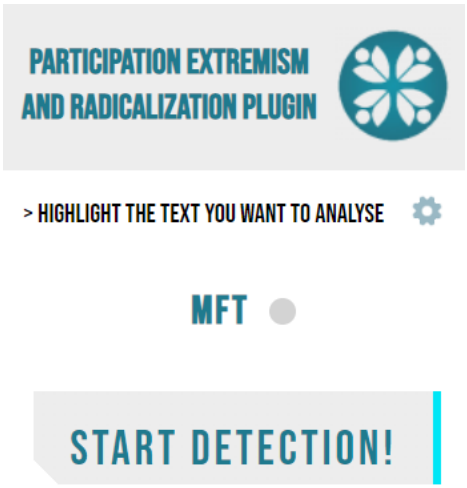

Figure 3.3: Splash Screen 2

This will be the main interface for the user. It has been customized to be simple yet interesting. The Figures 3.2 and 3.3 shows the main interface of the Chrome extension, made up using technologies such as HTML, CSS and JavaScript. The simple interfaces display the title and some basic instructions below. The interfaces have an interesting feature, the green circle. It indicates the possibility to use the color styled sentences mode, explained in Subsec. 3.5.2.

3.2.2 Configuration page

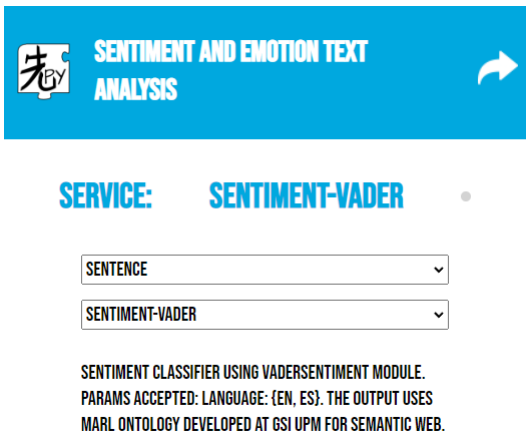

Figure 3.4: Configuration Page interface

This second interface is where the user can choose some options and save some preferences. There is list of all the different types of analysis that the user can select. Each option comes with a description and some graphic help.

To save the options and preferences, the JavaScript uses some APIs provided by Google Chrome. The API that will be used the most is going to be the `chrome.storage`. With this API we can store, retrieve and track changes to user data. It provides the same storage capabilities as the `localStorage` API with an important difference: it has a property called `storage.sync` that allows synchronization in Chrome profiles.

When using a web browser such a Google Chrome, you can have a profile to customize your internet experience. Your Chrome profile handles all of your Chrome browser preferences and settings including the extensions you like to use, bookmarks, saved passwords, and display settings (default fonts and color schemes).

Making use of the property `storage.sync` it allows us to save the user configuration of the extension in the Chrome profile.

```
chrome.storage.sync.set({key: value}, function() {
  console.log('Value is set to ' + value);
});

chrome.storage.sync.get(['key'], function(result) {
  console.log('Value currently is ' + result.key);
});
```

Figure 3.5: Storage API Structure

In the Figure 3.5 it is shown the structure of the `chrome.storage.sync` API. To save a user preference, we use the property `.set`, and whenever the new preference needs to be applied it uses the property `.get`.

At the moment, the user can customize two different parameters: the service and the delimiter. Both will be explained later.

3.3 Controller

The main purpose of this module is to collect the user inputs to build and send a HTTP request to an external API. It has been developed only using JavaScript. The user inputs are two different parameters: the first one is the "service" and we get it from the Configuration page; the second parameter is the "text selection" that the user makes in the web page.

Every extension has a JSON-formatted manifest file, named `manifest.json`, that provides important information. With this file, you specify basic metadata about your extension and you can also specify aspects of your extension's functionality (such as background scripts, content scripts or browser actions). The Chrome browser makes sure the `manifest.json` is working. One of the great features of the manifest file, is that we can specify the Chrome APIs and browser actions that we want to use. To achieve this, we have to declare them on the "`permissions`" field of the manifest file. If the "Controller" module needs to call a URL, the URL needs to be declared on the manifest file, under the *permissions* field. In the following excerpt, we can see that the *permissions* field has a value of "all_urls", which means that any URL can have access to the extension.

In Listing 3.1 there is an excerpt of the `manifest.json`, where we can see some of the metadata specified:

```
{
    "name": "Sentiment and Emotion Analysis Plugin",
    "version": "1.6",
    "manifest_version": 3,
    "description": "Implementing Sentence Styling + New logo",

    ...

    "permissions": ["<all_urls>","storage", "activeTab", "scripting"],

    ...

    "icons": { ... },
}
```

Listing 3.1: JSON-LD excerpt

When building the HTTP request, we need to keep in mind that we have to add some extra HTTP headers, so that we can allow CORS 2.3.4.

The structure of the request is made up by the following parts:

- **Endpoint:** This is just the URL of the Senpy API, developed in Subsec. 2.2.1.
- **CORS:** Explained above in Subsec. 2.3.4.
- **Split Plugin:** This parameter allows us to divide the user's text selection into sentences or paragraphs. Its value can be modified in the Configuration page
- **Service:** These are the different types of text analysis available. There are many different services we can choose from. All the services are explained in Subsec. 3.4.1.
- **Text Input:** This parameter is a *string* that contains the text that the user highlighted in the web page.

The Figure 3.6 shows how once the HTTP request has been sent to the external API, the "Controller" will receive a JSON-LD as an answer.

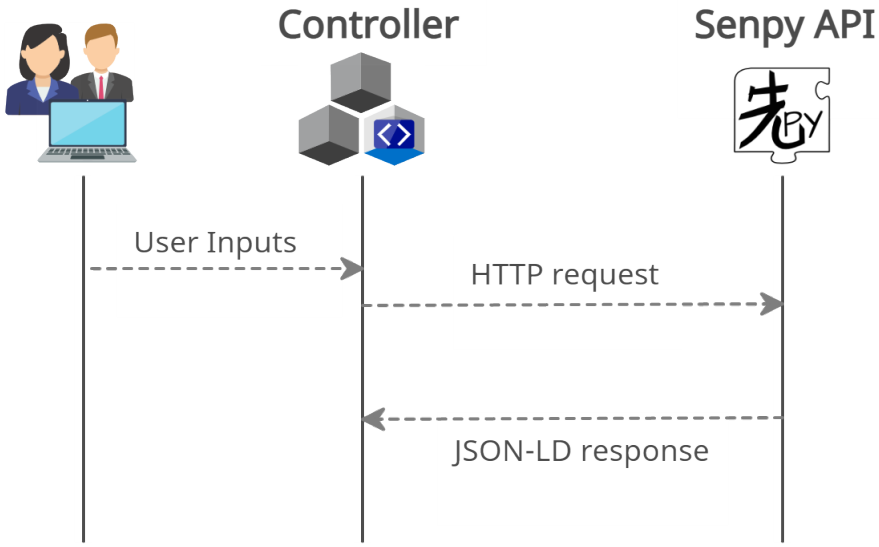

Figure 3.6: HTTP Request Process example

In Listing 3.2 there is an excerpt of the JSON-LD response received after sending a request:

```
{
    "entries": [
    {
      "@id": "prefix:",
      "@type": "Entry",
      "marl:hasOpinion": [
        {
          "@type": "Sentiment",
          "marl:hasPolarity": "marl:Positive",
          "prov:wasGeneratedBy": "prefix:Analysis_1651161672.9792848"
        }
      ],
      "nif:isString": "Who knew NLP and text preprocessing could be so easy with
          python? #DataScience #NLP",
      "onyx:hasEmotionSet": []
    },
    {
      "@id": "prefix:#char=0,65",
      "@type": "Entry",
      "marl:hasOpinion": [
        {
          "@type": "Sentiment",
          "marl:hasPolarity": "marl:Positive",
          "prov:wasGeneratedBy": "prefix:Analysis_1651161672.9792848"
        }
      ],
```

```
        "nif:isString": "Who knew NLP and text preprocessing could be so easy with
            python?",
        "onyx:hasEmotionSet": []
      },
      {
        "@id": "prefix:#char=66,83",
        "@type": "Entry",
        "marl:hasOpinion": [
          {
            "@type": "Sentiment",
            "marl:hasPolarity": "marl:Neutral",
            "prov:wasGeneratedBy": "prefix:Analysis_1651161672.9792848"
          }
        ],
        "nif:isString": "#DataScience #NLP",
        "onyx:hasEmotionSet": []
      }
    ]
  }
```


Listing 3.2: JSON-LD excerpt

The most important parts of the JSON-LD are the following parameters: "`marl:hasPolarity`", "`onyx:hasEmotionCategory`" and "`onyx:hasEmotionIntensity`". The "View" module will pick these values to show the results to the user properly.

3.4 Model

The "Model" module plays a very important role. There are tow different models depending on which plugin you use: the Sentiment and Emotion text analysis, and the Participation Extremism and Radicalization analysis. Each of this models has unique features, but both models follow the same basics.

3.4.1 Sentiment and Emotion Text Analysis

Figure 3.7: Logo for this model

This model integrates eight different services, but we can add as many as we want. Let's give a more in depth explanation of each service:

- **emotion-anew:** This plugin [25] consists on an emotion classifier that detects six possible emotions:
  - Anger : general-dislike.
  - Fear : negative-fear.
  - Disgust : shame.
  - Joy : gratitude, affective, enthusiasm, love, joy, liking.
  - Sadness : ingratitude, daze, humility, compassion, despair, anxiety, sadness.
  - Neutral: not detected a particularly emotion.

    The plugin uses ANEW lexicon dictionary [7] to calculate VAD(valence-arousal-dominance) [26] of the sentence and determinate which emotion is closer to this value. It will look for the words in the sentence that appear in the ANEW dictionary and calculate the average VAD score for the sentence. Once this score is calculated, it is going to seek the emotion that is closest to this value.
- **emotion-depechemood:** This is an emotion classifier. It uses depechemood [27], an emotion lexicon automatically generated from news articles where users expressed their associated emotions.
- **emotion-wnafect:** The plugin classifies among five different emotions:
  - anger : general-dislike.
  - fear : negative-fear.
  - disgust : shame.

  - joy : gratitude, affective, enthusiasm, love, joy, liking.
  - sadness : ingratitude, daze, humility, compassion, despair, anxiety, sadness.

  This plugin uses WordNet-Affect [28] to calculate the percentage of each emotion and gives the result of each one of them.

- **senticnet:** This plugin is a sentiment classifier that can detect three sentiments: positive, negative and neutral. This plugin connects to the Sencticnet free polarity detection API from GelBukh [6].
- **sentiment-basic:** This is a demo plugin that uses only some features from the TASS 2015 classifier [29]. It is a sentiment classifier that can detect three sentiments: positive, negative and neutral.
- **sentiment-meaningcloud:** This plugin is a sentiment classifier. This plugin uses API from Meaning Cloud to perform sentiment analysis, but you need to obtain an API key from meaningCloud signing up on their web page.
- **sentiment-vader:** Vader [30] is a plugin developed at GSI UPM for sentiment analysis. The response of this plugin uses Marl ontology [31] developed at GSI UPM for semantic web. It can detect three sentiments: positive, negative and neutral.
- **sentiment140:** This plugin is a sentiment classifier and connects to the sentiment140 free API [32]. This plugin detect three sentiments: positive, negative and neutral.

3.4.2 Participation Extremism and Radicalization Analysis

Figure 3.8: Logo for the new model

Taking advantage of the structure already developed for the Sentiment and Emotion Analysis, we can create a second plugin using the same architecture and almost the same code. This time, the plugin's goal is to detect extremism and radicalization in texts.

This model is in early development, and integrates four different services. To access them, we need to use a different endpoint: `senpy-participation.gsi.upm.es`.

Participation [33] is a project developed at GSI UPM,and one of the main tasks within Project Participation is the annotation of text with several types of information: emotions, narrative, ideology, etc. To that effect, and in alignment with the Linked Data principles, the project uses a combination of different ontologies dedicated to specific domains.

With this new endpoint, we can get in a different Senpy API, thus getting access into some new services that weren't available before:

- **propaganda:** This a simple plugin to analyse the text to detect any propaganda. It will give a result between 0 and 1 to see the propaganda level.
- **dbpedia:** As a result it will give the entities that are detected within the selected text.
- **liwc:** This plugin uses the Linguistic Inquiry and Word Count (LIWC) [34], the gold standard in software for analyzing word use. Basically, it analyses a given text and counts the percentage of words that reflect different emotions, thinking styles, social concerns, and even parts of speech.
- **mft:** The plugin is based on Moral Foundations Theory (MFT) [35] , created by a group of social and cultural psychologists to understand why morality varies so much across cultures yet still shows so many similarities and recurrent themes. It will give a numeric result of the five foundations of MFT: care/harm, fairness/cheating, loyalty/betrayal, authority/subversion and sanctity/degradation.
- **ExtremeSentilex:** This plugin its used to detect extreme sentiments using a lexicon [36] based on SentiWordNet and Senticnet. It can detect and show the specific words that will give a score depending on the extremism.
- **OSF Grievance:** This plugin is used to find the related topics about grievance of the selected text. This plugin uses The Grievance Dictionary [37], a psycho-linguistic dictionary that can be used to automatically understand language use in the context of grievance-fueled violence threat assessment. As a result it will give the related topics.
- **Hate Speech:** This plugin uses different lexicons [38] to find related topics with the hate detected on the selected text. This lexicons are: xenophobia lexicon (Hateful lexicon toward immigrants), immigrant lexicon (Contains words that refer to the

nationality of an immigrant), misogyny lexicon (Hateful lexicon toward women) and insult lexicon (General insults).

- **Offensive Language:** This plugin takes two lexicons that can be used to identify hate speech [39]. The plugin will give a score depending on the offensiveness level and the words that contributed to it.

3.5 View

The "View" module is responsible for visually rendering the contents received in the JSON-LD when sending the HTTP request, as showed in 3.6. This module uses HTML5, JSON and JavaScript to properly show the results in the User interface Sec. 3.2.

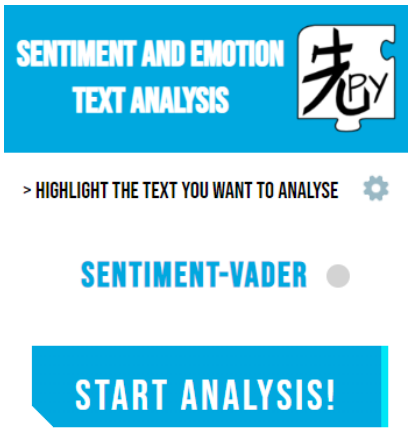

Figure 3.9: User interface before the text analysis.

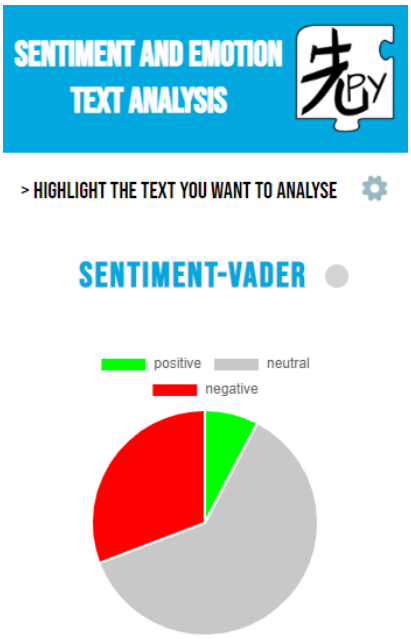

Figure 3.10: User interface after the text analysis.

The module has access to two different visualization tools: a chart maker (called Chart.js) and the dynamic styling of sentences depending on the sentiment or emotion they represent.

3.5.1 Chart.js

Chart.js has been introduced in Subsec. 2.3.2.5, but here we will explain it more in detail and argue why we used this tool instead of others. The other JS libraries that could be selected are described in Subsec. 2.3.2: Plotly in Subsec. 2.3.2.1, NVD3 in Subsec. 2.3.2.2, D3.js in Subsec. 2.3.2.3 and Highcharts in Subsec. 2.3.2.4.

After evaluating the different alternatives, Chart.js stands as the best possible one. It's the only one that gives us the customization level that we need with the easiest coding integration. Another good option would have been using D3.js, as both libraries enable the creation of common charts such as bar charts, line charts and scatter plots but their approaches differ significantly. Chart.js provides a selection of ready to go charts which can be styled and configured while D3 offers building blocks which are combined to create virtually any data visualisation. The reason why Chart.js was selected its because Chart.js renders its charts using the Canvas element which results in good performance compared with SVG, especially when rendering a large amount of data.

This library will be implemented in JavaScript in the "View" module, and will have access to the JSON-LD that we receive when making an HTTP request.

These are the chart types and their explanation on why we used each one of them:

- **Bar Chart:** A bar chart provides a way of showing data values represented as vertical bars. It is used to show the comparison of multiple data sets side by side. This type of chart is the most used on the extension. We can found it specially on the sentiment plugins, as their usual response is either: positive, negative or neutral.

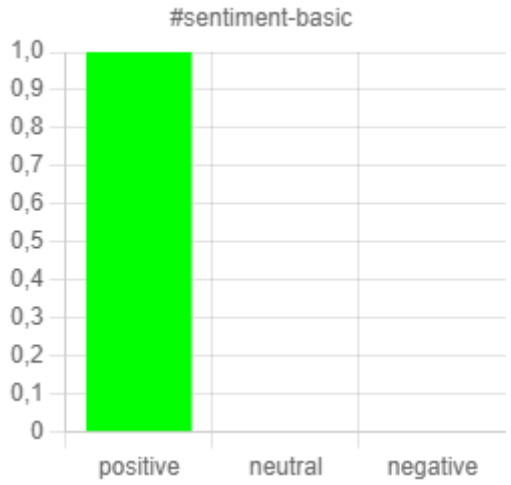

Figure 3.11: Bar Chart

- **Pie Chart:** This type of charts are divided into segments, the arc of each segment shows the proportional value of each piece of data. They are excellent at showing the relational proportions between data.

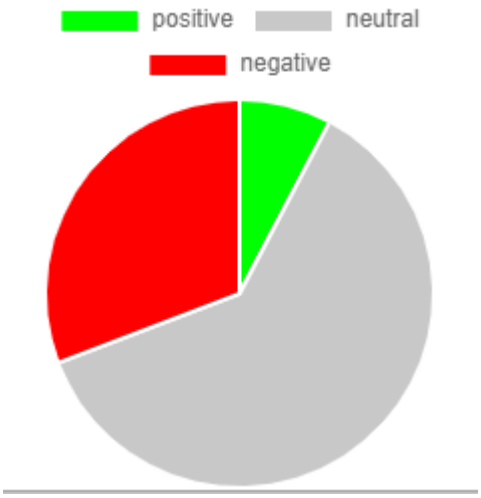

Figure 3.12: Pie Chart

- **Polar Area Chart:** Polar area charts are similar to pie charts, but each segment has the same angle - the radius of the segment differs depending on the value. This type of chart is often useful when we want to show a comparison data similar to a pie chart, but also show a scale of values for context.

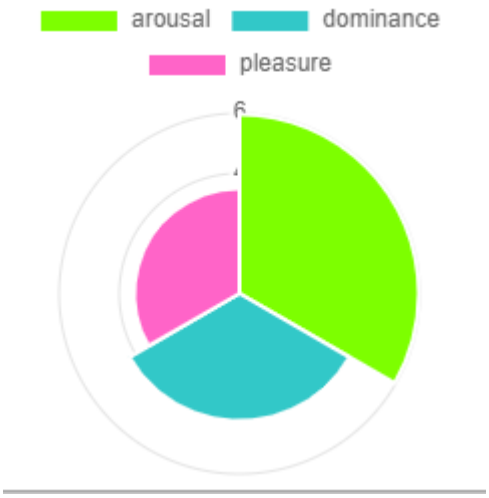

Figure 3.13: Polar Area Chart

- **Radar Chart:** A radar chart is a way of showing multiple data points and the variation between them.

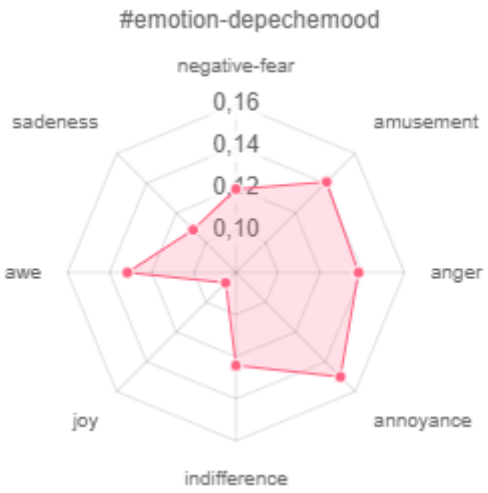

Figure 3.14: Radar Chart

3.5.2 Color Styled sentences

The other tool we can use in the "View" module, is the color styled sentences. With this tool, the module is able to color style to the sentences depending on the sentiment or emotion they represent. We can find two different color styling configuration: one for the service "emotion-anew" and another for the sentiment services ("senticnet, sentiment-basic and sentiment140"), explained in Subsec. 3.4.1.

For the "emotion-anew", we will add to the fetch a new parameter, which is the `emotion-model`, and in this case, we will use the model `emoml:big6`. This model is known as the "Six Basic Emotions", a term that refers to the theory of American psychologists Paul Ekman and Wallace V. Friesen [40]. They identified six basic emotions based on studying the isolated culture of people from the Fori tribe in Papua New Guinea in 1972. The tribe members were able to identify these six emotions on the pictures: Anger, Disgust, Fear, Happiness, Sadness, Surprise.

To represent each of the emotions, we will assign to each one a different color, and we will base the decision on the Wheel of Plutchik [41]. Robert Plutchik's three-dimensional model describes the relations among emotions, which is extremely helpful in understanding how complex emotions interact and change over time.

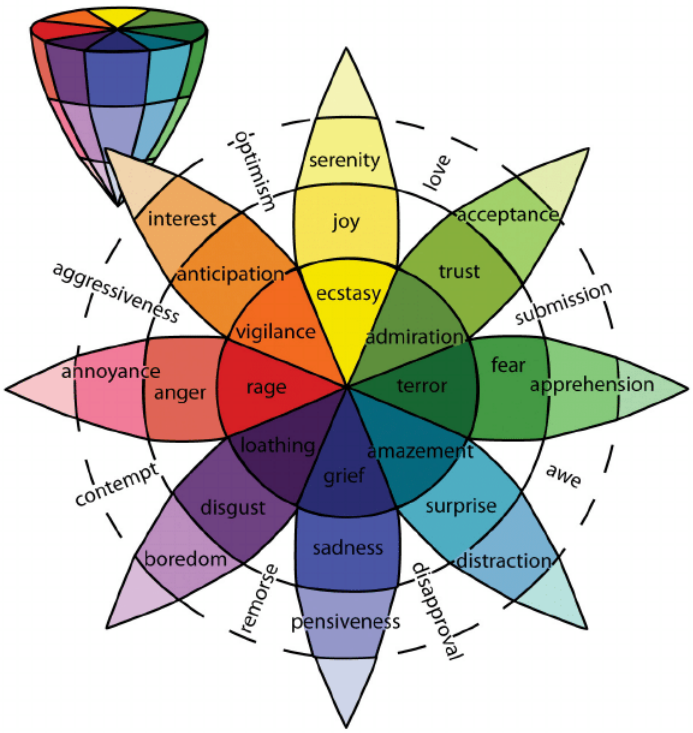

Figure 3.15: Wheel of Plutchik

If we try to break down the wheel of colors showed in Figure 3.15:

- *Primary:* The eight sectors are designed to indicate that there are eight primary emotion dimensions. They are anger, anticipation, joy, trust, fear, surprise, sadness and disgust.
- *Intensity:* The cone's vertical dimension represents intensity – emotions intensify as they move from the outside to the center of the wheel. For example, a feeling of boredom can intensify to loathing if left unchecked. This is an important rule about emotions to be aware of in relationships: If left unchecked, emotions can intensify. Herein lies the wisdom of enhancing your emotional vocabulary: it's the bedrock of effectively navigating emotions.
- *Relations:* Each circle sector has an opposite emotion. The opposite of sadness is joy, and the opposite of trust is disgust. The emotions with no color represent an emotion that is a mix of the 2 primary emotions. For example, anticipation and joy combine to be optimism. Joy and trust combine to be love. Emotions are often complex, and being able to recognize when a feeling is actually a combination of two or more distinct feelings is a helpful skill.

Figure 3.16: Sentiment colors

That is just a basic guide for interpreting the emotion wheel created by psychologist Robert Plutchik. And our emotions will end up styled following the Wheel of Plutchick, as in Figure 3.15

For the other color styling configuration, that we will use on the sentiment analysis, we will style the sentences in three different colors (Figure 3.16):

CHAPTER 4

Case study

4.1 Introduction

In this chapter, the functionalities created and described on the previous chapters are shown with different examples of actual use. The case studied is the general use of the Chrome plugin, which provides a graphic visualization in a chart and in the text that it is on the web page.

In the user interface, there will be a splash screen with some basic instructions on how to use the extension. We will find the selected service, that the user can change in the Configuration page, and the service comes with a indicator: when its in green color, it indicates the availability for the styling sentences functionality, explained in Subsec.3.5.2. Alongside this, some metrics will be shown in the form of a chart when the analysis is done.

In the following sections, examples of actual use of the Chrome extension will be provided, next to different demonstrations of the charts functionalities and the color styling.

4.2 Download and configuration

In this section, there is going to be a step-by-step guide on how the user can get the extension, how does it work and the different uses it has.

To get the extension, the user needs to go to the Chrome Web Store at `https://chrome.google.com/webstore/category/extensions?`. Next step is to search for our extension. In the Chrome Web Store search bar, the user can either: search for the developer's name *Grupo de Sistemas Inteligentes (GSI-UPM)* or search the name of the extension *Sentiment and Emotion Analysis Plugin*. Then the user will just have to click on `Add to Chrome` and the extension will be downloaded in his browser. And the last step for the user will be to pin the extension to the Chrome toolbar.

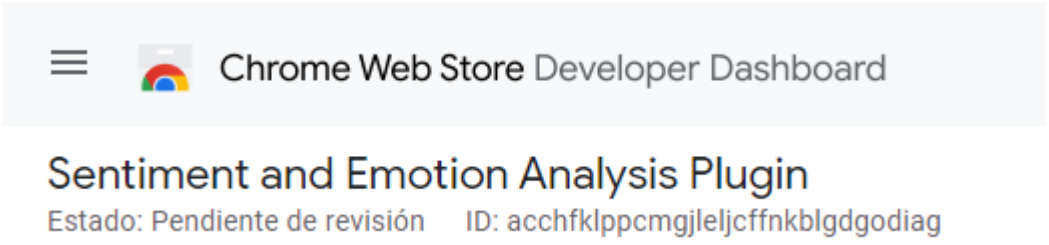

Figure 4.1: Chrome Extension Upload state

In Figure 4.1 we can see that currently the extension isn't available in the Chrome Store as it is going under revision. This process can take time, sometimes even weeks.

4.3 Sentiment Analysis

In this first use case, a user will perform a sentiment analysis of an article found in the famous newspaper "The New York Times" (`nytimes.com`).

The first thing the user needs to do is to choose a sentiment service analysis that fits its needs. In the configuration page, the user will have a brief explanation about what each service does and how do they work.

The Figure 4.2, shows the interface the user will see when clicking in the configuration icon. In the image, we can see the most important parts of the interface in red boxes (painted just for this explanation).

- The top red box represents the type of analysis that is currently selected and it appears

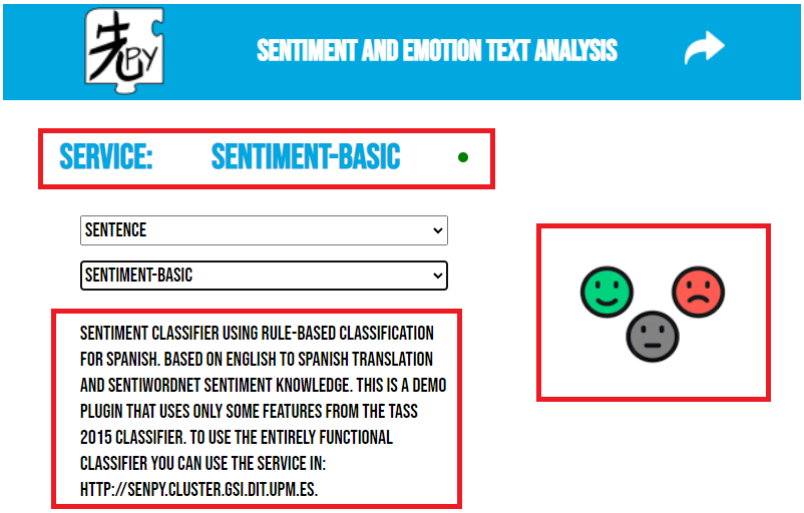

Figure 4.2: Configuration page example

with an indicator for the color styled sentences mode, as explained in Subsec. 3.5.2.

- The red box that appears on the right side of the interface, shows an easy guide for how the color styled sentences mode works. In this example, when a sentence appears in green it means it is positive, if its red, it means it is negative, and if the sentence appears in grey, the sentence sentiment is neutral.
- The last red box represent a brief explanation of the type of analysis that has been selected by the user.

Once the user has chosen a service, the next step is to open the article that he wants to analyse. In this case, we will be using the following article: `https://www.nytimes.com/2022/05/21/opinion/putin-nuclear-weapons.html`.

The user will have to manually select the text that he wants to be analysed and then click the button of the splash screen. Just for this example, we will use the first paragraph of the article:

The result that appears on Figure 4.9 will appear on the Splash Screen. We can see that the result is *Negative*, but this is because in the chart we will just get the overall sentiment of the article. When checking the Figure 4.10, each sentence will have a different color, representing a different sentiment.

Figure 4.3: Article.

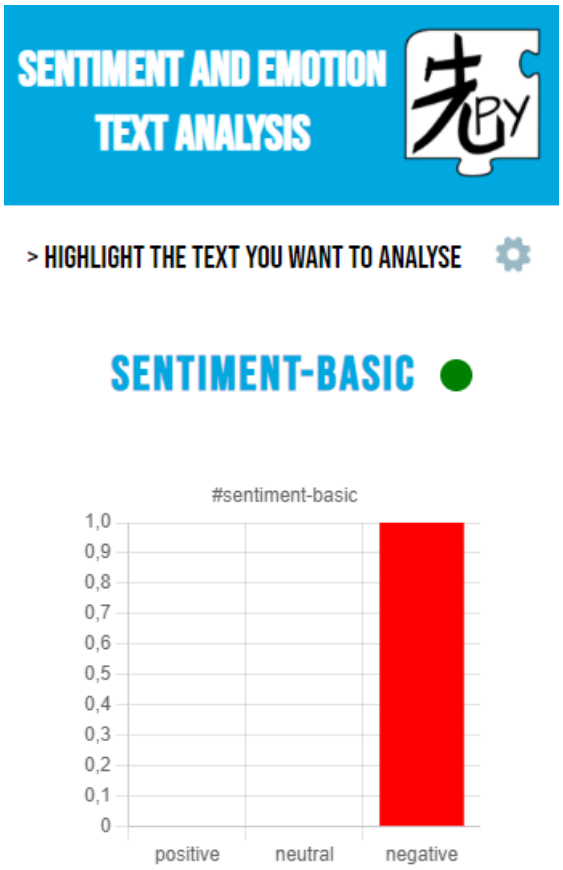

Figure 4.4: Chart result on the Splash Screen.

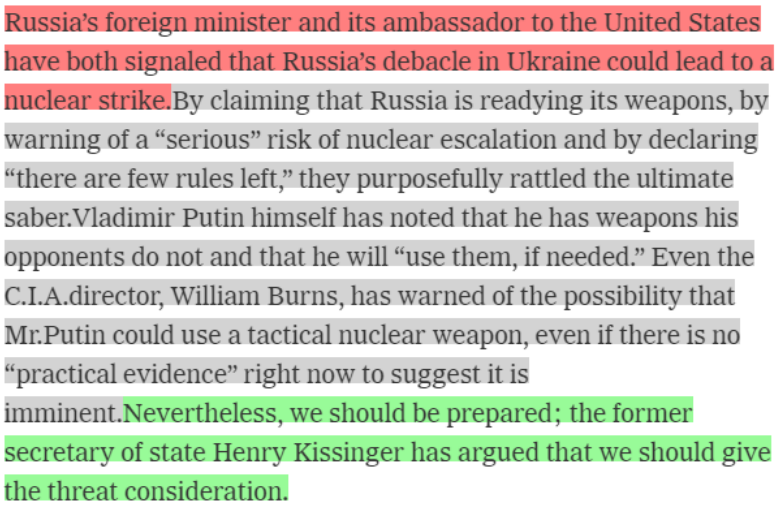

Figure 4.5: Article with the color styled sentences.

4.4 Emotion Analysis

In this use case, a user will perform an emotion analysis of a text. Emotionally rich texts can be found on product reviews, personal blogs/journals, social network websites, forums, fiction excerpts, analysis, critiques, and more. Although many databases focus on reviews of a product or a type of service, sources of text may come from news articles, stock market analyses, or political debates; anywhere that people discuss and share their opinion freely could be a source.

The steps to make an emotion analysis are pretty similar to the ones for the sentiment analysis. The first thing the user needs to do is to choose an emotion service analysis that fits its needs. In the configuration page, the user will have a brief explanation about what each service does and how do they work.

The Figure 4.6, shows the interface the user will see when clicking in the configuration icon. In the image, we can see the most important parts of the interface in red boxes (painted just for this explanation).

- The top red box represents the type of analysis that is currently selected and it appears with an indicator for the color styled sentences mode, as explained in 3.5.2.

- The red box that appears on the right side of the interface, shows an easy guide for how the color styled sentences mode works. The image represents the Wheel of Plutchick [41], we can see it more in detail in Figure 3.15. The wheel shows each

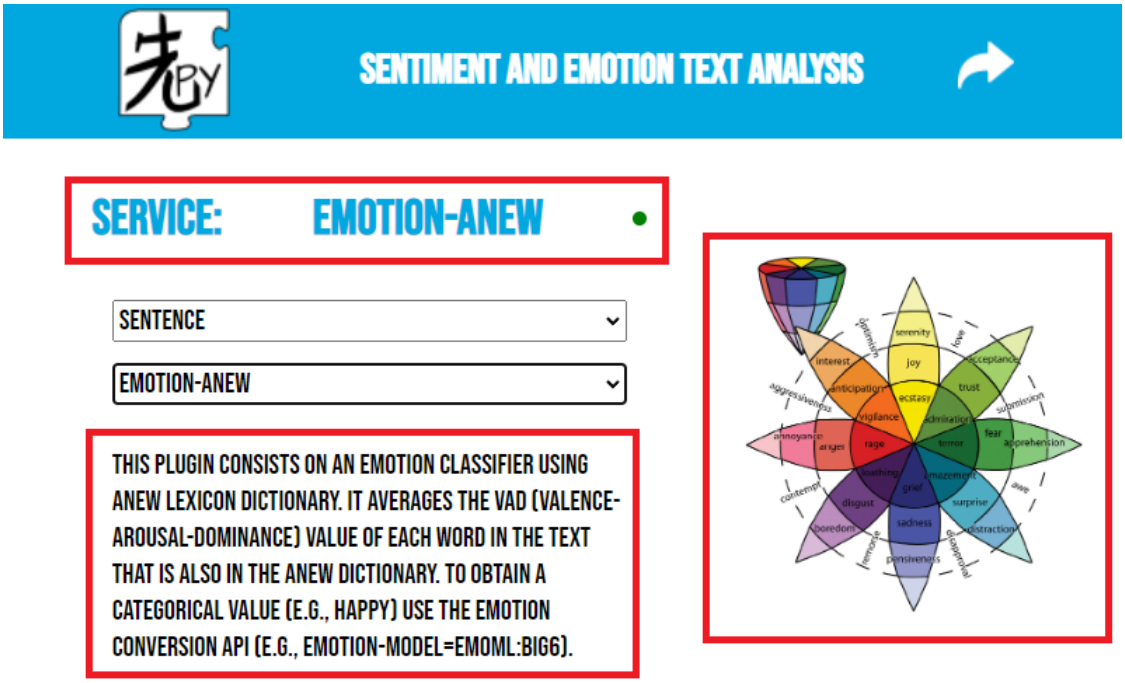

Figure 4.6: Configuration page example

emotion assigned with a color.

- The last red box represent a brief explanation of the type of analysis that has been selected by the user.

Once the user has chosen a service, the user has to select a text to be analysed, in this case, it will be an excerpt of "Harry Potter and The Sorcerer's Stone".

The user will have to manually select the text that he wants to be analysed and then click the button of the splash screen. Just for this example, we will use the any paragraph of the text:

The result that appears on Figure 4.9 will appear on the Splash Screen. The chart shows us three values, it averages the VAD (valence-arousal-dominance) value of each word in the text that is also in the ANEW dictionary. When checking the Figure 4.10, each sentence will have a different color, representing a different emotion from the "Six Basic Emotions" [40], explained in 3.5.2.

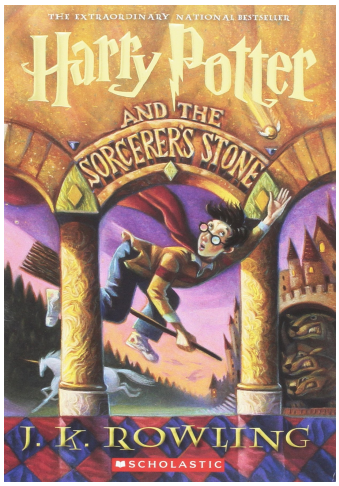

Figure 4.7: Harry Potter and The Sorcerer's Stone cover.

The zoo director himself made Aunt Petunia a cup of strong, sweet tea while he apologized over and over again. Piers and Dudley could only gibber. As far as Harry had seen, the snake hadn't done anything except snap playfully at their heels as it passed, but by the time they were all back in Uncle Vernon's car, Dudley was telling them how it had nearly bitten off his leg, while Piers was swearing it had tried to squeeze him to death. But worst of all, for Harry at least, was Piers calming down enough to say, "Harry was talking to it, weren't you, Harry?"

Uncle Vernon waited until Piers was safely out of the house before starting on Harry. He was so angry he could hardly speak. He managed to say, "Go - cupboard - stay - no meals," before he collapsed into a chair, and Aunt Petunia had to run and get him a large brandy. Harry lay in his dark cupboard much later, wishing he had a watch. He didn't know what time it was and he couldn't be sure the Dursleys were asleep yet. Until they were, he couldn't risk sneaking to the kitchen for some food.

Figure 4.8: Excerpt of the text.

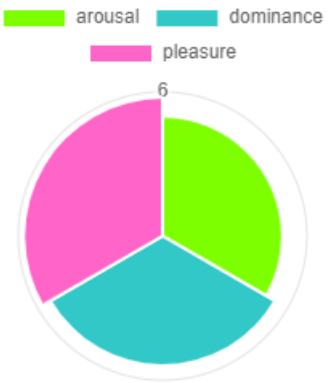

Figure 4.9: Chart result on the Splash Screen.

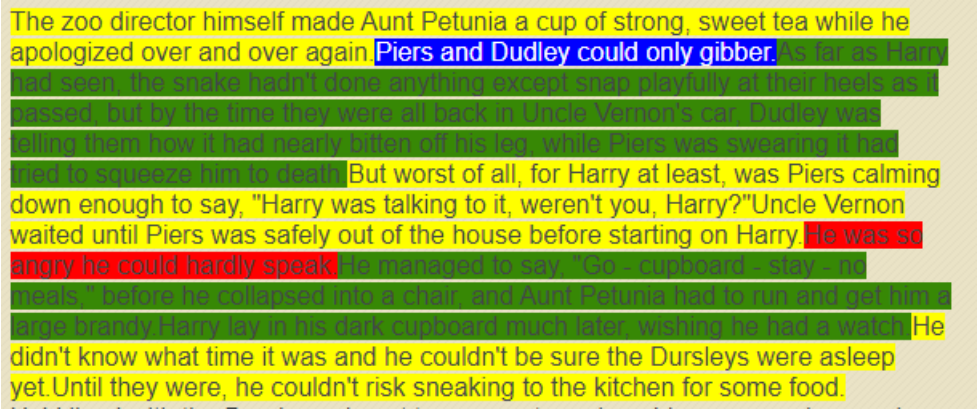

Figure 4.10: Excerpt with the color styled sentences.

4.5 Propaganda Detection

In this use case, the user will perform a propaganda detection. To use this service, we need to download from the Chrome Store the *Participation Extremism and Radicalization plugin*. To make use of it's services, we are going to use the `http://web.archive.org/`. This web page is a digital archive of the World Wide Web. Users can enter a URL to view and interact with past versions of any website contained in the Archive, even if the site no longer exists on the "live" web. For this example we will use the statement: "A Translation of the Statement of AQAP :: A Statement about the Martyrdom of Sheikh Anwar Al-Awlaqi and His Comrades".

To start the detection, the user needs to manually select the text he wants to analyse and then press the `Start Detection!` button.

This service will color style the sentences that are detected as propaganda in red.

The *Participation Extremism and Radicalization plugin* offers more services with different possibilities, but the propaganda service will be the main use case usually.

Figure 4.11: Front of the Islamic web page.

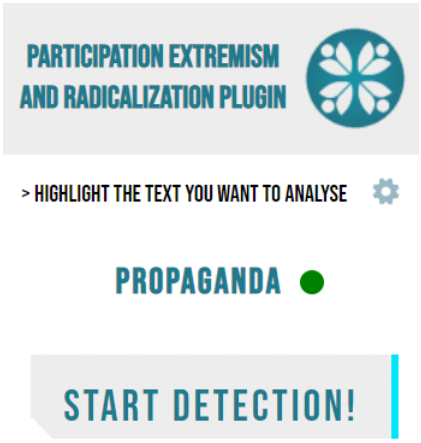

Figure 4.12: Splash Screen of the Participation Extremism and Radicalization plugin.

All praise be to Allah, the Lord of all that exists, and may peace and blessings be upon our Messenger, Muhammad, and his family.To proceed: We affirm to our Mujahid Ummah which has risen against injustice the martyrdom of the heroic Mujahid, Sheikh Abu Abdur-Rahman Anwar bin Nasir al-Awlaqi.His martyrdom as well as that of his comrades, Abu Muhsin al-Ma'ribi, Sameer Khan and Salim al-Marwani, may Allah have mercy on them all, was a result of an American airstrike in an area between Ma'rib and Jawf.By Allah's Grace, the Sheikh and Islamic preacher was killed while he remained firm in the path of Da'wah and Jihad, while both global and local pressure were unable to cause him to retract from it, despite lengthy attempts of seduction and threat, imprisonment and pursuit.This Islamic preacher is the one who achieved victory, by his beliefs, as remaining firm on beliefs is the true victory.He died upon his beliefs, as we reckon.This is a characteristic of the Muslim Ummah; their leaders do not die on their beds.The honor and the victory of the Ummah is that their leaders and preachers are martyred.Has any true call or the nation ever lived without sacrifices?If not, then we are an Ummah which never dies!Rather, it is victorious, it lives, and it is persists, because death gives it life!The Americans assassinated the Islamic preacher, Sheikh Anwar Al-Awlaqi and Sameer Khan without any brought proof of allegation, and they presented no evidence according to their own criminal laws of “freedom”.Where are their calls of freedom, justice, and human rights and freedom which they claim?America was so harmed by them that they contradicted their own principle, which they do daily, on which they claim their country was founded.America is the one who failed, as they did not abide by their principles, while the Sheikh proved victorious, as he lived according to his beliefs and died upon them.And thus does America commit acts of injustice and aggression, killing on a daily basis.Indeed they have a dark history in the likes of these crimes, too lengthy to be recorded.They fabricate brazen lies in public, claiming to be the protectors of human rights, freedom and justice.Their mendacious king makes a public appearance, stating that the death of Sheikh Anwar al-Awlaqi is a victory for the Yemeni people, because the Sheikh, according to him, killed many Yemeni people.But Obama, the yo-yo who changes his religion daily, that he was found confessing to the fact that he killed hundreds of the Sheikh's fellow tribesmen in Ma'jalah and Rafadh, most of them women, children and the elderly, with the blessings and participation of the butcher of Yemen, Ali Saleh.The Sheikh has proceeded to the Everlasting Gardens, Allah willing, while leaving behind for the Ummah a clear and pure ideology, a distinguished style, and a path of Da'wah and Jihad which the Ummah can adopt in their march towards honor and victory.Yes, America has killed Sheikh Anwar, may Allah have mercy on him, but they can never kill his ideology.Rather, the martyrdom of the Sheikh is a new and revived life of his ideology and style.The Sheikh has students which he taught, and students which have benefited from him in various parts of the world.They will march his march, following his tracks, continuing on the straight path, the path of the Prophet sallallahu alaihi wa sallam.As for American politicians, people have become accustomed to the fact that whenever they kill a leader of Islam, or whenever one of its heroes is martyred, they start to imagine things, saying that they have finished off Islam or the Muslim Ummah, that the Muslim Ummah is now impotent, or on that day they have achieved a great victory by killing such-and-such Muslim.Indeed it is only their imagination, as they are fighting an entire nation and a magnificent religion.They are fighting the Nation of True Monotheism (Taweed) and Jihad.How on earth can they think to break its resolve?!These patchy techniques, as well as the injustices of America against the entire world, and its continual support of the Jews and the idol-tyrant rulers, its proxies in various regions, have led Muslim masses to revolt and step into the fields, in outright refusal of the idiotic policies of America, their injustices against the entire world, and their support of their proxy rulers who they have placed over their neck.By Allah's Grace, America was slapped in Egypt when the revolutionaries refused to meet the U.S.Secretary of State.Then, they were given another slap in Libya, the land of Jihad.Now, they can only stand and watch the Muslim people of Syria, unable to do anything about the uprising in Yemen, who condemn and object to the ways of America, refusing to accept American interference, all of course except the opposition party, who by Allah's Grace do not represent the majority, nor do they have any strength on the ground.Our Yemeni revolutionaries!Indeed you left your homes in pursuit of freedom, justice, and in refusal of U.S.interference.You must state your firm decision about these flagrant and shameless transgression of airspace and these continual airstrikes against the best of your people, as well as the helpless from your women, children and the elderly.You must shout in the face of all those who have been appointed by America, just as you now shout in the face of the agent Ali Saleh, as by Allah they are two sides of the same coin.Are our calamities due to anyone other than America and their proxies?Ali Abdullah Saleh has taken America as a lord besides Allah.It is they who support him and grant him legitimacy, standing by his side for now thirty years!Our Yemeni revolutionaries!Indeed you must make your liberation from American slavery, the refusal of American plans for the region, and support for the people of Iraq, Afghanistan and Palestine against American occupation goals of your revolt as well.Our Yemeni revolutionaries!You have brought on a bright dawn against the proxy agent Ali Abdullah Saleh.You must bring on one more dawn against American interference in the region.In conclusion, we ask that Allah grants us and you the best of consolation, O tribe of Awaliq, for the assassination of your son, the magnificent, noble, brave and heroic, unjustly killed Sheikh Anwar bin Nasir al-Awlaqi.While just yesterday, the Mujahid Sheikh, Muhammad bin Umair al-Kalawi al-Awlaqi was killed along with a large number of your women, children and elderly.The regime of Ali Saleh killed the Mujahid Sheikh Zayid ad-Daghari, which is an open sign of them making these defiant and freeborn tribes a target due to their refusal to accept oppression, injustice and ignobility.O defiant tribes!Indeed the blood of the Sheikh and his comrades will not be spilled in vain!Indeed behind him there are heros which will not turn their eyes to injustice, in soon, Allah willing, they shall exact revenge!Tell the tribe of Awaliq that the best of consolation are the raves of the proud which cannot be muffled.We are at war with America, they inflict harms upon us as we do upon them, and the final end is for those who patiently persevere, and it is they who will prove victorious

Figure 4.13: Excerpt of the statement.

CHAPTER 5

Conclusions and future work

5.1 Introduction

In this chapter we will describe the conclusions obtained with the project completion. We will resume the principal concepts explained in the memory, the achieved goals, and finally a brief discussion about future work for the long run.

5.2 Conclusions

With the project already finished, we can take a look back and resume the principal concepts that are explained in this document. The goal of the project has been to develop a Chrome extension to let any user analyse any text on any web page. To accomplish this, the user will give some inputs to the browser extension and then communicate with the Senpy API, so that the extension can process the results and show them visually.

5.3 Achieved goals

The achieved goals obtained developing this project are:

Create a plugin to analyze text from the current web page the user is at

One of the main goals of the project was to make an easier way for the users to get text to be analysed. To achieve this, the easiest solution was to let the user select the text that it wants to be analysed. The user just needs to select manually the text and the click the button on the extension.

Give comprehensive visualization of the results

Another important goal to accomplish in this project was to give an easy visualization of the data results. We have developed two different solutions: a chart, with multiple choices and super simple parameters; and the possibility to get the text's sentences color styled depending on the analysis results. Both solutions sere the purpose of giving the user an easy and comprehensive visualization of the results.

Develop a page so the user can customize the parameters of the analysis

This goal has been accomplished partially. We have created a configuration page, where the user can change a few parameters to customize the experience and get the most out of it. But the goal was to add a lot more customization. There is no problem, as the configuration page is easily expandable and we can add more customization in the future.

Evaluate and distribute the plugin

This goal has also been accomplished, as the extensions both appear now in the Chrome Web Store, and any one that has a Chrome account can download the extensions and use them.

5.4 Future work

In this last section we will discuss about the possibilities for the extension in the future and all the features that can be added to improve, that were not implemented into this project because of the time limitation.

- Implement more parameters to the configuration page. It would be great if the user

could chose and change almost every variable: the different charts, the parameters and result that are showed, the colors, the theme for the extension, etc..

- Add a history, giving the user the possibility to go back to past analysis and see the results again.
- Develop a way to share the results of the analysis via mail or different social media platforms.
- Adapt the extension for the mobile devices: iOS and Android mainly.
- Increase the amount of services and types of analysis available.

APPENDIX A

Impact of this project

In this appendix we will analyze the different impacts in relation to this project, including social, economic, environmental and ethical impacts.

A.1 Social impact

This projects concerns the problems that some messages shared through Internet can reach a huge audience and can have a totally unexpected reaction from the users reading those texts. This is the reason why implementing text sentiments or emotion analysis for users could improve the overall user experience for them in these networks. The main objective of this project is to provide users an easy way to analyze any text, to know the real meaning of those texts. In this way, users can build a better virtual experience interacting with other people and extracting more information.

A.2 Economic impact

To discuss the economic impact, we can think of an example situation, for example, the damage to the public image of a person or business that an offensive message can generate. It could make a business loose a big percentage of its customers, it could lead to the firing of an experienced worker or it could make you loose friends and affect you psychologically. These kind of applications can contribute to the mitigation of this problem, allowing users to better analyze their messages before sending them and to avoid the possible economic impact that it could entail.

A.3 Environmental impact

To study the environmental impact we have to take a look to the equipment required to have internet access. This devices require electricity, which needs to be generated, in many cases in extreme polluting ways. Another drawback is the process and materials needed to build these devices, requiring mining heavy metals, as well as the incorrect disposal of the devices when they end their life cycle. We have mentioned the effect of making electronic devices, but for internet to work, it is necessary to have internet provider centers and data centers were information is stored. These have a great environmental impact, as they are buildings full of computers and servers, running uninterrupted.

A.4 Ethical impact

On the ethical impact we can see that text analysis tools can only help people with the heavy task of manually reading texts and classifying them. These tasks speed up the decision making processes, getting the necessary information quicker and with less human effort. Another benefit from text analysis tools is that they allow us to have more time to perform more important tasks.

APPENDIX B

Economic budget

In the appendix we will analyse the economic budget in relation to this project, including: Project structure, physical resources, human resources and taxes.

B.1 Physical resources

This section details the cost of the resources that allowed us to develop this project. For the operating system, we have used Windows 10 Home, whose license costs around 139 $.

The hardware in which the operating system runs is the following:

- CPU: Intel i7-7700K 4-Core Processor.
- GPU: GEFORCE GTX 1080 Ti.
- RAM: 16 GB DDR4 2133 MHz.
- Storage: 1 TB HDD and 120 GB SSD.

The estimated cost of the hardware is around 750 $.

B.2 Project Structure

The project was planned with a number of different tasks in mind, having the days structured as it is described in the text below:

Activity	Days
Researching about the topic	22
Learning Technologies Required	28
Looking for Possible Solutions	10
Planning the Structure	7
Developing the Application	34
Writing the Document	21
Total	122

B.3 Human resources

The budget required to cover the cost of human resources will be for one person, considering the project has been made individually. Estimating a salary of 1,500€monthly, that would be 50€per day. Knowing that the project took 122 days it would have a cost of 6,100€(without considering taxes).

B.4 Taxes

If the product was sold it would be subjected to the Value-Added Tax in Spain, that is 21 % of the product value.

Bibliography

[1] J Fernando Sánchez-Rada, Carlos A Iglesias, Ignacio Corcuera, and Oscar Araque. Senpy: A pragmatic linked sentiment analysis framework. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 735–742. IEEE, 2016.

[2] KeyCDN. Complete guide to cross-origin resource sharing (cors). `https://www.keycdn.com/support/cors`. Accessed: 2022-04-09.

[3] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[4] Christopher SG Khoo and Sathik Basha Johnkhan. Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. *Journal of Information Science*, 44(4):491–511, 2018.

[5] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

[6] Erik Cambria, Soujanya Poria, Rajiv Bajpai, and Björn Schuller. Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, pages 2666–2677, 2016.

[7] Finn Årup Nielsen. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*, 2011.

[8] Md Tajmilur Rahman, Peter C Rigby, and Emad Shihab. The modular and feature toggle architectures of google chrome. *Empirical Software Engineering*, 24(2):826–853, 2019.

[9] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 2011.

[10] Adam Lally and Paul Fodor. Natural language processing with prolog in the ibm watson system. *The Association for Logic Programming (ALP) Newsletter*, 9, 2011.

[11] Marcus Kracht. Introduction to linguistics. *Departement of*, 2007.

[12] Sebastian Hellmann, Jens Lehmann, and Sören Auer. Nif: An ontology-based and linked-data-aware nlp interchange format. *Working draft*, page 252, 2012.

[13] Carlos A Iglesias, J Fernando Sanchez-Rada, Gabriela Vulcu, and Paul Buitelaar. Linked data models for sentiment and emotion analysis in social networks. In *Sentiment Analysis in Social Networks*, pages 49–69. Elsevier, 2017.

[14] Paul Buitelaar, Mihael Arcan, Carlos A Iglesias, J Fernando Sánchez-Rada, and Carlo Strapparava. Linguistic linked data for sentiment analysis. In *Proceedings of the 2nd Workshop on Linked Data in Linguistics (LDL-2013): Representing and linking lexicons, terminologies and other language data*, pages 1–8, 2013.

[15] Peter Menke, Basil Ell, and Philipp Cimiano. On the origin of annotations: A module-based approach to representing annotations in the natural language processing interchange format (nif). *Applied Ontology*, 12(2):131–155, 2017.

[16] J Fernando Sánchez-Rada and Carlos A Iglesias. Onyx: A linked data approach to emotion representation. *Information Processing & Management*, 52(1):99–114, 2016.

[17] Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta, and Harshit Surana. *Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems*. O'Reilly Media, 2020.

[18] Juan Diego Gauchat. *El gran libro de HTML5, CSS3 y Javascript*. Marcombo, 2012.

[19] Helder Da Rocha. *Learn Chart. js: Create interactive visualizations for the web with chart. js 2*. Packt Publishing Ltd, 2019.

[20] Carson Sievert. *Interactive web-based data visualization with R, plotly, and shiny*. CRC Press, 2020.

[21] Adrián González Martín. Adaptación de grafos reutilizables d3 para nvd3. 2017.

[22] Elijah Meeks. *D3. js in Action*. Manning Shelter Island, NY, 2015.

[23] Joseph Kuan. *Learning Highcharts*. Packt Publishing Ltd, 2012.

[24] James Bucanek. Model-view-controller pattern. *Learn Objective-C for Java Developers*, pages 353–402, 2009.

[25] Tegar Satrio Utomo, Riyanarto Sarno, et al. Emotion label from anew dataset for searching best definition from wordnet. In *2018 International Seminar on Application for Technology of Information and Communication*, pages 249–252. IEEE, 2018.

[26] Saif Mohammad. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 174–184, 2018.

[27] Oscar Araque, Lorenzo Gatti, Jacopo Staiano, and Marco Guerini. Depechemood++: a bilingual emotion lexicon built through simple yet powerful techniques. *arXiv preprint arXiv:1810.03660*, 2018.

[28] Carlo Strapparava, Alessandro Valitutti, et al. Wordnet affect: an affective extension of wordnet. In *Lrec*, volume 4, page 40. Lisbon, Portugal, 2004.

[29] Oscar Araque, Ignacio Corcuera, Constantino Román, Carlos Angel Iglesias, and J Fernando Sánchez-Rada. Aspect based sentiment analysis of spanish tweets. In *TASS@ SEPLN*, pages 29–34, 2015.

[30] Clayton Hutto and Eric Gilbert. 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*, volume 8, pages 216–225, 2014.

[31] Adam Westerski J. Fernando Sánchez-Rada. Marl ontology specification. `https://www.gsi.upm.es/ontologies/marl/`. Accessed: 2022-04-21.

[32] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

[33] J. Fernando Sánchez-Rada Guillermo García-Grao. Ontologies for the participation project. `http://www.gsi.upm.es:9080/ontologies/participation/`. Accessed: 2022-06-10.

[34] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of liwc2015. Technical report, 2015.

[35] Jesse Graham, Jonathan Haidt, Matt Motyl, Peter Meindl, Carol Iskiwitch, and Marlon Mooijman. Moral foundations theory. *Atlas of moral psychology*, 211, 2018.

[36] Sebastiao Pais, I Tanoli, Miguel Albardeiro, and Joao Cordeiro. A lexicon based approach to detect extreme sentiments. In *Proceedings of the Fifteenth International Conference on Internet Monitoring and Protection*, 2020.

[37] Isabelle van der Vegt, Maximilian Mozes, Bennett Kleinberg, and Paul Gill. The grievance dictionary: understanding threatening language use. *Behavior research methods*, 53(5):2105–2119, 2021.

[38] Elisa Bassignana, Valerio Basile, and Viviana Patti. Hurtlex: A multilingual lexicon of words to hurt. In *Proceedings of the 5th Italian Conference on Computational Linguistics*, pages 1–6, 2018.

[39] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515, 2017.

[40] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124, 1971.

[41] Robert Plutchik. Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984:197–219, 1984.

[42] Oscar Araque. Design and Implementation of an Event Rules Web Editor. Trabajo fin de grado, Universidad Politécnica de Madrid, ETSI Telecomunicación, July 2014.

[43] J. Fernando Sánchez-Rada. Design and Implementation of an Agent Architecture Based on Web Hooks. Master's thesis, ETSIT-UPM, 2012.

[44] Paul Ekman. Facial expression and emotion. *American psychologist*, 48(4):384, 1993.

[45] Jacopo Staiano and Marco Guerini. Depechemood: a lexicon for emotion analysis from crowd-annotated news. *arXiv preprint arXiv:1405.1605*, 2014.