ESTG

INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

GENERATION OF BREAKING NEWS CONTENTS USING LLM (LARGE LANGUAGE MODELS) AND SEO (SEARCH ENGINE OPTIMIZATION)

GERAÇÃO DE CONTEÚDOS NOTICIOSOS QUENTES COM UTILIZAÇÃO DE INTELIGÊNCIA ARTIFICIAL (LARGE LANGUAGE MODELS) E SEO (SEARCH ENGINE OPTIMIZATION)

João Paulo Machado Pereira

GENERATION OF BREAKING NEWS CONTENTS USING LLM (LARGE LANGUAGE MODELS) AND SEO (SEARCH ENGINE OPTIMIZATION)

2024

Escola Superior de Tecnologia e Gestão

INSTITUTO POLITÉCNICO DE VIANA DO CASTELO

Generation of Breaking News Contents using LLM (Large Language Models) and SEO (Search Engine Optimization)

Geração de Conteúdos Noticiosos Quentes comutilização de Inteligência Artificial (Large Language Models) and SEO (Search Engine Optimization)

João Paulo Machado Pereira

MSC PROJECT WORK

Mestrado em Engenharia informática
Escola Superior de Tecnologia e Gestão

Orientador: Doutor António Miguel Rosado da Cruz
Co-adjuvado por: Mestre Wenderson Wanzeller

December 18, 2024

Abstract

With the rapid advance of digital technologies, verifying the authenticity of news has become increasingly difficult. The proliferation of social media platforms and websites has facilitated the spread of misinformation and biased content, undermining the traditional role of news as a source of unbiased information. The fake news phenomenon, amplified by platforms such as Facebook, WhatsApp and Twitter, has demonstrated its ability to influence public opinion by taking advantage of the viral nature of social sharing.

This thesis aims to address these challenges by proposing a solution to support the creation of credible news content quickly, impartially and effectively. The aim is to harness Artificial Intelligence (AI) and Search Engine Optimisation (SEO) technologies to develop a news platform capable of quickly producing comprehensive articles on trending topics. Using AI, the platform seeks to aggregate and analyse information from multiple reliable sources, ensuring that the news content is factual and unbiased.

The main components of the proposed platform include real-time monitoring of hot topics using tools such as Google Trends, which identify popular searches related to current events. These trending topics serve as input for the AI-powered system, which retrieves and synthesises information from reliable news sources. The reliable sources are selected by the user from a list of available sources (online newspapers and news agencies, etc.). Using generative AI, based on a Large Language Model (LLM), the platform generates concise and informative news summaries, presenting a complete picture of the topic.

User empowerment is fundamental to the platform's design. Users have the autonomy to select, modify and approve the articles generated by the AI, ensuring that the final result meets their editorial standards and is in line with the context in which it will be published. This process not only increases the user's credibility as a reliable source of information, but also promotes transparency in the dissemination of news content.

Through this innovative approach, the thesis aims to contribute to the evolution of digital journalism by demonstrating how AI and SEO technologies can be effectively integrated to combat disinformation and improve access to credible news sources. By enabling users to quickly disseminate factual and unbiased news in an era dominated by digital information overload, the platform represents a significant step towards restoring trust and reliability in online news consumption.

Resumo

Com o rápido avanço das tecnologias digitais, o acesso e a verificação da autenticidade das notícias tornaram-se cada vez mais difíceis. A proliferação de conteúdos de fácil acesso nas redes sociais facilitou a propagação de desinformação e de conteúdos tendenciosos, pondo em causa o papel tradicional das notícias enquanto informação imparcial e isenta. O fenómeno das notícias falsas, amplificado através de plataformas como o Facebook, o WhatsApp, o Twitter e outras, demonstrou a capacidade de influenciar a opinião pública, tirando partido da natureza viral da partilha social. Esta tese pretende abordar estes desafios, propondo uma solução para apoiar a criação de conteúdos noticiosos credíveis de forma rápida, imparcial e eficaz. O objetivo é tirar partido da Inteligência Artificial (IA) e das tecnologias de Search Engine Optimization (SEO) para desenvolver uma plataforma de notícias capaz de produzir rapidamente artigos noticiosos abrangentes sobre tópicos nas tendências. Ao tirar partido da IA, a plataforma procura agregar e analisar informações de várias fontes credíveis, garantindo que o conteúdo das notícias é factual e imparcial. Os principais componentes da plataforma proposta incluem a monitorização em tempo real de tópicos quentes utilizando ferramentas como o Google Trends, que identifica pesquisas populares relacionadas com eventos atuais. Estes tópicos em voga servem de entrada para o sistema de IA, que recupera e sintetiza informações de agências noticiosas fiáveis.

As fontes fiáveis são selecionadas pelo utilizador a partir de uma lista de fontes disponíveis. Utilizando Large Language Models (LLM), a plataforma gera resumos de notícias concisos e informativos, apresentando uma imagem completa do tópico e respeitando as normas jornalísticas de objetividade e precisão. A capacitação do utilizador é fundamental para a conceção da plataforma. Os utilizadores têm autonomia para selecionar, modificar e aprovar os artigos noticiosos gerados pelo LLM, garantindo que o resultado final cumpre as suas normas editoriais e está em conformidade com o contexto em que será publicado. Este processo não só aumenta a credibilidade do utilizador enquanto fonte fiável de informação, como também promove a transparência na divulgação de conteúdos noticiosos. Através desta abordagem inovadora, este projeto visa contribuir para a evolução do jornalismo digital, demonstrando como as tecnologias de IA e SEO podem ser eficazmente integradas para combater a desinformação e melhorar a acessibilidade a fontes de notícias credíveis. Ao permitir que os utilizadores divulguem rapidamente notícias factuais e imparciais numa era dominada pela sobrecarga de informação digital, a plataforma representa um passo significativo para restaurar a confiança e a fiabilidade no consumo de notícias.

Acknowledgements

I would like to express my sincere gratitude to my family, girlfriend and friends for their unconditional support throughout this academic endeavour. Their words of encouragement, patience and understanding were fundamental during the challenging moments of this process. I would especially like to thank you for your constant encouragement and emotional support, which enabled me to stay focused and determined to achieve my goals.

My deepest thanks go to my family for their unconditional love and understanding during the intense periods of study and research. I would also like to thank my girlfriend, for her constant presence, understanding and daily inspiration that brought light to the moments of academic pressure.

To my friends, for their true friendship, mutual support and moments of relaxation that were essential for my emotional and mental balance throughout this journey. Your conversations and laughter provided welcome relief in moments of intense concentration and hard work.

This work would not have been possible without the support and encouragement of those who have been by my side along the way. Each of you has played a fundamental role in my personal and academic growth, and for that I am eternally grateful.

João Pereira

"Those who can imagine anything,
can create the impossible."

Alan Turing

Contents

List of Figures

List of Tables

Accronyms

AI	Artificial Intelligence
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
AWS	Amazon Web Services
BERT	Bidirectional Encoder Representations from Transformers
CSRF	Cross-Site Request Forgery
CSS	Cascading Style Sheets
DOM	Document Object Model)
DSR	Design Science Research
GPT	Generative Pre-trained Transformer
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
LLM	Large Language Models
MVC	Model-View-Controller
MVT	Model View-Template
NLP	Natural Language Processing
ORM	Object Relational Mapper
SEO	Search Engine Optimization
SQL	Structured Query Language
T5	Text-to-Text Transfer Transformer
URL	Uniform Resource Locator
WWW	*World Wide Web*
XSS	Cross-site scripting

Chapter 1

Introduction

1.1 Contextualisation

With the advance of technologies in the digital world and the speed with which information reaches us, it has become very difficult to verify the authenticity of the news that reaches us. Any individual can access a social network or build a website and create fake news or news that are somehow false, partial and/or biased. News should, by definition, be totally impartial, not opinionated, and inform the reader effectively. Fake news are gaining more and more notoriety on social media, as links and "illustrative" images are easily spread on WhatsApp, Twitter, Facebook and others, and tend to influence the reader's opinion. A study by the Columbia Journalism Review [Nelson, 2017] states that 30 per cent of fake news are linked to Facebook, which allows the creation of a chain of message shares that lead the user to believe that it is spreading real news.

1.2 Objectives

The aim of this thesis is to support the creation of credible news content quickly, impartially and seriously. The goal is that, when a news topic is "hot" or trending, the user can quickly consult a news story with all the details, without blockers, and obtain a news text with the help of AI, from different sources catalogued as credible.

The aim is to create a news platform with the help of Artificial Intelligence and Search Engine Optimisation (SEO) technology [Google, 2024], so that it is optimised for search engines, i.e. to create a portal for generating news content with daily trending topics. The user will be able to use one of these news items to publish on their website or social network. The user has full control over the news item and can freely edit it to suit the required context. To give an example: With the resignation of the Portuguese Prime Minister, news trend monitoring platforms, such as Google Trends, will have words such as "António Costa", "Resignation António Costa", "Prime Minister", etc. in their trends. Initially, we will try to collect these trends, so that we can search for news stories with these trend-words in credible news sources, and then interact with the generative AI engine, using a Large Language Model [OpenAI, 2024]. The goal is to obtain a news text that

summarises the news content obtained from credible sources. From the texts collected, a complete and impartial news item is created, with all the details so that the reader is properly informed. With this whole process, the user will always be in charge of modifying and finalizing the news text, approving the news, being able to enter the generative news application and consult and approve, or not, the complete news with the latest hot topics, thus becoming a more reliable and impartial source due to the help of Artificial Intelligence.

1.3 Research methodology

The Design Science Research (DSR) methodology is an iterative problem-solving process common in areas such as information systems and computer science. This approach involves creating and evaluating innovative artefacts to deal with complex problems. The process includes several phases: 1 problem identification, 2 artefact design, 3 development, 4 demonstration or validation, 5 evaluation and 6 iterative refinement [Cruz and Rosado da Cruz, 2020].

In the problem identification phase, as shown in the image below 1.1, researchers thoroughly understand the context and scope of the problem. They then move on to the design and development of artefacts, which can range from models to software systems. This stage emphasises creativity and innovation, combining existing knowledge with innovative approaches to create effective solutions.

Once the artefact is created, it undergoes a demonstration or validation to show its functionality and potential impact. This involves implementing it in real or simulated environments and establishing evaluation criteria. The evaluation phase analyses the artefact's usefulness, usability and contribution to knowledge, focusing on both practical application and theoretical advancement.

Iterative refinement is a central aspect of DSR. Based on the results and feedback of the evaluation, researchers continuously improve the artefact to increase its relevance and effectiveness. Finally, the results and insights are communicated through publications and other channels, contributing to collective knowledge in the respective field.

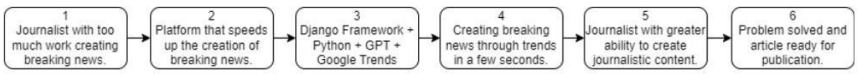

Figure 1.1: DSR activities flow within the project.

1.4 Structure of this report

In addition to the introduction, this dissertation contains five more chapters. In this chapter 1, the research topic is introduced, as well as its objective and its contextualisation is applied. Chapter 2 explains some concepts that are used along the dissertation, together with technologies that used, and also gives a theoretical explanation of some essential technologies.

In Chapter 3, the state of the art is described and related work is presented.

Chapter 4 will talk about all the conceptualisation of the project and all its development, such as the diagrams needed for the project and all the structure of the project or important parts of the code.

Finally, in chapter 5 discusses the difficulties involved in realising the project, as well as some features for future work and the conclusion of the project.

Chapter 2

Concepts and Technologies

This chapter will mention the technologies that are used in this dissertation. Some journalistic terms will be explained, as well as some of the project's core technologies.

2.1 Domain Concepts

2.1.1 News

A news story is a piece of information about recent, relevant and factually accurate events [Manual, 2024]. News are usually written by journalists and broadcast in media such as newspapers, news websites, radio or television. News stories aim to inform the public about what is happening in the world, in society or in a particular area, such as politics, economics, sports, among others. In addition, news stories are usually objective, impartial and based on verifiable facts. They are important for keeping the public informed about important events and play a crucial role in democracy by enabling people to form their own opinion and make informed decisions.

2.1.2 Newspaper

A newspaper is a publication that contains informational news, reports and analyses on current events and developments. Usually printed on paper or available digitally, a newspaper aims to inform the public about a wide range of topics, from local and national news to international news on sports, culture, economics, politics and much more states [of Encyclopædia Britannica, 2024b].

Newspapers play a vital role in society by serving as a reliable source of information and playing a key role in monitoring democracy by investigating and reporting on issues of public interest. They are an essential tool for keeping people informed about what is happening in the world..

2.1.3 Breaking News

Breaking news are a type of news stories that report recent and important events or developments that are taking place at the moment. The authors in [Rogers, 2024] consider, this type of news

as urgent and highly relevant, often interrupting the normal programming of media outlets such as TV news, news websites and social networks to inform the public immediately. In short, a breaking news is an extremely recent and significant piece of news that is released immediately to keep the public informed about events that are happening at the moment, often impacting society or the community in some way.

2.1.4 Fake News

Fake news refers to information content that has been deliberately fabricated or distorted and presented as if it were true. The concept ranges from completely invented information to the partial manipulation of facts or images, with the aim of deceiving, influencing public opinion, or achieving specific objectives, whether political, ideological, financial, or social [Aïmeur et al., 2023]. Fake news proliferates especially in the contemporary digital ecosystem, driven by the speed and reach of social networks, search engines, and instant communication platforms. This phenomenon is not new, but it has taken on new dimensions with the advent of the Internet and information and communication technologies. The ease of publishing, sharing, and viralizing content online, combined with the difficulty of verifying it in real time, has made fake news a threat to traditional journalism and the credibility of information sources. Furthermore, fake news often exploits strong emotions, such as fear, anger, or hope, to engage the public and encourage viral spread. The resulting disinformation can undermine trust in institutions, polarize societies, and damage democratic processes such as elections and public debates.

2.1.5 Journalist

A journalist is a professional dedicated to collecting, investigating, analysing and disseminating relevant information to the public [Hill, 2024]. This information gathering can involve research, interviews, observation and investigation to create news, reports or other informative content. Journalists can work in different media, including newspapers, magazines, radio, television and online media.

Journalists play a crucial role in society, providing the public with objective and accurate information about local, national and international events, issues and developments. In addition, journalists often adhere to ethical standards to ensure the integrity and impartiality of their work.

2.1.6 News Agency

A news agency is an organisation that collects, verifies and distributes information and news to media outlets such as newspapers, magazines, radio stations, television networks and news websites [of Encyclopædia Britannica, 2024a] states. These agencies play a key role in the media ecosystem, providing coverage of events, reports and stories for the media outlets that subscribe to their services.

News agencies usually have reporters and correspondents in various parts of the world to ensure comprehensive and international coverage. They can offer a wide range of news, from political and economic affairs to sports, entertainment and cultural events.

2.1.7 Opinion piece or article

An opinion piece is a textual genre in which the author presents and defends their own points of view, interpretations and conclusions on a specific subject [Signy, 2024]. In this type of text, subjectivity is valued, allowing the author to become emotionally involved with the topic and influence the reader with their point of view. By using arguments, examples and personal experiences, the author seeks to persuade, inform and stimulate reflection in the reader. Opinion pieces are an important form of individual expression, contributing to public dialogue and the diversity of perspectives on various platforms, from newspapers and magazines to digital media such as blogs. In addition, it is important to note that opinion pieces often challenge established norms, explore alternatives and encourage critical thinking. Authors often use persuasive language and stylistic resources to captivate the reader's attention and convey their message in an impactful way. These texts not only reflect the uniqueness of individual perspectives, but also play a significant role in shaping public opinion, influencing social, political and cultural debates. By encouraging the open expression of ideas, opinion pieces contribute to a more pluralistic society, where a variety of voices and points of view enrich the information landscape and promote mutual understanding.

Opinion pieces should be explicitly labeled as such to ensure they are not misinterpreted as news.

2.2 Tools e Technologies

Web development projects often involve the integration of various tools and technologies to achieve the desired goals. This report explores the main tools and technologies used in developing the solution proposed in this work, namelly a Django Python project aimed at generating breaking news. We will discuss the use of the Django framework, databases, APIs, and cloud services, highlighting their functionalities and benefits.

2.2.1 Chat-GPT

ChatGPT is a conversational system based on artificial intelligence developed by OpenAI. It uses the GPT (Generative Pre-trained Transformer) architecture to understand and generate natural language text. ChatGPT is trained on a wide variety of texts, allowing it to answer a wide variety of questions and perform various linguistic tasks.

Users can interact with ChatGPT by asking questions, requesting information, generating text, obtaining recommendations and much more.

It is important to note that ChatGPT responds based on language patterns learnt from its training data, and the information available is updated up to the date the model is trained (knowledge

cut-off) [Hetler, 2024]. Therefore, it may not have up-to-date information in real time and the accuracy of the answers may vary depending on the context and nature of the questions.

2.2.2 Google Trends

Google Trends is a free tool provided by Google that allows users to explore and analyse search trends on the internet. It offers information on the popularity of specific search terms over time and in different geographical regions. Google Trends is widely used by marketers, journalists, researchers and anyone interested in understanding web search behaviour and emerging trends [Pol, 2024]. It offers valuable insight into what is capturing the public's attention and helps them make informed decisions relating to content, advertising and much more.

2.2.3 Python

Python is a high-level programming language known for its simplicity and versatility. It was created by Guido van Rossum and launched in 1991. Its ease of use and vast support community make it a solid choice for programmers wishing to develop applications of all kinds. To summarise, Python is a versatile programming language that is easy to learn and widely adopted in a variety of domains due to its user-friendly syntax, large community and vast amount of available resources. [Fran, 2024] It is a solid choice for programmers of all levels of experience and can be used for a wide range of projects and applications.

2.2.4 Django - a Python Framework

Django is a high-level web development framework for the Python programming language, which promotes rapid development and clean, pragmatic design [Johnson, 2024]. First released in 2005, Django is designed to help developers create web applications quickly and efficiently, with a set of reusable components and a well-defined architecture. This framework includes a robust ORM (Object-Relational Mapping) that facilitates interaction with relational databases, allowing developers to manipulate data using Python classes instead of writing SQL directly [Johnson, 2024]. Also, the framework comes with an administrative interface generated automatically from the models defined, which allows you to manage application data without the need to code a customised administrative interface. Django is built with a modular architecture that allows components to be reused. This includes modules for authentication, session management, caching, internationalisation and more. Also Django was developed with security in mind, incorporating measures to prevent common vulnerabilities in web applications, such as Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), and SQL Injection. Django is scalable and can be used to create applications ranging from small personal projects to large enterprise systems. It supports various performance optimisation techniques and can be easily integrated with other technologies. Their architecture adopts the Model-View-Template (MVT) architecture pattern, which is a variation on the traditional Model-View-Controller (MVC). This pattern helps to separate business logic, user interface and data management, facilitating code maintenance and scalability.

- Model: Represents the data layer and business logic. It defines the structure of the database and interacts with it through the ORM.
- View: Deals with presentation logic. It receives user requests, processes the data using the models and returns a response.
- Template: Defines the user interface. It uses a template language to generate dynamic HTML that will be displayed to the user.

2.2.5 Python libraries

Python libraries are collections of modules and packages that provide additional functionality for the Python programming language. They allow developers to reuse pre-written and tested code to perform common tasks such as manipulating data, creating graphical interfaces, communicating with web servers, and much more. Using libraries helps speed up software development, improves code efficiency and reduces the likelihood of errors.

Below, is listed the structure of Python Libraries:

- Module: A single Python file (.py) that contains definitions of functions, classes and variables that can be reused in different programmes.
- Package: A collection of modules organised in a directory structure. Packages can contain subpackages, allowing for hierarchical organisation.

Python libraries allow developers to reuse existing functionality without the need to rewrite code from scratch. They also increase productivity by providing ready-made solutions for common tasks, allowing developers to concentrate on the specific aspects of their projects [Tuama, 2024]. Popular libraries are widely tested and used by large communities, reducing the likelihood of bugs and problems. Python libraries are essential components for efficient, high-quality software development. They offer a wide range of functionalities covering diverse areas, from scientific computing and data visualisation to web development and machine learning. By taking advantage of these libraries, developers can create robust and scalable applications more quickly and efficiently.

2.2.5.1 Beautiful Soup

Beautiful Soup is a Python library used to extract data from HTML and XML files. It creates a parse tree from the content of the files, allowing developers to navigate, search and modify the document content in an intuitive and efficient way [Sulcas, 2024]. Beautiful Soup is especially useful for web scraping tasks, where is wanted to collect information from web pages.

Beautiful Soup is known for its simplicity and ease of use, offering an intuitive API for browsing and manipulating HTML and XML documents. Although Beautiful Soup uses Python's HTML parser by default, it is also compatible with other parsers, such as lxml and html5lib, providing flexibility in choosing the parser best suited to the project's needs. The library allows to

navigate the analysis tree using tag names, attributes, text and other elements. This makes it easier to extract specific data from complex documents. Beautiful Soup allows you to modify HTML and XML documents, including adding, removing and changing elements and attributes.

Beautiful Soup allows data to be extracted from web pages quickly and with just a few lines of code. It can be used to extract data from HTML and XML documents of different formats and complexity. It also Works well with other Python libraries, such as Requests for making HTTP requests and Pandas for analysing data.

Common Applications of Beautiful Soup:

- Web Scraping: Extracting data from websites for analysis and further processing.
- Data Analysis: Collecting data from web sources for analysis and research purposes.
- Task Automation: Automating processes that involve browsing and collecting information from web pages.

Beautiful Soup is a powerful and easy-to-use tool for working with HTML and XML data in Python. Its ability to extract and manipulate data from web documents makes it a popular choice among developers and data scientists for web scraping and automation tasks.

2.2.5.2 Selenium WebDriver

Selenium WebDriver is a test automation tool for web browsers. It allows developers to write scripts in various programming languages, such as Python, Java, Ruby, and others, to control and interact with elements of a web page programmatically. Selenium WebDriver is widely used for automated testing of web applications, simulation of user interactions, and web scraping tasks that require the execution of JavaScript or dynamic interaction with the page [Unadkat, 2024]. Selenium WebDriver supports multiple browsers, including Google Chrome, Mozilla Firefox, Safari, Internet Explorer, Microsoft Edge, and Opera. This allows you to test web applications in different environments consistently. It also allows you to interact with page elements, such as clicking buttons, filling in forms, navigating between pages, and executing JavaScript scripts. Selenium WebDriver provides a rich API that lets you control the browser in detail, including operations such as navigating to URLs, capturing screenshots, and handling cookies. It can be integrated with testing frameworks such as JUnit, TestNG, pytest, and others, facilitating the execution of automated tests as part of the continuous development process. Selenium WebDriver can be configured to run in "headless" mode, allowing tests to be executed without the need to open a graphical browser interface.

Below, a list of benefits from Selenium WebDriver was compiled:

- Functional Test Automation: Facilitates the creation of automated tests that simulate real user interactions with the web application.
- Versatility: Supports different browsers and operating systems, allowing testing in multiple environments.

- Cost-Effective: It is an open source tool, which reduces the costs associated with test automation compared to commercial solutions.
- Integration with CI/CD: It can be easily integrated into Continuous Integration and Continuous Delivery (CI/CD) pipelines, enabling automated testing at every stage of development.
- Automated testing: Creation of test suites to verify the functionality of web applications repeatedly and reliably.
- Dynamic Web Scraping: Extracting data from web pages that require interaction or JavaScript execution, where other scraping tools may fail.
- Task Automation: Automating repetitive tasks in web browsers, such as filling in forms, navigating websites, and executing programmatic actions.

Selenium WebDriver is a powerful and flexible tool for test automation and interaction with web browsers. Its ability to control browsers programmatically, along with its support for multiple programming languages and browsers, makes it a popular choice for developers and testers looking to automate functional testing and web scraping tasks.

2.2.5.3 PyTrends

Pytrends is an unofficial API for Google Trends, written in Python. It allows developers to access and analyse Google search trend data programmatically. Pytrends makes it easy to extract information about the popularity of search terms over time, interest by region, related queries, and other metrics available in Google Trends. It is a valuable tool for data analysts, marketing professionals, researchers and developers who want to explore and utilise search trend data [Kung, 2024]. Pytrends enables the extraction of historical and real-time data on the popularity of search terms on Google. Iy also facilitates the analysis of search interest by different geographical regions, allowing comparisons between countries, states and cities. Pytrends provides data on related queries and topics that are growing in popularity. Also allows you to compare the popularity of multiple search terms simultaneously. It can be easily integrated into Python scripts to automate the collection and analysis of trend data. Below, a list of benefits from Pytrends was compiled:

- Search Insights: Provides valuable insights into Google search trends, helping to understand user behaviour and identify emerging patterns.
- Informed Decisions: Marketing and business professionals can use the data to make informed decisions about content strategies, SEO and advertising campaigns.
- Comparative Analysis: Makes it easy to compare the popularity of different search terms, helping to identify which topics are gaining or losing interest.
- Report Automation: Can be integrated into automation systems to generate periodic reports on search trends, saving time and resources.

- Market Analysis: Companies use Pytrends to monitor the popularity of products and services, identifying market opportunities.
- SEO and Content Marketing: SEO professionals use Pytrends to identify emerging keywords and adjust content strategies to maximise search engine visibility.
- Academic Research: Academic researchers can use trend data to study society's behaviours and interests in various areas.
- Competitor Monitoring: Companies can track trends related to competitors and adjust their strategies accordingly.

Pytrends is a powerful tool for accessing and analysing Google search trend data. Its ability to provide insights into the popularity of search terms over time and by region makes it a valuable choice for data analysts, marketing professionals, researchers and developers. By integrating Pytrends into their projects, users can explore trend data programmatically, allowing for more informed decisions and more effective strategies.

2.2.5.4 Open AI

The OpenAI Python Library is a collection of modules and functions developed by OpenAI to facilitate access to and interaction with OpenAI's artificial intelligence models, such as GPT-3, GPT-4 and others. This library allows developers to easily integrate advanced natural language processing (NLP) capabilities into their applications, using just a few lines of Python code. The library is designed to be simple to use, providing a clear and intuitive interface for sending requests to models and processing their responses. OpenAI Python Library provides an easy-to-use API for accessing OpenAI models, allowing developers to send prompts and receive responses without additional complexity [Karl, 2024]. It supports a variety of operations, including text generation, translation, summarisation, answering questions, code completion, and much more. Also allows you to configure parameters to control the behaviour of templates, such as temperature, number of tokens, and repetition penalties, ensuring responses that are more in line with user needs.The library comes with detailed documentation that includes examples of use, descriptions of functions and guides to solving common problems.It can be integrated with other Python libraries and frameworks, making it easy to include the capabilities of OpenAI models in different types of applications.

Below, a list of benefits from OpenAI Python Library was compiled:

- Easy Access: Simplifies access to OpenAI's advanced AI models, allowing developers to integrate NLP functionalities into their applications quickly and easily.
- Productivity: Reduces the time and effort needed to implement advanced AI capabilities, allowing developers to focus on other parts of development.
- Customisation: Offers flexibility in configuring parameters to adjust the behaviour of models, allowing for more appropriate responses to the desired context.

- Scalability: It can be used in different types of applications, from chatbots and virtual assistants to complex data analysis and content generation systems.
- Chatbots and Virtual Assistants: Development of systems that interact with users in a natural and effective way.
- Content Generation: Automatic creation of articles, blog posts, product descriptions and other types of textual content.
- Text Analysis: Processing and analysing large volumes of text to extract insights and patterns.
- Customer Support: Automating responses to common queries, improving the efficiency of customer service.
- Research and Education: Assisting with research activities, creating educational materials and explaining complex concepts in an accessible way.

The OpenAI Python library is a powerful and versatile tool for integrating advanced natural language processing capabilities into Python applications. Its simplicity of use, combined with the flexibility and power of OpenAI's models, makes it an ideal choice for developers looking to harness AI to create innovative and effective solutions.

2.2.5.5 Pandas

Pandas is a free software library for the Python programming language, widely used for manipulating and analysing data. Initially developed by Wes McKinney in 2008, Pandas provides fast, flexible and expressive data structures, such as DataFrame and Series, which are designed to make working with structured data (such as spreadsheets and database tables) simple and efficient. The library is essential for data scientists, analysts and engineers who work with large volumes of data and need to perform complex manipulation and analysis operations.

Pandas supports reading and writing data in various formats, including CSV, Excel, SQL, JSON and many others and facilitates data cleaning, transformation, restructuring and analysis, offering functions for filtering, grouping, aggregating and combining datasets [S, 2024].

It enables efficient data selection through indexing, slicing and filtering techniques and includes advanced functionalities for working with temporal data and performing time series operations.

Pandas works well with other data science and machine learning libraries in Python, such as NumPy, SciPy, Matplotlib and scikit-learn.

Below, a list of benefits from Pandas was compiled:

- Ease of Use: Simplifies data manipulation and analysis, allowing complex operations to be carried out with just a few lines of code.

- Efficiency: Highly optimised for performance, it can handle large volumes of data efficiently.
- Versatility: Offers a wide range of functionalities for different types of data operations, from reading and writing to cleaning and analysing.
- Community and Support: It has a large community of users and developers, with extensive documentation and resources available for learning and support.
- Data Analysis: Used to analyse large data sets and extract meaningful insights.
- Data Preparation: Cleans and transforms data before applying it to machine learning models.
- Data Visualisation: Facilitates the creation of graphs and visualisations to interpret and communicate data effectively.
- Research and Data Science: Widely adopted in research and data science projects for manipulating and analysing structured data.

Pandas is a fundamental library for manipulating and analysing data in Python. Its intuitive interface and powerful functionalities make the process of working with structured data more efficient and productive. Using Pandas, data scientists, analysts and engineers can perform complex data operations with ease, from reading and cleaning to analysis and visualisation, making it possible to extract valuable insights from large volumes of data.

2.2.5.6 Psycopg2

Psycopg2 is a PostgreSQL database adapter for the Python programming language. It is a popular and widely used library that allows developers to connect and interact with PostgreSQL databases efficiently and securely. Developed by Federico Di Gregorio, psycopg2 is known for its robustness, compliance with the DB-API 2.0 specifications and its ability to handle database operations in high-demand production environments.

Psycopg2 follows the Python Database API 2.0 specification, providing a standard interface for working with relational databases in Python and it´s optimised for performance, Psycopg2 is able to handle large volumes of data and complex operations efficiently. It Fully supports transactions, allowing the execution of multiple database operations in an atomic and secure manner.

Psycopg2 includes mechanisms to prevent SQL injection attacks and other security vulnerabilities common in database interactions and it has fully compatible with PostgreSQL's advanced features, including support for customised data types, arrays, and database functions. Psycopg2 provides support for asynchronous operations, allowing non-blocking execution of database operations in Python applications.

Below, a list of benefits from Psycopg2 from [Arif, 2024] was compiled:

- Simple Integration: Facilitates integration with PostgreSQL databases in Python applications.

- Robustness: Reliable and suitable for production use, supporting high-demand operations.
- Security: Includes functionalities to prevent SQL injections and other vulnerabilities.
- Flexibility: Supports PostgreSQL's advanced features, allowing the use of customised data types and other database-specific characteristics.
- Web Development: Used in web applications to interact with PostgreSQL databases, especially in frameworks such as Django and Flask.
- Data Analysis: Facilitates the extraction and manipulation of large volumes of data stored in PostgreSQL databases.
- Business Applications: Supports the backend of business information management systems, including CRM and ERP.

Psycopg2 is an essential library for Python developers working with PostgreSQL databases. Its DB-API 2.0 compliance, efficiency, security and robustness make it a popular choice for database applications in production environments. By using psycopg2, developers can harness the power of PostgreSQL effectively and securely, facilitating the creation of web applications, data analysis and robust business systems.

2.2.5.7 Requests

Requests is a high-level Python library for making HTTP requests in a simple and human way. Developed by Kenneth Reitz and initially released in 2011, Requests is widely used to interact with APIs and websites, making it easy to send HTTP requests and receive responses without the complexity associated with standard HTTP handling libraries in Python, such as urllib.

It has an intuitive and readable user interface, which simplifies making HTTP requests (GET, POST, PUT, DELETE, etc.) and handling responses and also includes integrated support for HTTPS connections, SSL certificate verification and HTTP basic/digest authentication [Ronquillo, 2024].

Requests allows session parameters (such as cookies) to be persisted between multiple requests, facilitating continuous interactions with websites and APIs that require authentication and facilitates sending data in the URL, customised headers, form data and files in requests.

It provides error and exception handling to capture and deal with request failures efficiently and allows advanced configurations such as timeouts, automatic redirects, proxies and connection pooling.

Below, compiled a list of benefits from Requests was compiled:

- Ease of Use: The simple and intuitive interface allows complex HTTP operations to be carried out with just a few lines of code.
- Readability and Maintainability: Code written with Requests is more readable and easier to maintain compared to other standard HTTP libraries.

- Wide Adoption and Support: Requests is widely used in the Python community, with extensive documentation, tutorials and community support.
- Interaction with APIs: Used to send and receive data from RESTful APIs, facilitating integration between systems and services.
- Web Scraping: In combination with libraries such as Beautiful Soup, Requests is used to collect data from websites.
- Web Task Automation: Automating processes that involve repetitive interactions with websites, such as logging in and submitting forms.

Requests is an essential library for Python developers who need to interact with the web and perform HTTP operations. Its simplicity, security and robustness make it an indispensable tool for tasks involving communication with APIs, web scraping and automating web interactions. By using Requests, developers can focus on building advanced functionality without worrying about the underlying complexity of HTTP operations.

2.2.5.8 NumPy

NumPy (Numerical Python) is a fundamental library for scientific computing in Python. It provides support for multidimensional arrays and matrices, along with a comprehensive collection of high-level mathematical functions for operations on these arrays. Originally developed by Travis Oliphant in 2006, NumPy is the basis for many other data science and machine learning libraries in Python, such as SciPy, Pandas, Matplotlib and scikit-learn.

NumPy introduces the ndarray object, which is an efficient data structure for storing and manipulating data in multiple dimensions. These arrays are more memory and performance efficient compared to native Python lists. It also allows vectorised arithmetic and logical operations on integer arrays without the need for explicit loops, resulting in more concise and faster code [M, 2024].

NumPy includes a wide range of mathematical functions for complex operations, such as linear algebra, Fourier transforms, random number generation, among othersand facilitates the integration of Python code with libraries written in C, C++ and Fortran, allowing the use of high-performance libraries in these languages.

It offers powerful mechanisms for indexing and slicing arrays, making it easier to access and manipulate subsets of data and allows arithmetic operations to be performed efficiently between arrays of different shapes, without the need to duplicate data.

Below, compiled a list of benefits from NumPy was compiled:

- Performance: NumPy is highly optimised for array operations, offering significantly better performance compared to native Python lists.
- Simplicity and Productivity: Simplifies the writing of scientific and engineering code, allowing developers to focus on high-level problems rather than low-level implementation details.

- Community and Support: It has a large and active community, with extensive documentation, tutorials and resources available for learning and support.
- Base for Other Libraries: Serves as a base for many other data science and machine learning libraries in Python.
- Data Analysis: Used widely for manipulating and analysing large data sets.
- Machine Learning: The basis for many data processing operations and numerical calculations in machine learning libraries.
- Image and Signal Processing: Applied in image and signal processing and analysis operations.
- Scientific Simulations: Used in scientific simulations and modelling, where efficient numerical calculations are essential.

NumPy is an essential library for scientific computing in Python. Its ability to handle multidimensional arrays and perform complex mathematical operations efficiently makes it indispensable for data scientists, analysts and engineers. By using NumPy, developers can take advantage of the library's simplicity and performance to manipulate, analyse and process data in a wide range of scientific and engineering applications.

2.2.5.9 urllib3

urllib3 is a powerful and easy-to-use Python library for making HTTP requests. It's a standalone library that offers many useful features, such as managing persistent connections, automatic resending in the event of failures and HTTPS support. Developed as a more user-friendly and robust alternative to Python's standard urllib module, urllib3 is widely used in web applications and scripts that need to interact with the web reliably and efficiently.

urllib3 supports connection pooling and connection reuse with HTTP and HTTPS, reducing connection overhead and improving performance and includes SSL/TLS support for secure HTTPS connections, with certificate verification to guarantee communication security.

It also facilitates sending HTTP/HTTPS requests and handling responses, enabling complex operations such as file uploads and form data submissions and implements automatic resubmission in the event of temporary network failures, such as connection drops or timeouts, increasing the robustness of applications [Small, 2024].

It also includes functionalities for encoding and decoding URLs, making it easier to construct requests with query parameters and allows the configuration and use of proxies to redirect HTTP requests.

urllib3 is fully compatible with both versions, guaranteeing its use in a wide range of projects and legacy systems.

Below, a list of benefits from urllib3 was compiled:

- Performance: Optimised for efficient use of connections, improving the performance of applications that make multiple HTTP requests.
- Ease of Use: Simple and intuitive interface that makes it easy to send HTTP requests and handle responses.
- Robustness: Features such as automatic resending and support for persistent connections increase the robustness of applications.
- Security: SSL/TLS support for secure connections, guaranteeing data integrity and confidentiality.
- Interaction with APIs: Used to send and receive data from RESTful APIs, facilitating integration between systems and services.
- Web Scraping: In combination with other libraries, urllib3 is used to collect data from websites.
- Web Task Automation: Automating processes that involve repetitive interactions with websites, such as logging in and submitting forms.

urllib3 is an essential library for Python developers who need to perform HTTP operations efficiently and securely. Its simplicity, robustness and support for advanced features such as automatic resending and persistent connections make it an indispensable tool for any application that interacts with the web. By using urllib3, developers can focus on building advanced functionality without worrying about the underlying complexity of HTTP operations.

2.2.5.10 TinyMCE

TinyMCE is a JavaScript-based rich text editor (WYSIWYG - What You See Is What You Get), widely used to integrate advanced text editing functionalities into web applications. It allows users to edit and format content directly in the browser, offering an interface similar to a traditional word processor. TinyMCE is highly customisable and extensible, supporting a wide range of plugins and configurations that allow the editor to be adapted to the specific needs of each application.

TinyMCE Offers a friendly and easy-to-use interface, similar to common text editors such as Microsoft Word or Google Docs. It supports a variety of plugins and configurations that allow you to add or remove functionality as required. Also works consistently in all major browsers, including Chrome, Firefox, Safari, Edge and Internet Explorer. Allows advanced text formatting, insertion of images, links, tables, lists and much more. TinyMCE can be easily integrated into any web application, regardless of the framework or back-end language used.And tt has an active community of developers and detailed documentation that makes it easy to solve problems and implement new features [Long, 2024]. Below, a list of benefits from TinyMCE was compiled:

- Advanced Content Editing: Facilitates the creation and editing of rich content directly in the browser, improving the end-user experience.

- Productivity: The intuitive interface and advanced functionalities help users edit content efficiently and quickly.
- Flexibility: Can be configured and customised to meet the specific needs of different applications and use cases.
- Consistency: Guarantees a consistent editing experience across different browsers and devices.
- Content Management Systems (CMS): Used in platforms such as WordPress, Joomla and Drupal to allow users to edit content intuitively.
- E-Learning Platforms: Facilitates the creation and editing of educational materials, quizzes and learning resources.
- Blogging and Publishing Applications: Allows bloggers and authors to easily create and format blog posts.
- Feedback and Comment Forms: Used to allow users to enter and format rich feedback or comments.

TinyMCE is a powerful and versatile tool for editing rich text in web applications. Its intuitive interface, extensive customisation and compatibility with various browsers make it a popular choice among developers and users looking to improve the online content editing experience. By integrating TinyMCE, web applications can provide a richer and more efficient editing experience, allowing users to create and manage content with ease.

2.2.6 HTML

HTML (HyperText Markup Language) is the standard language for creating web pages. It defines the structure and content of a page, using a series of elements and tags to outline different parts of the document, such as paragraphs, headings, links, images, and other types of media. HTML is the backbone of any website and is used in conjunction with CSS (Cascading Style Sheets) and JavaScript to create complete and interactive web pages.

HTML is a markup language, which means it uses tags to mark up different elements in the document [Hayes, 2024]. Each tag has a specific meaning and determines how the content within it should be displayed by the browser. It allows you to structure the content of the page in a hierarchical way, with elements nested within other elements, providing a logical and semantic organization. HTML tags provide meaning to content. For example, <h1> it represents a level 1 header, <p> represents a paragraph, and <a> represents a link. Proper use of semantic tags improves accessibility and search engine optimization (SEO). It can be combined with CSS for styling and with JavaScript to add interactivity, allowing for rich and dynamic user experiences. The ability to create links (hyperlinks) to other documents and resources is a key feature of HTML, allowing navigation between different pages and websites and HTML supports the inclusion of

various types of media, such as images, videos, and audios, using specific tags such as <img>, <video> and <audio>.

Below, I compiled a list of benefits from HTML:

- Simplicity: HTML is easy to learn and use, making it accessible to beginners in web development.
- Compatibility: Supported by all web browsers, ensuring that HTML pages are displayed correctly on different platforms and devices.
- Flexibility: Can be used to create anything from simple web pages to complex web applications, in conjunction with CSS and JavaScript.
- Accessibility: The proper use of semantic tags improves accessibility for people with disabilities, making web content more inclusive.
- Web pages: Creating static and dynamic pages for the web.
- Web applications: Structuring web applications in conjunction with JavaScript frameworks such as React, Angular, and Vue.js.
- Documentation: Use to create online documentation, tutorials and user guides.
- Email Marketing: Creating layouts for emails that are sent as part of marketing campaigns.

HTML is the fundamental markup language for creating content on the web. Its simplicity, flexibility and universal compatibility make it an indispensable tool for web developers of all levels. With the ability to structure content in a semantic way and interact with other technologies such as CSS and JavaScript, HTML continues to be the foundation on which the modern web is built, facilitating the creation of rich and accessible user experiences.

2.2.7 CSS

CSS (Cascading Style Sheets) is a style language used to describe the presentation of HTML documents. While HTML provides the structure of the content, CSS defines how that content will be displayed, allowing structure and style to be separated. This facilitates maintenance, consistent design and an improved user experience. Developed by the W3C (World Wide Web Consortium), CSS is essential for designing attractive and responsive web pages.

CSS allows the separation of HTML content and visual presentation, improving the organisation and maintenance of code. It also uses selectors to apply styles to specific elements of the document, based on tags, classes, IDs, attributes and pseudo-elements.

CSS Cascading refers to the way styles are applied, allowing multiple rules to be combined and prioritised based on specificity, origin and importance.

[Thorndyke, 2024] says that CSS includes functionalities such as media queries that allow the creation of responsive layouts, adjusting the presentation based on screen size and device and

also modern tools such as Flexbox and Grid Layout make it easy to create complex and flexible layouts.

IT also allows you to create smooth animations and transitions to improve interactivity and the user experience.

Below, I compiled a list of benefits from CSS:

- Maintenance and Reusability: Allows the creation of reusable style sheets, facilitating maintenance and design consistency across multiple pages.
- Consistent Appearance: Guarantees a consistent appearance throughout the site, regardless of the number of pages.
- Improves User Experience: With features such as animations and responsive layouts, CSS contributes to a more engaging and intuitive user experience.
- Performance: Improves the performance of web pages by separating the structure of content from style, allowing asynchronous loading of stylesheets. Common Applications
- Web page design: Used to style HTML elements and create attractive and functional layouts.
- Web Applications: In conjunction with JavaScript frameworks and libraries, CSS is used to style complex components and user interfaces.
- Responsive Development: Creating designs that adapt to different screen sizes and devices.
- Prototyping and Interface Design: An essential tool for designers and developers when creating and testing user interfaces.

CSS is an essential language for modern web development, enabling the creation of sophisticated and responsive designs. Its ability to separate style from the structure of HTML content facilitates maintenance and promotes a consistent design across all pages of a website. With advanced tools such as Flexbox, Grid Layout, and media queries, CSS empowers developers to create rich and adaptable user experiences, remaining a cornerstone in the development of web interfaces.

2.2.8 JavaScript

JavaScript is a high-level programming language often used to create interactive web pages. It is one of the three main web technologies, along with HTML and CSS, and is essential for front-end development. JavaScript is an interpreted, dynamic, object-orientated and prototype-based language that allows you to add interactive and dynamic functionality to web pages. Originally developed by Brendan Eich at Netscape in 1995, JavaScript has evolved significantly over the years and is now supported by all major web browsers.

JavaScript allows the creation of interactive web pages, responding to events such as mouse clicks, keyboard inputs and cursor movements and also supports dynamic typing and allows the content and structure of web pages to be modified in real time.

It uses an object-based model, but unlike traditional languages, uses prototypes instead of classes for inheritance and code reuse and can be easily integrated with HTML and CSS to create complex, responsive web applications.

JavaScript was initially designed to run on the client side (browser), but with the advent of environments such as Node.js, it is also widely used on the server side and it has a rich ecosystem of libraries and frameworks, such as React, Angular, Vue.js, and jQuery, which facilitate the development of complex web applications [O'Grady, 2024].

Below, I compiled a list of benefits from JavaScript:

- Interactivity: Allows the creation of rich, interactive user experiences, essential for modern web applications.
- Versatility: It can be used on both the client and server side, making it possible to create full-stack applications with the same language.
- Robust Ecosystem: It has a vast number of libraries, frameworks and tools that speed up development and increase productivity.
- Active Community: Large community of developers, with numerous learning resources, tutorials and support available.
- Web Development: Creation of dynamic and interactive web pages, with DOM manipulation, form validation and communication with servers.
- Web Application Development: Used in frameworks and libraries such as React, Angular, and Vue.js to create complex user interfaces and SPA (Single Page Applications).
- Server Development: Used with Node.js to create efficient and scalable web servers.
- Mobile and Desktop Application Development: Used with frameworks such as React Native and Electron to develop native mobile and desktop applications.

JavaScript is an essential programming language for modern web development. Its ability to add interactivity and dynamism to web pages, combined with its versatility and vast ecosystem of tools and libraries, make it an indispensable choice for front-end and full-stack developers. With an active and constantly growing community, JavaScript continues to evolve and expand its applications, remaining relevant and essential in the creation of innovative and interactive web experiences.

2.2.9 jQuery

jQuery is a fast, small and feature-rich JavaScript library designed to simplify the manipulation of HTML documents, event handling, animation and AJAX (Asynchronous JavaScript and XML) interactions on web pages. Launched in 2006 by John Resig, jQuery has become extremely popular with web developers due to its concise and easy-to-use syntax, which allows you to write

less code to achieve complex results. jQuery Allows you to easily select HTML elements using CSS-like selectors, which simplifies DOM (Document Object Model) manipulation. It facilitates the efficient addition, removal and modification of HTML elements and their attributes and provides methods for attaching and manipulating user events such as clicks, key presses and page scrolling, making user interaction with the page more dynamic. jQuery supports animations of HTML elements, such as fade, slide, and customised animation effects, which can be easily applied to page elements and simplifies the sending of asynchronous HTTP requests, allowing you to load and send data from the server without reloading the entire page, improving the user experience. [Echout, 2024] says that it has a vast library of plugins that extend its basic functionality, allowing you to add advanced features such as carousels, image galleries and calendars with just a few lines of code. jQuery supports all major browsers, ensuring that the functionalities developed with jQuery are consistent and work correctly in different browser environments.

Below, I compiled a list of benefits from jQuery:

- Simple Syntax: Simplifies the writing of JavaScript code, reducing the amount of code needed to perform common tasks.
- Compatibility and Support: Supports all major browsers, ensuring a consistent user experience.
- Efficiency: Allows you to develop interactive and responsive web applications quickly and efficiently.
- Large Community: It has an active community of developers who contribute plugins and offer support through forums and documentation.
- Web development: Widely used in the development of dynamic and interactive user interfaces.
- E-commerce: Implementation of functionalities such as dynamic shopping carts, product filters and checkout pages.
- Single-Page Applications: Development of single-page applications (SPA) that update content asynchronously without reloading the page.
- Image Galleries and Carousels: Creation of animated image galleries and dynamic content carousels.

jQuery is an essential JavaScript library for web development, offering a simplified way to manipulate the DOM, respond to user events, create animations and carry out asynchronous communications with the server. Its widespread adoption and ease of use continue to make it a popular choice among developers looking to create interactive and responsive user interfaces.

2.2.10 Bootstrap

Bootstrap is an open source front-end framework developed by Twitter to facilitate and speed up the development of responsive and modern websites and web applications. It provides a collection of CSS, HTML and JavaScript tools to create consistent and attractive user interfaces. Bootstrap is widely used by web developers due to its ease of use, extensibility and compatibility with various browsers and devices. Bootstrap includes a flexible and responsive grid system that automatically adapts to different screen sizes and devices, ensuring that layouts are displayed appropriately on desktops, tablets and smartphones. It offers a vast library of predefined components, such as buttons, forms, navbars, modals, carousels and much more, which can be easily customised and integrated into web projects [Ouellette, 2024]. Also guarantees compatibility with all the main modern browsers, including Chrome, Firefox, Safari, Edge and Internet Explorer and allows customisation through SASS variables, making it possible to adapt the framework to the specific needs of each project. Bootstrap includes a variety of JavaScript plugins to add interactive functionalities, such as sliders, popovers and alerts, without the need for complex coding and it has detailed documentation and practical examples to help developers understand and use the framework efficiently.

Below, a list of benefits from Bootstrap was compiled:

- Rapid Development: Facilitates the rapid creation of responsive and functional layouts, reducing development time.
- Visual Consistency: Ensures a consistent and professional appearance across all user interface elements.
- Community and Support: It has a large community of developers and an extensive base of resources and tutorials available online.
- Accessibility: Includes support for accessibility best practices, helping to create sites that are accessible to all users.
- Corporate Sites and Portfolios: Used to create professional sites and portfolios that adapt to different devices.
- Administration Panels: Widely used in administration panels due to their flexibility and variety of reusable components.
- Blogs and Content Sites: Facilitates the creation of blogs and content sites with attractive, responsive layouts.
- Web Applications: Integration into web applications to provide a modern, responsive user interface.

Bootstrap is a robust and versatile front-end framework that simplifies the development of responsive websites and web applications. Its wide range of components and functionalities, combined

with its ease of use and comprehensive documentation, make it a popular choice among developers looking to create efficient and attractive user interfaces. By using Bootstrap, developers can guarantee a consistent and pleasant experience for users on all types of devices.

2.2.11 Search Engine Optimization

SEO (Search Engine Optimisation) is the set of practices and techniques used to increase the visibility and positioning of a website in the organic results of search engines such as Google, Bing and Yahoo. The main objective of SEO is to attract more quality traffic to the site, improving the user experience and, consequently, increasing the chances of conversion and engagement [Google, 2024]. It can identify words and phrases that users type into search engines when looking for information, products or services related to the site's content. Tools such as Google Keyword Planner, SEMrush and Ahrefs are commonly used for this purpose. AlsoRefers to the practices carried out within the site to improve its ranking in search engines. This includes optimising content, meta tags, URLs, images and the internal structure of the site. Important components of on-page optimisation are:

- Title Tag: This should be relevant, contain keywords and be of an appropriate length.
- Meta Description: A brief summary of the content of the page that appears in search results. It should be attractive and contain keywords.
- Header Tags (H1, H2, H3, etc.): Used to structure the content, making it more readable for users and search engines.
- Friendly URLs: Short, descriptive URLs that contain relevant keywords.
- Quality Content: Original, relevant and useful text that meets users' needs. It should be well written and formatted for easy reading.

I also involves activities carried out outside the site to improve its visibility and authority. The main component is building backlinks, which are links from other sites that point to yours. Other aspects include content marketing, social media marketing and brand mentions.

Technical aspects of the site that affect SEO its indexing and crawling by search engines. It includes optimising loading speed, responsive design, site structure, use of HTTPS, creation of XML sitemaps and configuration of robots.txt files.

Search engines consider the user experience when ranking websites. Factors such as dwell time, bounce rate and ease of navigation influence SEO. A clean design, fast loading times and intuitive navigation are essential.

Below is a list of important points in SEO practices:

- Organic Traffic: Increases the number of visitors coming to the site via unpaid search engine results.

- Cost-Effective: Unlike paid advertising, organic traffic obtained through SEO requires no ongoing investment.
- Credibility and Authority: Sites that rank well in search engines are often perceived as more trustworthy and authoritative by users.
- Improved User Experience: SEO practices often result in a more organised and user-friendly website.
- Competitiveness: Effective SEO helps you compete with other companies in the same niche, achieving greater online visibility.

2.2.12 Large Language Model

Large language model refers to a type of Natural Language Processing (NLP) model that is trained on large amounts of data to understand and generate human language text. These models are designed to handle complex language tasks such as translation, question answering, text summarisation and even original text generation [Kasneci et al., 2023].

The term "large" indicates that these models are trained on massive data sets and are usually made up of millions or even billions of parameters. The more parameters a model has, theoretically, the greater its ability to understand and generate text in a sophisticated way.

2.2.13 Types of Large Language Models

There are various types of large language models, each with its own specific characteristics and applications. Here are some examples from [Zhao et al., 2023]:

- **GPT** (Generative Pre-trained Transformer) , developed by OpenAI, is a series of models that utilise the transformer architecture. GPT-3, for example, is a massively large language model that has demonstrated remarkable abilities in various natural language processing tasks.
- **BERT** (Bidirectional Encoder Representations from Transformers), created by Google, is designed to understand the context of words in a sentence, taking into account both preceding and subsequent words. It is often used for natural language understanding tasks.
- **XLNet** is another transformer-based model, also developed by Google, that incorporates elements of BERT and introduces an autoregressive approach, allowing the model to consider the entire sequence of words in order to understand context.
- **T5** (Text-to-Text Transfer Transformer), developed by Google, proposes a unified approach to natural language processing tasks, where all tasks are formulated as "text-to-text" tasks.
- **RoBERTa** (Robustly optimised BERT approach) is a variation of BERT, RoBERTa optimises various hyperparameters and training techniques to improve performance on specific natural language processing tasks.

These are just a few examples, and research into language models continues to evolve. Each of these models has its own advantages and may be better suited to specific tasks, depending on the requirements of the project.

2.2.14 Why ChatGTP is linked with Large Language Models

The GPT (Generative Pre-trained Transformer) is an example of a large language model, and it falls into this category due to its architecture and size. OpenAI trains GPT models on large datasets containing a wide variety of texts.

GPT is based on the Transformer architecture, which is a neural network architecture designed to process sequences of data, such as text, efficiently [Muehmel, 2024]. Before being used for specific tasks, GPT models are pre-trained on large data sets. During this phase, the model learns to understand the structure of the language and to capture patterns in different contexts. In the case of GPT-3, it is trained on a massive scale, with billions of parameters.

After pre-training, the model can be fine-tuned for specific natural language tasks. This involves training the model on a smaller dataset and labelling it for a specific task, thus adapting to the user's needs.

When you use ChatGPT or other GPT-based services, you are interacting with a fine-tuned version of the model. This adjustment may involve specific optimisations to make the model more suitable for cohesive and useful responses in a conversational context.

To summarise, ChatGPT is a specific implementation of GPT that has been tuned for interactive conversation. It is part of the family of large language models, which are trained to understand and generate text on a large scale.

Chapter 3

Review of Related Work

This chapter presents and discusses the related work, so that it can be realised that there is still no solution to this problem, or if there is, it may not be the right one. The aim is to compare various articles that focus problems similar to the one presented here, and to present and discuss the proposed solutions. This chapter will demonstrate the relationship that technology has with journalism and how technology may help journalists in doing their job.

3.1 Related Work

In order to fit the topic of this project into the state of the art, the research focused on the following points: support tools for journalists and artificial intelligence in journalism.

In the first article analysed [Franks et al., 2021], entitled "Using computational tools to support journalists' creativity", the authors analyse the lack of specific digital tools adapted to support journalists' creativity. While some general tools are adapted to help with story development, such as search engines and text analysis, few are designed specifically for journalists and do not explicitly focus on ideas' generation during news development.

Examples of digital tools for journalism include DocumentCloud [DocumentCloud, 2024], which analyses documents for references and timelines, and prototypes such as NewsReader [NewsBlur, 2024], that uses text analysis and AI to categorise news and financial data, although they do not focus on generating new angles for news stories.

Other tools are mentioned, such as the Story Discovery Engine [Broussard, 2014], which helps analyse text and discover stories, but does not yet support the creative discovery of new story angles. The text also mentions startups such as Loyal.ai, which offers interactive assistants for quick searches, but has no concrete evidence of supporting journalists' creativity [Franks et al., 2021].

Another article [Dhiman, 2023] discusses the role of artificial intelligence (AI) in journalism, emphasising that although AI can automate aspects such as data analysis, fact-checking and even news production, it cannot replace human journalists. The crucial skills for quality journalism - such as empathy, understanding the nuances of language and culture and recognising ethical considerations - are inherent to human beings.

AI is seen as a tool to complement human journalists, allowing them to concentrate on complex reporting while AI handles routine tasks. It highlights how AI can reduce variable costs in journalism by automating data analysis, fact-checking, news production and personalising content for readers. However, it notes that implementing AI can require significant technological investment and raises ethical concerns, such as potential bias in algorithms.

The article also looks at how AI, such as ChatGPT, can help with tasks related to journalism, such as fact-checking, writing news articles, creating headlines and analysing data. It emphasises the importance of using AI tools judiciously and verifying information from any source, including AI language models.

Overall, while AI presents opportunities to simplify journalistic processes and reduce costs, it emphasises the need for ethical considerations and human oversight when using these technologies.

Another article [Kotenidis and Veglis, 2021] notes that algorithmic technology has advanced considerably in recent years, but faces challenges, especially in the automated production of content. One crucial limitation is the reliance on structured data. In addition, although algorithms can mimic human writing, they still lack in areas such as analytical thinking, flexibility and creativity. This creates a disconnect between algorithms and humans, especially in automated newsrooms.

In addition to automated content production, there are challenges in other areas, such as data mining, where the results can be insignificant or even erroneous. The personalisation of content raises concerns about the fragmentation of public opinion and the creation of "echo chambers", reinforcing one-sided views.

Despite this, algorithmic technology is promising for solving contemporary problems in journalism, such as information overload and credibility. Although the introduction of more sophisticated algorithms may cause turbulence, it is hoped that they will help produce news faster and on a larger scale, expanding coverage to unprofitable events. However, this could lead to information overload, exacerbated by the spread of fake news.

A possible future solution would be a fully autonomous news system, capable of combining data mining, algorithmic content production and news distribution without human intervention.

This recent study [Lermann Henestrosa et al., 2023] investigated how readers perceive content produced automatically by algorithms, focussing specifically on science journalism articles. Although much content is already generated automatically, there is little knowledge about how artificial intelligence (AI) authoring affects audience perception, especially in more complex texts.

The researchers highlighted technological advances in automated text generation, citing large-scale language models such as OpenAI's GPT-3, Microsoft and NVIDIA's Megatron-Turing as examples of advanced natural language generation (NLG) capabilities. These models illustrate the ability to simulate human writing, but there is still a dearth of studies on the impact of AI authoring on complex texts.

The studies conducted analysed readers' perceptions of science journalism articles written by algorithms. Surprisingly, even when an AI author presented information in an evaluative way on a scientific topic, there was no decrease in the credibility or trust attributed to the text. The

presentation of the information was identified as the main factor influencing readers' perceptions, regardless of the declared authorship.

An interesting point was that although the participants considered the AI author to be less "human" in their writing, this perception did not impact the evaluation of the messages. This raised questions about the importance of the nature of the author for readers in terms of credibility and trust in the content.

The results indicated an acceptance of AI as an author of scientific texts, provided certain conditions were met. In addition, the studies revealed positive attitudes towards automation, suggesting a favourable attitude towards the use of algorithms in content production.

However, the studies had some limitations, including the influence of participants' prior beliefs and the perceived neutrality of the information presented. Future research should further explore how readers understand the workings of text production algorithms and the relationship between the perceived "humanisation" of AI and the credibility of the content generated.

In summary, the study has contributed to a deeper understanding of how AI authorship affects the public's perception of complex content, paving the way for reflection on the acceptance of algorithmically generated texts in more diverse contexts.

AI has had a significant impact on the field of journalism, particularly in addressing various challenges. The article discusses the integration of AI technologies into news organizations, specifically focusing on the use of natural language generation (NLG) to automate the creation of news stories. The implementation of AI in journalism aims to improve efficiency, enhance journalistic work, and strengthen audience relationships.

According to a survey conducted by the media think-tank at the London School of Economics (LSE), news organizations see potential for AI throughout the entire production process, including news gathering, production, and distribution. The main motivations for adopting AI in journalism are increased efficiency in the newsroom, improved business functionality, and enhanced relevance for audiences.

AI technologies, particularly NLG, have been utilized for various applications in journalism. These include content recommendation, improved tagging, automated stories, summaries, and text-to-audio conversion. AI is also valuable for data cleaning, extraction, linking records, and identifying news angles. It is particularly useful for handling resource-intensive or technically challenging stories.

The article highlights that AI is viewed as a partner in the journalistic process, augmenting and optimizing newswork. It can assist in sorting out meaning from vast amounts of information and contribute to the production of scoops, analysis, and brand building. The focus is on augmenting work processes that lack creativity, allowing journalists to concentrate on more important aspects of their work.

While AI has the potential to address certain challenges in journalism, there is also recognition of the impact on jobs. The article emphasizes the importance of defining journalistic value in terms of investigative stories, analysis, and brand building, rather than simply relying on AI for routine news stories.

In summary, AI, particularly NLG, has been integrated into news organizations to automate the creation of news stories. It offers potential benefits in terms of efficiency, relevance, and improved audience relationships. However, there is a need to strike a balance between AI automation and the preservation of journalistic creativity and value.

The study [Sirén-Heikel et al., 2023] examines the influence of AI technologies on journalism by studying the logics underpinning the construction of technical solutions. It uses institutional logics as a theoretical framework to understand the interrelationships between institutions, individuals, and organizations in social systems. The integration of AI technologies into news organizations impacts how work is organized and reshapes journalism. The study explores companies that develop and sell NLG (natural language generation) services for journalism, revealing how technologists view their interactions with news organizations.

The participants in the study represent different educational backgrounds, cultures, and languages, yet share a sensemaking of their relationship with journalism. The presupposition that technologists and journalists occupy separate fields of logic is validated through the interviews. The companies involved in the study differ in size, market reach, and age, allowing for analysis of both established players and new entrants to the field.

The study [Sirén-Heikel et al., 2023] focuses on the interplay that occurs when AI technologies are incorporated into organizations. It reveals that AI technologies in news organizations require interaction between the logics of journalism and technologists, leading to competition and assimilation of logics. The study is limited by only interviewing one representative from each company, but sheds light on the influence exerted by AI technologies on journalism.

The study identifies [Sirén-Heikel et al., 2023] a shared theory of rationalization, a frame of optimization, and a narrative of misinterpretation among the companies involved in developing AI solutions for journalism. The theory of rationalization is distilled as solutionist, rationalizing news organizations by solving the problem of reaching audiences at scale without adding human resources. This allows newswork to refocus on creating value for audiences. The frame of optimization centers around optimizing newswork, freeing journalists from having to do "boring news that doesn't add any value." It derives from the theory of rationalization and verbalizes a normative conceptualization of what journalism ought to be. The narrative of misinterpretation is expressed through the limitations of explaining AI systems, to whom it is explained, and when it is explained.

This also delves into the perspectives of technologists involved in developing AI solutions applied in journalism. It reveals that different companies provide different solutions for automated content, with varying generation and distribution models. The study sheds light on the influence exerted by AI technologies on journalism by studying the logics underpinning the construction of the technical solutions.

According to a survey conducted by the media think-tank at the London School of Economics (LSE), news organizations see potential for AI throughout the entire production process, including news gathering, production, and distribution. The main motivations for adopting AI in journalism are increased efficiency in the newsroom, improved business functionality, and enhanced relevance for audiences.

AI technologies, particularly NLG, have been utilized for various applications in journalism. These include content recommendation, improved tagging, automated stories, summaries, and text-to-audio conversion. AI is also valuable for data cleaning, extraction, linking records, and identifying news angles. It is particularly useful for handling resource-intensive or technically challenging stories.

The article highlights that AI is viewed as a partner in the journalistic process, augmenting and optimizing newswork. It can assist in sorting out meaning from vast amounts of information and contribute to the production of scoops, analysis, and brand building. The focus is on augmenting work processes that lack creativity, allowing journalists to concentrate on more important aspects of their work.

While AI has the potential to address certain challenges in journalism, there is also recognition of the impact on jobs. The article emphasizes the importance of defining journalistic value in terms of investigative stories, analysis, and brand building, rather than simply relying on AI for routine news stories.

In conclusion, the study provides valuable insights into the interplay between AI technologies and journalism, shedding light on the influence exerted by AI technologies on journalism. It highlights the need for interaction between the logics of journalism and technologists, and the competition and assimilation of logics that occur when AI technologies are incorporated into organizations. The study contributes to a better understanding of the impact of AI technologies on journalism and the dynamics of the evolving relationship between technology and news organizations.

3.2 Discussion

Based on all the articles analyzed, it is clear that there is still a long way to go on both sides. It is sometimes difficult to introduce technological evolution in certain areas, but journalism has increasingly benefited from artificial intelligence. So let us analyze 3.1:

Support Tools and AI in Journalism: An article highlights the lack of specific digital tools designed to foster journalists' creativity. Although there are general tools to help develop stories, few are dedicated to generating ideas during news production. Examples mentioned include DocumentCloud and NewsReader, but there is a lack of tools focussed on generating new angles for news stories.

Role of AI in Journalism: AI is seen as a complementary tool to human journalists, capable of automating tasks such as data analysis, fact-checking, and even news production. However, it is emphasized that AI cannot replace essential human skills for journalism, such as empathy, understanding the nuances of language and ethical considerations.

Challenges of AI in Journalism: Automation in content production faces challenges, including limitations in interpreting unstructured data and the difficulty in replicating human analysis, flexibility, and creativity. There are concerns about excessive personalisation of content, which can create echo chambers and reinforce one-sided views.

Audience Perception of AI-Generated Texts: Studies on the perception of AI-generated texts indicate that the presentation of information is crucial for credibility, regardless of whether it is produced by humans or algorithms. Although participants recognize AI-generated writing as less "human", this does not affect their evaluation of the message.

Impact of AI on News Organizations: The incorporation of AI technologies into news organizations influences the way work is organized and transforms journalism. The interaction between the logics of journalism and technology generates competition and assimilation of logics.

Potential Use of AI in Journalism: AI is seen as a partner in the journalistic process, capable of optimizing tasks and freeing journalists to focus on more complex aspects of the job. Applications of AI include content recommendation, automatic summaries, data analysis, and assistance with technically challenging stories.

Overall, AI is perceived as a promising tool for simplifying journalistic processes, but there is an emphasis on the need for ethical considerations and human supervision when using these technologies.

Table 3.1: Summary of conclusions from the state of art.

Title and Reference	**Main conclusions**
Using computational tools to support journalists' creativity [Franks et al., 2021]	- Lack of digital tools to support journalists' creativity.
How Artificial Intelligence Helped Me Investigate Textbook Shortages [Broussard, 2014]	- AI offers interactive assistants, but has no concrete evidence of supporting journalists' creativity
Does Artificial Intelligence Help Journalists: A Boon or Bane? [Dhiman, 2023]	- AI can automate aspects such as text analysis, fact-checking, and even news production. - Algorithms or models may be biased. - Ethical considerations and human oversight when using these technologies.
Algorithmic Journalism —Current Applications and Future Perspectives [Kotenidis and Veglis, 2021]	- Crucial limitation of AI is reliance on structured data. - AI lacks analytical thinking, flexibility, and creativity. - Some results can be insignificant or erroneous.
Automated journalism: The effects of AI authorship and evaluative information on the perception of a science journalism article [Lermann Henestrosa et al., 2023]	- Readers tend to attribute similar credibility or trust to AI-generated news contents as to human-written news.
At the crossroads of logics: Automating newswork with artificial intelligence—(Re)defining journalistic logics from the perspective of technologists [Sirén-Heikel et al., 2023]	- Integration of AI technologies into news organizations impacts how work is organized and reshapes journalism; - Participants in a study, representing different backgrounds, share a presupposition that technologists and journalists occupy separate fields of logic; - AI technologies in news organizations require interaction between the logics of journalism and technologists, leading to competition and assimilation of logics; - Identifies a shared theory of rationalization, a frame of optimization, and a narrative of misinterpretation among the companies involved in developing AI solutions for journalism.

Chapter 4

Project Modeling and Development

In the sub-chapters below, we will talk about the project both in terms of diagrams and more technical details. In this first sub-chapter, it´s showned all the diagrams that were made, and explain a little about them. In the development part of the project, it´s explaines the platform created, all the pages and also its algorithm.

4.1 Project Modeling and Contextualization

The use case diagram in Figure 4.1 illustrates the process of generating breaking news content with the help of artificial intelligence and search engine optimisation. The main use cases identified in the diagram are:

- **Create Breaking News Generated by Trend:**

  This use case involves the creation of breaking news based on current trends. It includes the following sub-cases: Search on the News Sources: Search the available news sources for relevant information. Get Texts from Sources: Get the necessary texts from the searched news sources. Build News Text with LLM (AI): Build the news text using Large Scale Language Models (LLM). Edit, Save and Share News: Edit, save and share the generated news.

- **Create News Generated by Keyword:**

  This use case involves creating news based on specific keywords. As with the previous use case, it includes the sub-cases: Search on the News Sources Get Texts from Sources Build News Text with LLM (AI) Edit, Save and Share News

- **List and Select Trending Topics:**

  This use case allows to list and select the topics that are trending, serving as the basis for generating news. It includes the sub-case: Get Trending Topics: Get the trending topics to be used in news creation.

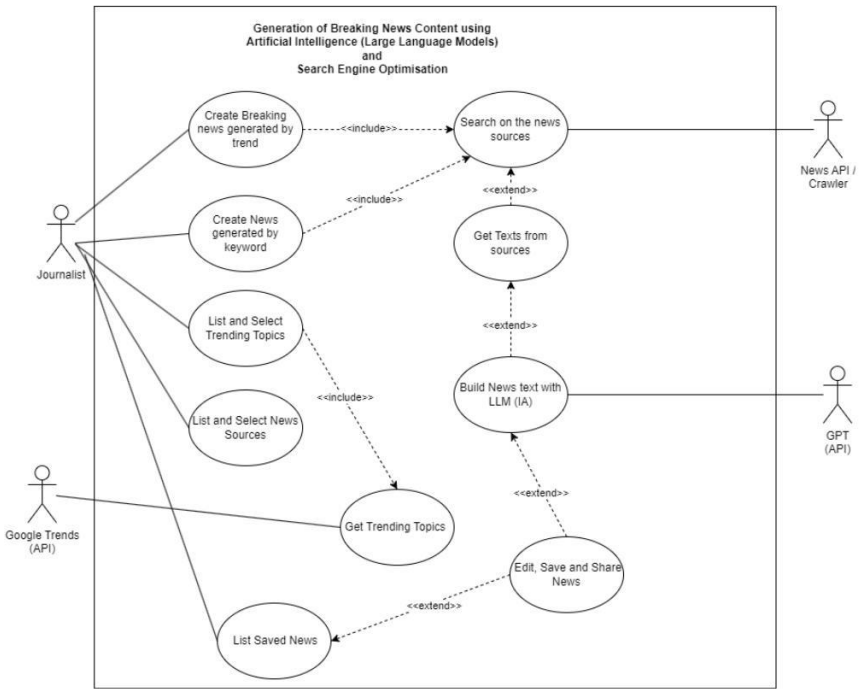

Figure 4.1: Use case Diagram of the project.

- **List and Select News Sources:**

  This use case involves listing and selecting the news sources that will be used to search for information and generate content.

- **List Saved News:**

  This use case allows to list the news that has been saved after being created and edited.

- **Relationships between Use Cases**

  - **Inclusion («include»):** Creating news, by both trends and keywords, includes searching news sources and selecting trending topics, for generating the news contents. Listing and selecting trending topics includes retrieving trending topics.
  - **Extension («extend»):** Searching the news sources can be extended by obtaining texts from the sources. Building the news text with AI can be extended by editing, saving and/or sharing the news.

This use case diagram serves to illustrate how artificial intelligence can be used to automate and optimise the creation of breaking news content. Tt demonstrates the different steps and interactions required to generate relevant, up-to-date news using Large-Scale Language Models (LSMs). The use case diagram also highlights the importance of selecting and analysing news sources and trends, as well as integrating search engine optimisation techniques to ensure that the content created is not only accurate and current, but also easily accessible and findable by users.

Through this diagram, it is possible to understand the complexity and interconnections of the different processes involved in generating news content, providing a solid basis for discussing the benefits and challenges of applying AI in this context.

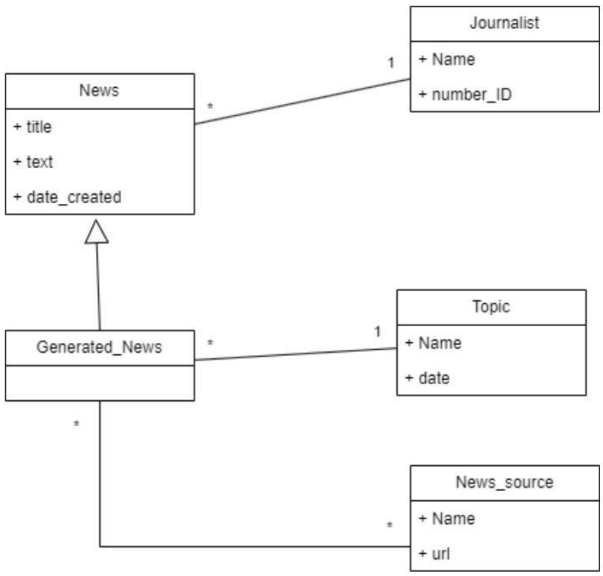

Figure 4.2: Domain Entity Diagram of the project.

The Domain Entity Diagram in Figure 4.2 represents the main entities involved in the news generation, editting and management process and their relationships. The entities and their characteristics are described below:

- **News**
  - **Attributes:**
    - **title:** The title of the news item.
    - **text:** The textual content of the news item.
    - **date_created:** The date on which the news item was created.
  - **Relationships:**

A news item can be associated with one or more journalists.

A news item can be part of a generated news item.

- **Generated_News**
  - **Attributes:** Has no attributes of its own other than those inherited from the News entity.
  - **Relationships:** A generated news item can be associated with several topics. A generated news item can be associated with several news sources.
- **Journalist**
  - **Attributes:**
    - **name:** The name of the journalist.
    - **number_ID:** The journalist's unique identifier.
  - **Relationships:** A journalist can create several news items.
- **Topic**
  - **Attributes:**
    - **name:** The name of the topic.
    - **date:** The date relevant to the topic.
  - **Relationships:** A topic can be associated with several generated news items.
- **News_source**
  - **Attributes:**
    - **name:** The name of the news source.
    - **url:** The URL of the news source.
  - **Relationships:** A news source can be associated with several generated news items.

This domain entity diagram illustrates how the different components of the news generation system interact with each other. The focus of the dissertation is the use of artificial intelligence to automate the creation of breaking news. This diagram is essential for understanding the relationships between data and how they are organised in the system.

- **Entities and Their Functions:**
  - **News:** Represents the news created, which is the main product of the system.
  - **Generated_News:** Represents the news specifically generated by the AI system, highlighting the use of advanced technology to create content.
  - **Journalist:** Represents the human creators of news, whose contribution is crucial to validating and editing the automatically generated content.

- **Topic:** Represents the topics that drive news creation, essential for ensuring that the content generated is in line with current trends and interests.
- **News_source:** Represents the sources from which information is collected, crucial to guaranteeing the accuracy and reliability of the news generated.

Through this diagram, the data entry points and how they flow through the system can be clearly identified, from the collection of information from sources to the creation and final editing of the news story. It facilitates discussion about the technical implementation of the system, providing a basis for analysing requirements and developing software.

4.2 Project Development

This project has been developed using Django technology. Django is a framework of the Python language focused on web development. The Pycharm Professional IDE, from the company Intellij, has been used. Firstly, Python had to be installed, in this case version 3.12.1.

After installing python, a new folder has been created and the python environment has been set up for the project.

After this creation, the Django Python framework was installed with version 5.0.4. After that, project was created and then an app. After the creation of the app, it was decided to do a proof of concept to substantiate the initial idea. When the platform starts up the user needs to login (Figure 4.3).

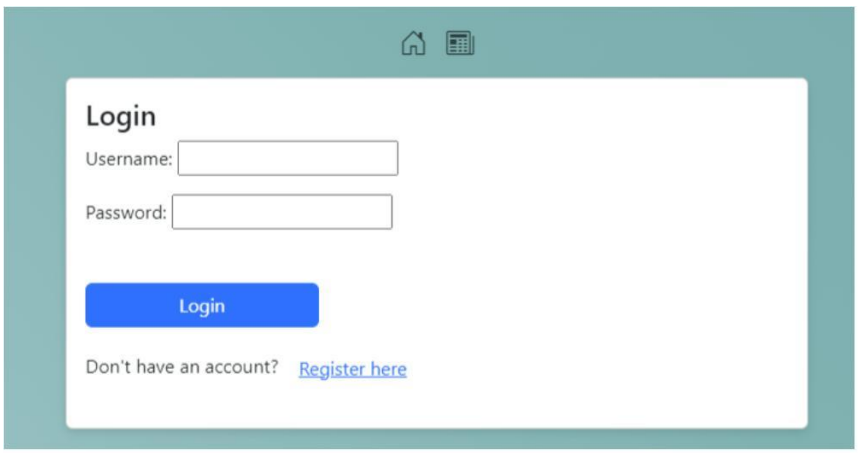

Figure 4.3: Login Page.

If the user does not have an account on the platform, they can then register on the platform (Figure 4.4).

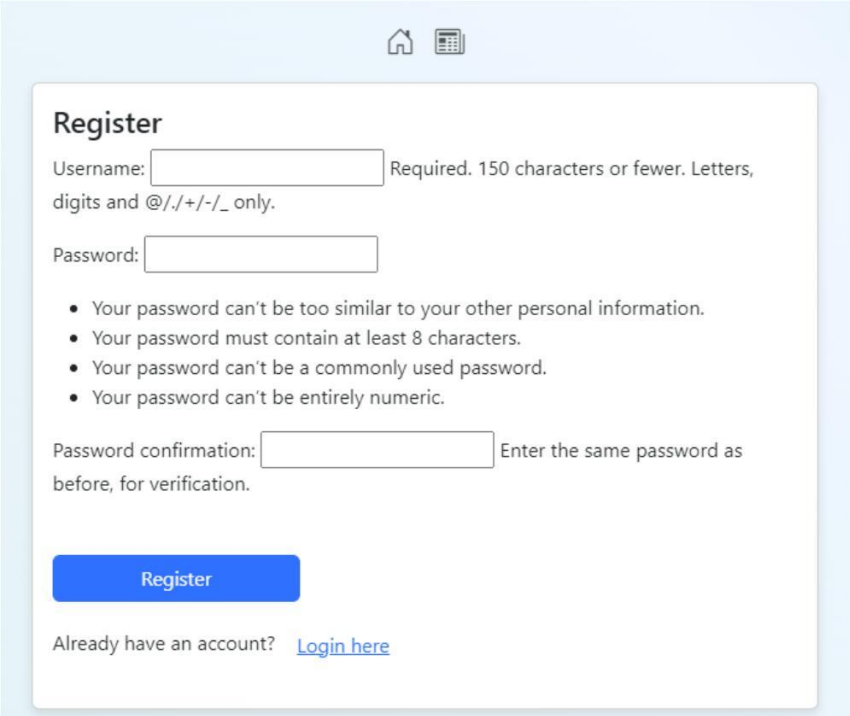

Figure 4.4: Register Page.

Then, the user is presented with a home screen (Figure 4.5) where they can perform two actions: create news using daily trends, or create news using keywords.

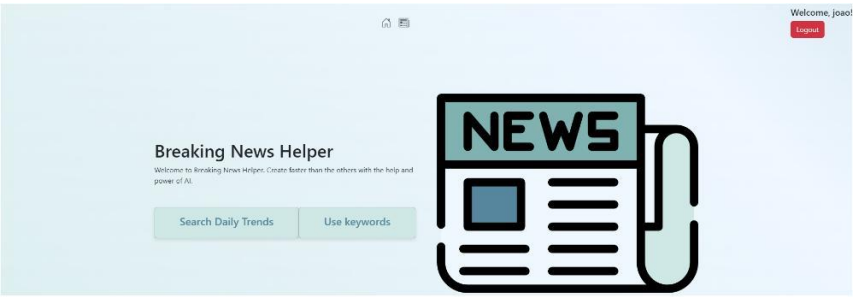

Figure 4.5: Home Page.

If the user clicks on the first option "create news through daily trends", they are taken to a screen (Figure 4.6) where they can choose the trend to be used to create the news. This screen

shows a list of all Portuguese daily news' trends.

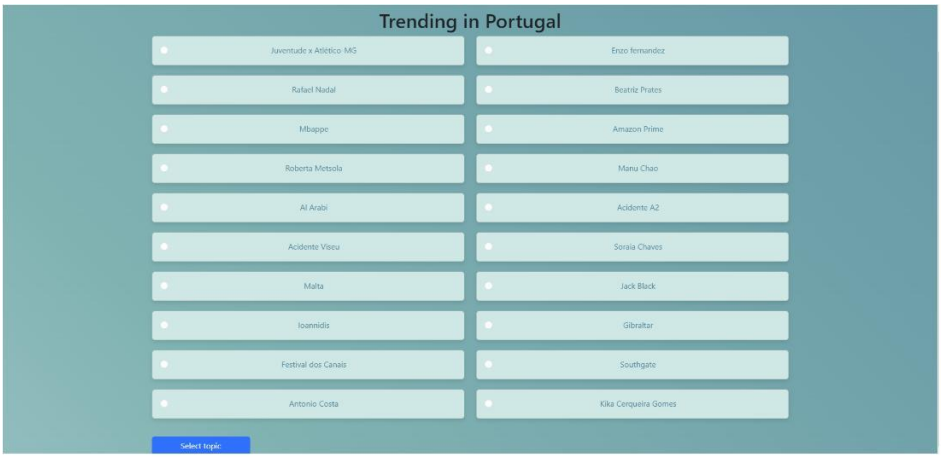

Figure 4.6: Create breaking news using trends.

After choosing the trend, the user is taken to the sources screen. From the sources, the user can choose the following: Jornal de Notícias, or Público, and whether they want the full story or a list of topics (see Figure 4.7).

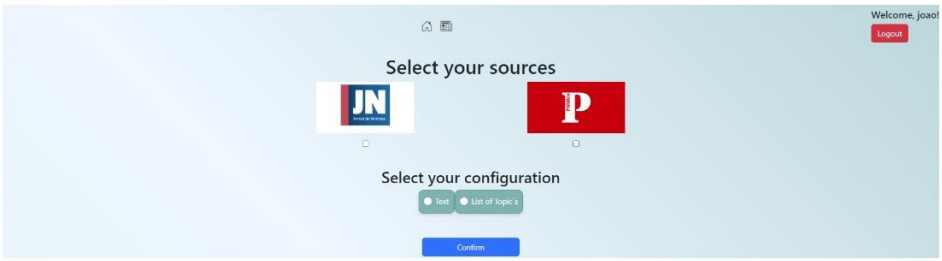

Figure 4.7: News configuration.

After selecting the sources, the user will have to wait a few seconds for their news item to be created, based on the chosen trend and sources.

After a few seconds, the user is taken to a page where they will find their new story in a text editor, ready to be edited to their liking, and with a field to fill in the title of the story (see Figure 4.8). If the user is satisfied with the news item, they can click on the save news button and they will be taken to a back office area.

In this back office area, users can manage all the news they have created (see Figure 4.9).

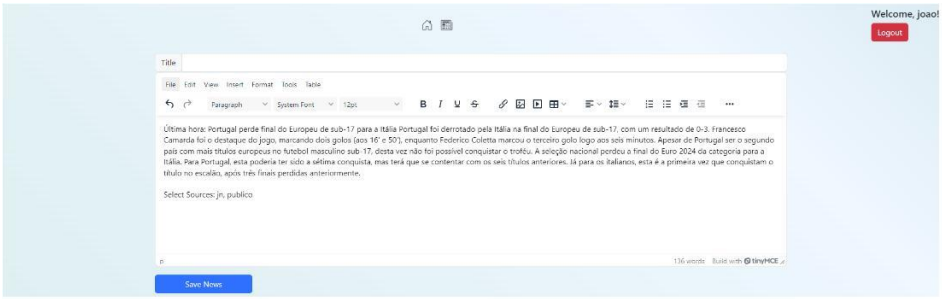

Figure 4.8: Breaking News generation and edition.

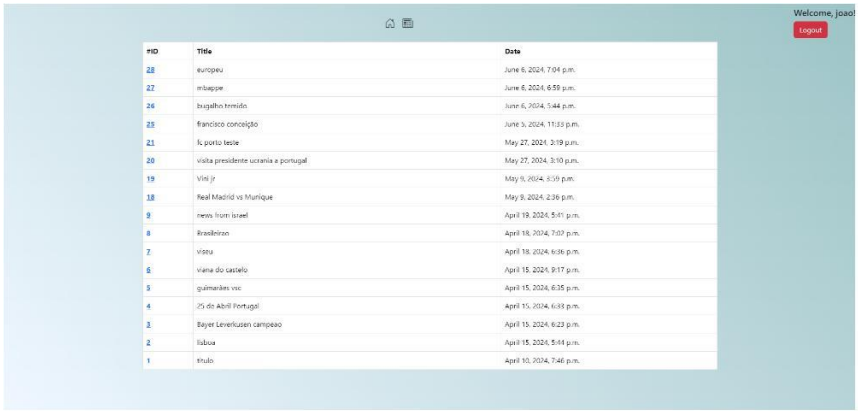

Figure 4.9: Back Office for Users.

By selecting an item on the list, they navigate to a new page and can edit the news they want, or even delete it (see Figure 4.10).

In the second home screen option, the user is taken to a screen where they are asked to type in the topic on which they want the news story to be created. After entering the topic, the user is taken to the sources page and the process is the same as described above (see Figure 4.11).

The system is being supported by a PostgresSQL database, which is supported in the Cloud by an Amazon Web Service. It was first necessary to configure this database in AWS, from information about the IPs in the inbound rules to the integration of this database service into the Django project. Django, by default, comes with a SQL lite installation, but in discussions it was felt that this database would not serve the purpose of the project as it would only be used for testing some developments, so it is demanding a more stable and secure database. This is why PostGresSQL, supported by AWS, was chosen. Once the database had been configured and integrated with the

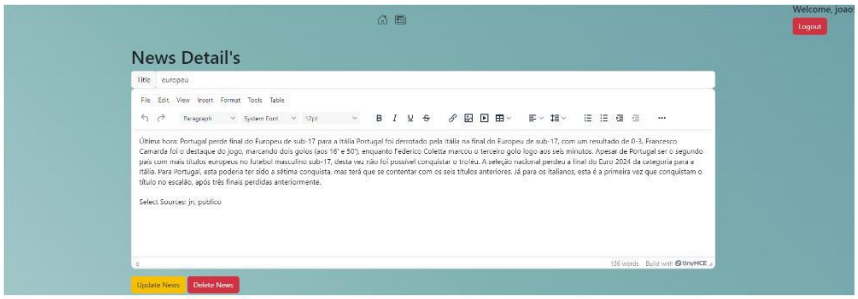

Figure 4.10: Edit or Delete Mode.

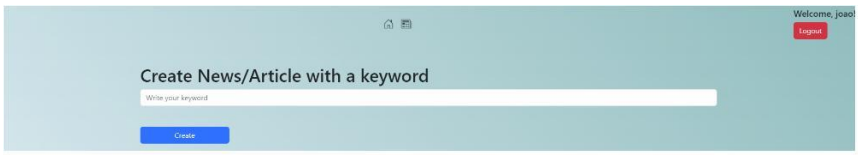

Figure 4.11: Second option for creating Breaking News.

project, the project models created before needed to be implemented. These models represent our tables from a traditional database, but with the Django framework they are created using "models". Once the News, Generated_News, Topic, News_source and Journalist models had been created, the project could begin.

Firstly, it is learnt how to display data on an HTML page that comes from a get from the Jornal Público API. This first part was a little tricky because there are very few tools that the Portuguese newspaper Público makes available for public use, so we had to create tools for the purpose of the project. Here is where Selenium and Beautiful Soup are introduced, because with these two tools we could be able to extract text from the sources for the construction of the news story. The Google news portal has been used. Here, it can search directly for the news source and topic and use Beautiful soup to extract the link. Then the link is openned with selenium and the necessary information is extracted. Once it was possible, we wanted to obtain the texts only from the daily topics. To get the daily topics, it was installed the Pytrends library. Using Pytrends, it was able to get the daily Google trends for the country in question (Portugal). Once these trends were obtained, it was consulted the Público newspaper API and, using the current date and the Pytrends trending keywords, it was searched the API for all the recent news stories on the desired topic. The aim is to go to all the desired APIs or subsequent web crawlers.

Once the texts from the Público newspaper were saved, a prompt has been built, so that our LLM model (GPT chat) could create a news story using the desired texts. Firstly, it was found how to connect to the ChatGPT API. To do this, it was necessary to log in to an account and

generate a token to work with the API. After generating this token, it was followed the OpenAI documentation and managed to connect to ChatGPT. Through one of the API service calls, two instructions are sent: it is sent to the system what we want it to be (the role it plays) and an LLM prompt with the description of the expected result, i.e. a news item generated from the texts provided. After sending the texts in the prompt, the Chat GPT API returns the complete news item.

4.2.1 Project Structure

This project was, as has been discussed throughout this document, with the Django framework in the Python language. In the root folder of the computer, a folder was created with the name Generation of Breaking News Content using AI (LLM) and SEO, and so a project was created with the name brekingNews_website and an app with the name Home. The environment was given the name venv. In this project, a folder called templates was created within the app with the following files: detail_news.html, generated_news.html, home.html, keyword.html, login.html, menu.html, news.html, register.html, sources.html, trends.html.

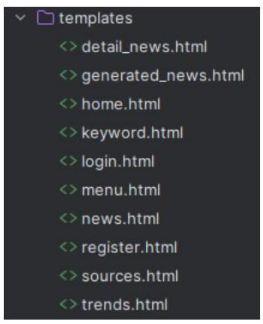

Figure 4.12: Template Folder.

The login.html and register.html files are part of the project's login system. The first file is used to log in to the platform and the second file is used to register again if necessary. The menu.html file is present in all .html files as it is the project's nav-bar menu, so it is needed on all pages. The home.html is the main page of the project and from there the user can navigate to keyword.html or trends.html. In keyword.html the user can create news using expressions or several words, while in trend.html the user creates breaking news using daily trends. From these pages, the user is taken to the sources.html file to select the news sources and also configure their news item, i.e. whether they want a complete news item or just a topic-based news item. After creating the news item, the user is taken to the generated_news.html file where they can find their news item generated using LLMs. In News.html, the user will be able to see all the news they have created and in detail_news.html the user will be able to see in detail the news they have already created.

In addition to this folder, a folder called Static has been created, inside which is another folder called CSS and inside which is a style.css file. All the project's styles are stored in this file. In the project's settings.py file, **STATICFILES_DIRS = [BASE_DIR / "static"]** had to be set for the css to work correctly in the project.

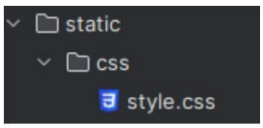

Figure 4.13: Styles Folder.

As for the python files, the following files were created: generate_news.py, GPT.py, jn.py, publico.py, trends.py, urls.py and of course the files previously generated when creating this Django project.

Some of the most important files in the project are generate_news.py and GPT.py. In generate_news.py you get all the text from the sources and call the OpenAI API. GPT.py is where all the news is built with the necessary API calls and text selection. Files jn.py and publico.py are the sources where the texts are fetched so that the API receives them. Trends.py is where the daily trends will be popularised on our platform. urls.py was a file that had to be created in order to create the paths for the views. The models.py file is where all the database tables are created, such as the news table, etc. Another very important file for the project.

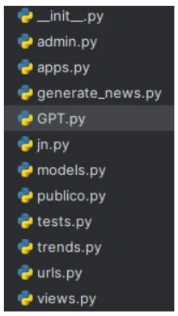

Figure 4.14: Python files.

In line with the text above, here's a picture of all the files in these products so that you can understand them better.

4.2.2 Python libraries used in the Project

4.2.3 PostgreSQL and AWS (Amazon Web Services)

PostgreSQL is an open source object-relational database management system (ORDBMS). It is known for its robustness, extensibility and compliance with SQL standards. PostgreSQL guaran-

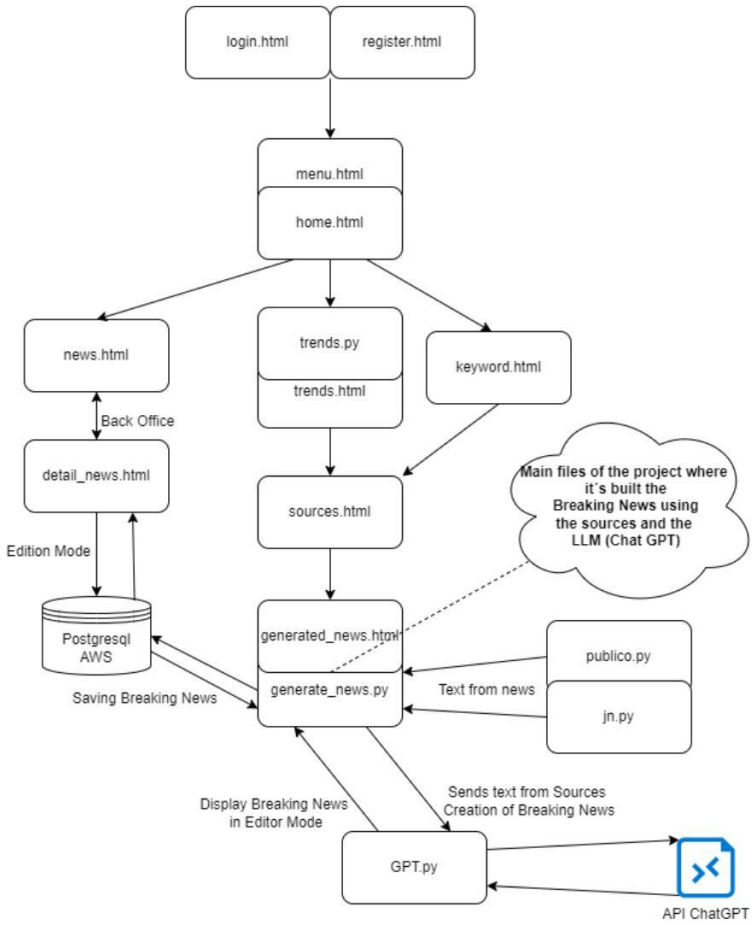

Figure 4.15: Software components of the proposed solution.

tees that database operations are carried out reliably and allows new functionalities to be added via extensions. Supports advanced data types such as JSON, XML and custom types and advanced indexing, parallel queries and optimisations for large volumes of data [Scott, 2024].

Amazon Web Services (AWS) is a cloud computing services platform offered by Amazon. AWS offers a wide range of services, including computing, storage, databases, networks, analysis and much more. Some of the most popular services include: Amazon EC2, Amazon S3 and Amazon RDS (which supports various database management systems, including PostgreSQL) [Amazon, 2024].

It was configured the project to connect to a PostgreSQL database hosted on Amazon RDS (Relational Database Service).

First, it was created a PostgreSQL database in AWS RDS and then chosen PostgreSQL as the database management system.

To connect Django to the PostgreSQL database on AWS RDS, it has been configured Django's settings.py file as follows:

```
    DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'postgres',
        'USER': 'masteruser',
        'PASSWORD': '###',
        'HOST': 'bnw-project.c9i20cqseg8f.eu-north-1.rds
        .amazonaws.com',
        'PORT': '5432'
    }
}
```

Let's take a closer look at each field:

- ENGINE: Specifies the database backend that Django should use.

  In this case, it's django.db.backends.postgresql for PostgreSQL.

- NAME: The name of the database that was created in AWS RDS.

- USER: The name of the user with privileges to access the database.

- PASSWORD: The user's password.

- HOST: The endpoint of the PostgreSQL database on AWS RDS. This is the address that Django will use to connect to the database.

- PORT: The port on which PostgreSQL is running. The default is 5432.

Integrating a Django project with a PostgreSQL database hosted on AWS RDS offers several advantages, including the scalability, high availability and ease of maintenance provided by AWS. It uses environment variables or secret management services to store sensitive credentials and sets up automatic backups in AWS RDS to ensure that our data can be recovered in the event of a failure. Also it´s possible to take advantage of RDS's scalability options to increase or decrease database resources as needed.

4.2.4 ChatGPT API

The ChatGPT API, provided by OpenAI, allows advanced language models to be integrated into various applications. These models, such as GPT-3.5-turbo, can be used for tasks such as text generation, translation, summarisation and much more. The API is accessible via HTTP and supports authentication via API keys. It supports the creation of text from supplied prompt and continuation of a text started by the user. The API maintains the context of a conversation, responding in a manner consistent with previous messagesand adjust parameters such as temperature to control the creativity and variability of responses [Ali, 2024].

A typical request to the completions API contains the following components:

- Endpoint URL: https://api.openai.com/v1/chat/completions
- Headers:
  - Authorisation: includes the API key.
  - Content-Type: application/json
- Body (JSON):
  - model: The name of the model to be used (e.g. gpt-3.5-turbo).
  - messages: A list of messages representing the conversation.
  - temperature: A value between 0 and 1 that controls the creativity of the response (lower temperatures result in more deterministic responses).

In this project, it´s been user the ChatGPT API to generate breaking news topics in Portuguese based on a supplied text.

We're defining a specific prompt that configures the model's role as a journalist and provides the necessary instructions for generating news topics, or it can be a full text news, depends on the user. In this case it will be used the topics scenario.

```
prompt = [
    {
        "role": "system",
        "content": "You are a journalist that creates breaking
        news for a newspaper"
    },
    {
        "role": "user",
        "content": f"Create a list of topics, which each topic
        must be in separate lines for breaking news article in
        Portuguese based on this text: {text}. Please ignore any
        images or links, just focus on the theme and content of
        the text."
```

```
        }
]
```

The API call is made using an OpenAI client configured to communicate with the completions endpoint. Below is the code used to make the request:

```
    chat_completion = client.chat.completions.create(
    messages=prompt,
    model="gpt-3.5-turbo",
    temperature=0.2
)
```

The result of the API call is stored in the final_news variable, which contains the list of topics generated by the model:

```
    final_news = chat_completion.choices[0].message.content
```

The model is configured to assume the role of a journalist creating breaking news it´s asked to create a list of topics or full text based on the text provided, ignoring images and links, focussing only on the theme and content. The temperature is set to 0.2, ensuring that responses are more deterministic and less creative, which is desirable for consistency in news topics. The response generated by the model is extracted and used as required in the application for the final news.

Integrating the ChatGPT API into this project allowed the automatic generation of breaking news topics or full text in Portuguese. This system can be expanded and adjusted as required to meet different journalistic or content generation needs.

4.2.5 Login System

The Django framework offers a robust authentication and authorisation system that can be easily integrated into web applications to manage user access. In this section, is discussed the implementation of a login system that requires authentication for all views, using Django's standard User table. We'll include configuring the system, creating login and logout pages, and applying the authentication requirement to all views. Django comes with a ready-to-use user model, which is managed by the auth application. There is no need to create a customised user model for basic authentication functionality. First it was created a template for the login page, such as login.html, within the application's templates directory. In the application's views.py file, it was imported and configured the LoginView and in the urls.py file, it was created the route to the login view. For the logout, it was used Django's generic LogoutView and it was added a logout button to a template, in the menu.html. To require all views to be accessible only by authenticated users, is used Django's login_required decorator.

In the views.py file, it was added the login_required decorator to the views that require authentication. To apply protection to all views efficiently, it was configured LOGIN_URL in the settings.py file, which defines where users will be redirected to if they are not authenticated.

With these settings, the authentication system is complete and all protected views will require users to be authenticated to access them, and users will be able to log in and out via the configured pages. This basic authentication system was expanded with the functionality for registering new users, resetting passwords, user profiles, etc. Django provides several tools and generic views that make it easy to implement these additional functionalities.

4.2.6 Django Admin

One of Django's most powerful and useful components is the administrative interface **(/admin)**, which provides a ready-to-use tool for managing application data. This functionality is particularly useful for developers and system administrators who need to manipulate data efficiently without the need to develop customised administration interfaces. Django's administrative interface offers a series of robust functionalities that make data management easier. The administrative interface allows you to create, read, update and delete (CRUD) records from any registered model. This eliminates the need to develop CRUD functionalities from scratch. Using ModelAdmin classes, you can customise the display and behaviour of the administrative interface. Settings such as field listing, search filters, customised actions and form layouts can be adjusted to meet the specific needs of each application. Django Admin integrates with Django's authentication system, providing granular control over who can access and modify data. Permissions can be set for different access levels, guaranteeing data security and integrity.The administrative interface includes powerful filtering and search tools, allowing users to quickly locate records of interest. Customised search fields and specific filters can be added to improve usability. In addition to standard CRUD operations, Django Admin lets you create customised actions that can be applied en masse to records [DjangoProject, 2024]. This is useful for common operations such as exporting data or batch updates.

Django Admin offers several benefits that make it a popular choice among developers:

- Speed of Development: The out-of-the-box administrative interface speeds up development, allowing developers to focus on other parts of the application.
- Consistency: Django Admin follows the best design and usability practices, providing a consistent experience for users.
- Extensibility: Through customisation, it is possible to adapt Django Admin to meet various specific needs.

Despite its many advantages, Django Admin also has some limitations:

- Performance: For very large databases or complex operations, performance can become an issue, requiring additional optimisations.
- Advanced customisation: Although it is very customisable, some very specific customisations can be difficult to implement, requiring extensions or alternative solutions.

- Appearance: The standard Django Admin interface may not be suitable for all applications in terms of design, and may require CSS adjustments for better visual integration with the application.

The Django admin interface is a powerful tool that offers a complete and extensible data management solution. Its integration with Django's authentication system, combined with the ability to customise it using ModelAdmin classes, makes it an efficient choice for many projects. However, it is important to consider its limitations and assess the need for additional customisations to fully meet the needs of each application. In short, Django Admin is a valuable component that can significantly speed up the development and improve the maintenance of Django-based web applications.

4.2.7 Superuser - Django

A superuser in Django is a special type of user who has full administrative permissions, allowing them to access and modify all the data in the system, as well as manage other users and their permissions.

Creating a superuser in Django is a simple process, facilitated by a command line command. During development, superusers are usually created to configure and manage the application

In Django, permissions are managed through the authentication and authorisation system. Each user can have specific permissions to create, read, update and delete (CRUD) objects in the system. The superuser, by definition, has all these permissions without restriction. The superuser can access the administrative interface, where they can view and manage all registered models and also can create, edit and delete other users and groups, as well as assign permissions to them. The superuser can modify any data in the application, regardless of the permissions associated with specific models or fields [DjangoProject, 2024].

Granting superuser privileges must be done carefully due to the high level of access and control it provides. Superusers should have strong and secure passwords to prevent unauthorised access. Django includes password strength checks to help with this. Assigning superuser status should be restricted to trusted individuals, due to the potential for abuse of power and activities carried out by superusers should be monitored and audited regularly to ensure that there are no malicious or unauthorised actions.

The superuser facilitates the centralised administration of a Django application. Through the administrative interface, a superuser can manage all aspects of the application efficiently. During the initial configuration of a Django application, the superuser is essential for defining templates, configuring the data structure and adjusting permissions. This speeds up development and ensures that the application is ready for use quickly.

The existence of superusers allows great flexibility in the administration of the application. New functionalities can be added and configured without the need to develop additional administrative interfaces.

Despite its advantages, the superuser also has some limitations and points of caution. The superuser has full access to the system, which represents a significant risk in the event of the account being compromised and also inadequate permissions management can lead to an environment where many users have excessive privileges, increasing the risk of errors or abuse.

The superuser in Django is a powerful and essential tool for the effective administration of web applications. With full permissions and unrestricted access, it allows centralised and efficient management of data and users. However, due to the high level of access, it is crucial to implement strict security measures to protect the integrity of the system. When used correctly, the superuser can significantly improve the efficiency of administering a Django application.

4.2.8 Users Model - Django

Django, the popular Python web development framework, includes a robust and extensible authentication system that manages user creation, authentication and authorisation. At the heart of this system is the user model, which serves as the basis for managing user accounts in Django applications.

Django's standard User model is located in the django.contrib.auth.models module. It includes essential fields for authenticating and managing user profiles, such as:

- Username: A unique identifier for the user.
- Password: Stored securely using hashing.
- Email: The user's email address.
- First Name and Last Name: The user's first and last name.
- Date Joined: Date the user was created.
- Last Login: Date and time of the user's last login.
- is_active: Indicates whether the user's account is active.
- is_staff: Indicates whether the user can access the administrative site.
- is_superuser: Indicates whether the user has all permissions.

The User model also integrates a permissions system, allowing each user to have different levels of access. This is managed through groups and individual permissions that can be assigned to users. The standard User model covers most authentication and user management needs, speeding up development and has the ability to customise the User model allowing developers to adapt the authentication system to the specific needs of their applications. Django implements good security practices, such as secure password storage and protection against brute force attacks [DjangoProject, 2024].

Django's user model provides a robust foundation for authenticating and managing user accounts in web applications. Its out-of-the-box structure and customisation capabilities make it a

versatile choice for developers. By allowing the addition of customised fields and the configuration of specific permissions, Django's User model can be adapted to a wide range of application scenarios, guaranteeing security and efficiency in user management.

4.2.9 Building the Project with SEO Guidelines

SEO is a set of practices and techniques used to improve the visibility of a website in the organic results of search engines such as Google, Bing and Yahoo. Good SEO increases the likelihood that content will be found by a larger and more relevant audience. Such tools as Google Keyword Planner, SEMrush, and Ahrefs to identify relevant keywords that the target audience is searching for and also incorporate keywords strategically into the title, subheadings, body text, meta descriptions and URLs to optimise the relevance and ranking of the content in search engines.

It´s important to use appropriate HTML tags (H1, H2, H3, etc.) to structure the content hierarchically, making it easier to read and understand for both users and search engines. Create attractive and informative descriptions that encourage users to click on the link when the content appears in search results. It´s appropriate to ensure that URLs are short, descriptive and contain relevant keywords to improve indexing and visibility [Google, 2024].

It is important to optimise the site's loading time using techniques such as image compression, minification of CSS and JavaScript files and the use of content distribution networks (CDNs). It was ensured that the site is accessible and navigable on mobile devices, since a significant proportion of search traffic comes from smartphones and tablets.

Improving the navigation and structure of the site to provide a better user experience, increasing the time spent on the site and reducing the bounce rate was a assured asset.

The integration between the automatic generation of trending news with LLMs and SEO optimisation involves a series of coordinated steps to ensure that the content is not only relevant and of high quality, but also highly visible in search engines.

Using LLMs to generate articles and news items that incorporate the identified keywords in a natural and fluid way and developing scripts in Python/Django to automatically insert SEO tags, meta descriptions and other optimisation elements into the content generated, ensuring that each piece of content is optimised from the moment it is created.

Another guideline would be monitoring content performance in terms of search rankings, organic traffic, click-through rate (CTR) and other relevant metrics and use feedback and analytical data to continually adjust SEO strategy and content generation, refining techniques and approaches to maximise effectiveness [Google, 2024].

Below, the implementations guidelines, with HTML, CSS and Django, are listed:

- HTML: Structure content in a semantic way using appropriate HTML elements, such as <article>, <section>, <header>, <footer>, to improve accessibility and indexing by search engines.
- CSS: Styling the content to ensure a good user experience and adherence to SEO best practices, such as responsive design and fast loading times.

- Content Management: Create a CMS (Content Management System) in Django to manage the creation and publication of content in an efficient and scalable way.
- SEO automation: Develop functions and scripts to automatically generate meta tags, sitemaps, robots.txt files and other essential SEO elements.
- API integration: Implement APIs to collect trend data, monitor SEO performance and integrate analysis and monitoring tools.

The integration between creating trending news with LLMs and search engine optimisation (SEO) offers a robust approach to increasing the visibility and impact of digital content. Using HTML, CSS and Django, it is possible to build an efficient platform that automates both content generation and SEO optimisation, ensuring that news stories not only reach a wider audience, but also provide a high-quality user experience.

This synergy is essential to any modern digital marketing strategy, allowing organisations to maximise their reach and effectiveness in communicating with their target audience. With the combination of LLMs' advanced capabilities and SEO best practices, it is possible to create a dynamic, responsive and highly optimised content ecosystem that meets the needs and expectations of the contemporary digital market.

Chapter 5

Conclusions

In this chapter, the main difficulties faced during the development of this study are adressed, including challenges in data collection, technological limitations and the impact of external factors, such as access to APIs. We will discuss how these obstacles influenced the progress of the research and the strategies adopted to overcome them.

In addition, we will present the conclusions obtained, highlighting the main findings and their relevance to the field of study. Finally, we will explore future directions for the research, suggesting possible improvements and new areas of investigation that have emerged from the results achieved.

5.1 Difficulties

The development of the credible news generation platform using LLM and SEO presented several significant challenges. These challenges were overcome over the course of the project, but they played a crucial role in shaping the technical decisions and the approach adopted. One of the biggest difficulties was integrating reliable news portals into the platform. This process required a thorough understanding of the APIs and data formats used by different news sources. Each news portal has its own data structure, access policies and limitations, which complicated the task of creating a unified interface for collecting information. In addition, guaranteeing the authenticity and credibility of the sources was a constant challenge. It was necessary to establish strict criteria for selecting sources and implement verification mechanisms to ensure that only high-quality data was used. Maintaining these integrations, with the constant updating of APIs and portal policies, also proved challenging, requiring a flexible and adaptable approach. The project involved learning a new programming language, which represented a significant challenge. The learning curve was steep but this challenge was addressed through intensive training and ongoing practice. Understanding the nuances and best practices of the new language was crucial to writing efficient and secure code. This process required time and patience, but was fundamental to guaranteeing the platform's robustness and scalability. Adapting to the new language also involved familiarising oneself with its ecosystem of libraries and tools, which in turn influenced the system's design

and architecture decisions. In addition to the new language, adopting a new development framework presented its own challenges. The choice of framework was based on its ability to meet the specific needs of the project, but its practical implementation revealed complexities not initially foreseen. It was needed to familiarise with the framework's architecture, its main components and the best practices for using it. This process involved reading extensive documentation, and carrying out experiments to understand how to adapt the framework to our needs. The integration of different modules and the resolution of framework-specific problems were activities that required significant effort. The integration of Large Language Models (LLM) was one of the most technical and challenging components of the project. LLMs are powerful tools, but their efficient integration requires a thorough understanding of their inner workings and the best practices for their use. Training and optimising the LLMs to generate coherent and unbiased news content required a rigorous process of experimentation and adjustment. Considerable effort was required to adjust model parameters, select appropriate datasets and implement bias reduction techniques. In addition, ensuring that the LLMs operated in real time, providing fast and accurate responses, was a significant technical challenge. The integration of a robust and scalable database was crucial for the efficient storage and management of the data collected and generated by the platform. This process presented challenges related to choosing the right database management system (DBMS), data modelling and implementing efficient read and write operations. Ensuring data integrity and security was a priority. This required the implementation of strict access and control policies, as well as the adoption of best practices for data backup and recovery. Database scalability was also a critical aspect, since the platform needs to handle large volumes of data and serve many users simultaneously. To overcome these challenges, it was adopted a collaborative and iterative approach. The use of agile methodologies made it possible to adapt quickly to changes and solve problems effectively.

In addition, the implementation of test-driven development (TDD) practices and process automation were crucial strategies to guarantee the quality and reliability of the system. Collaboration with experts in AI, SEO and journalism also provided valuable insights that helped shape the platform to meet the specific needs of users.

To summarise, the difficulties encountered during the development of the platform were significant, but provided valuable opportunities for learning and growth. Each challenge overcome contributed to the robustness and functionality of the platform, resulting in a tool capable of generating credible and impartial news content. The lessons learnt throughout this process will be essential in guiding future developments and improvements to the platform.

5.2 Discussion and Conclusions

With the advance of technologies in the digital world and the speed with which information reaches us, verifying the authenticity of news has become a significant challenge. This project was motivated by the growing concern about fake news and aimed to develop a platform that would

support the creation of credible, fast and unbiased news content, using Artificial Intelligence (AI) and search engine optimisation (SEO) techniques.

The construction of the platform followed a systematic approach to guarantee the integrity and impartiality of the news generated. The first step was to identify and integrate reliable news sources. These sources were selected based on their reputation, track record of accuracy and impartiality. From these sources, the system collects relevant information, ensuring that the data used to generate the news is of high quality and reliable.

The core of the platform is a Large Language Model (Chat GPT), which has been trained to generate coherent and well-structured texts. This model uses the data collected from reliable sources to create news, which are then transformed into full texts. The language model's ability to understand and synthesise large volumes of information enabled the creation of detailed and informative content, while maintaining the impartiality required for reliable news. One of the platform's distinctive features is its ability to monitor trends in real time. Using tools such as Google Trends, the platform identifies highly relevant topics, enabling a rapid response to important events. This feature ensures that users have access to up-to-date and relevant information, which is crucial in a news environment where speed and accuracy are essential. Another key aspect of the project is the functionality that allows users to edit and approve the texts generated. This feature ensures that editorial control remains with the user, allowing for the necessary adjustments for different contexts and needs. The user can review the generated text, make changes and finally approve the final version before publication. This flexibility is essential to ensure that the content meets the specific expectations of each user, whether they are a journalist, a blogger or an ordinary individual.

Fake news and the veracity of news represent two extremes of the information spectrum, with significant implications for society and the field of communication. Fake news, often created with the intention of deceiving, manipulating or generating profit, has proliferated rapidly due to the power of social networks and algorithms that amplify sensationalist and polarising content. Unlike real news, which is based on verifiable facts and produced by reliable sources, fake news often lacks adequate sources or is based on distorted information, which makes it difficult to verify.

Combating fake news is particularly challenging due to the speed with which it spreads. By prioritising content that generates the most interaction, social networks end up favouring false, sensationalist or polarising stories, which are shared on a large scale before the information can even be verified. In addition, a lack of digital education and a low level of media literacy make many individuals vulnerable to this type of misinformation, making it difficult to differentiate between what is true and what is false.

On the other hand, real news contributes to building an informed and transparent society. When information is properly verified and based on reliable sources, it has the power to educate, enlighten and promote more informed decisions. Truthful news is an essential pillar for public trust, which in turn underpins democracy and healthy debate. In short, the fight against fake news is a complex issue that involves not only technology and verification techniques, but also a commitment to media education and the promotion of critical information consumption. Only

by combining these efforts will it be possible to reduce the negative impact of fake news and strengthen trust in the veracity of news, which is essential for society to function properly.

The benefits of this platform are manifold. Firstly, it contributes to the fight against fake news by providing a tool that generates content based on reliable and verified sources. Secondly, it facilitates the creation of high-quality news quickly, saving users time and resources. In addition, the platform promotes transparency and accountability, as the user has the ability to edit and approve the content generated.

5.3 Future Work

Although the project has achieved significant results, there are several areas that offer opportunities for improvement and future expansion. Below are some directions for future work. One of the most promising areas for future work is the continuous improvement of AI model accuracy. This can be achieved by training the model with more extensive and diverse data sets. In addition, it is crucial to continue investigating and implementing techniques to reduce biases in the model, ensuring that the content generated is truly unbiased and neutral, expanding the list of reliable news sources is another area of focus. Including sources from different languages and cultures can enrich the diversity of content and make it more inclusive. In addition, integrating local and regional news sources can provide a more complete and detailed perspective on specific events, increasing the platform's relevance and reach. Integrating sentiment analysis tools can help identify and mitigate possible emotional biases in the news generated and the implementation of automatic fact-checking mechanisms can further strengthen the credibility of information. These mechanisms can check the claims contained in the news against a database of known facts, helping to guarantee the accuracy and veracity of the content, developing a more intuitive user interface and adding additional functionalities, such as real-time notifications about new trends, can increase the usability and attractiveness of the platform, the possibility of collaboration between multiple users, allowing newsroom teams to work together to create and edit news, could also be a valuable addition. Carrying out longitudinal studies on the impact of the platform on users' perception of the credibility of the news is crucial. These studies can provide valuable insights into how users interact with the platform and its effectiveness in combating the spread of fake news. In addition, investigating the adoption of the platform in different contexts, such as newsrooms, schools and individual use, can help adapt and improve the platform for different target audiences. In addition to providing a news creation tool, the platform can be expanded to include educational modules that teach users about the importance of fact-checking and how to identify fake news. These modules could be especially useful for young people and those less familiar with the digital world, promoting a culture of critical news consumption.

To summarise, this project has demonstrated the significant potential of AI in creating high-quality journalistic content. The platform developed offers an innovative solution to the problem of fake news, providing a tool that generates fast, impartial and credible content. However, the continuous search for improvements and adaptation to users' emerging needs are essential to maintain

the platform's relevance and effectiveness in combating the spread of fake news. With future work directions identified, this project can continue to evolve and contribute significantly to a more reliable and informed news environment.

Bibliography

[Ali, 2024] Ali, M. (2024). A beginner's guide to using the chatgpt api. Last visited on July 24th, 2024. Available at: https://www.datacamp.com/tutorial/a-beginners-guide-to-chatgpt-api.

[Amazon, 2024] Amazon (2024). Aws documentation. Last visited on July 24th, 2024. Available at: https://docs.aws.amazon.com/.

[Arif, 2024] Arif, A. (2024). When and how to use psycopg2. Last visited on July 24th, 2024. Available at: https://www.timescale.com/blog/when-and-how-to-use-psycopg2/.

[Aïmeur et al., 2023] Aïmeur, E., Amri, S., and Brassard, G. (2023). Fake news, disinformation and misinformation in social media: a review. *Social Network Analysis and Mining*, 13:30.

[Broussard, 2014] Broussard, M. (2014). How artificial intelligence helped me investigate textbook shortages - american journalism review.

[Cruz and Rosado da Cruz, 2020] Cruz, E. and Rosado da Cruz, A. (2020). Design science research for IS/IT projects: Focus on digital transformation. In *15th Iberian Conference on Information Systems and Technologies (CISTI 2020), Spain, June 2020*.

[Dhiman, 2023] Dhiman, D. B. (2023). Does artificial intelligence help journalists: A boon or bane? *SSRN Electronic Journal*. Available at: http://dx.doi.org/10.2139/ssrn.4401194.

[DjangoProject, 2024] DjangoProject (2024). django-admin and manage.py. Last visited on July 24th, 2024. Available at: https://docs.djangoproject.com/en/5.0/ref/django-admin/.

[DocumentCloud, 2024] DocumentCloud (2024). Documentcloud. Last visited on June 12th, 2024. Available at: https://www.documentcloud.org/home.

[Echout, 2024] Echout, M. (2024). What is jquery & why's it the top js library in 2024? Last visited on July 17th, 2024. Available at: https://www.learnenough.com/blog/what-is-jquery.

[Fran, 2024] Fran (2024). What is python used for? 10 practical python uses. Last visited on July 15th, 2024. Available at: https://www.futurelearn.com/info/blog/what-is-python-used-for.

[Franks et al., 2021] Franks, S., Wells, R., Maiden, N., and Zachos, K. (2021). Using computational tools to support journalists' creativity. *Journalism*. Available at: https://journals.sagepub.com/doi/full/10.1177/14648849211010582.

[Google, 2024] Google (2024). Guia de otimização de mecanismos de pesquisa (seo) para iniciantes. Last visited on June 12th, 2024. Available at: https://developers.google.com/search/docs/fundamentals/seo-starter-guide.

[Hayes, 2024] Hayes, A. (2024). Hypertext markup language (html): What it is and how it works. Last visited on July 9th, 2024. Available at: https://www.investopedia.com/terms/h/html.asp.

[Hetler, 2024] Hetler, A. (2024). What is chatgpt? Last visited on July 8th, 2024. Available at: https://www.techtarget.com/whatis/definition/ChatGPT.

[Hill, 2024] Hill, A. (2024). Who is a journalist? Last visited on July 7th, 2024. Available at: https://www.writersdigest.com/write-better-nonfiction/who-is-a-journalist.

[Johnson, 2024] Johnson, S. (2024). What is django and what is django used for? Last visited on July 16th, 2024. Available at: https://stxnext.com/blog/what-is-django.

[Karl, 2024] Karl, T. (2024). The complete guide for using the openai python api. Last visited on July 7th, 2024. Available at: https://www.newhorizons.com/resources/blog/the-complete-guide-for-using-the-openai-python-api.

[Kasneci et al., 2023] Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., Stadler, M., Weller, J., Kuhn, J., and Kasneci, G. (2023). Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

[Kotenidis and Veglis, 2021] Kotenidis, E. and Veglis, A. (2021). Algorithmic journalism—current applications and future perspectives. *Journalism and Media 2021, Vol. 2, Pages 244-257*, 2:244–257.

[Kung, 2024] Kung, S. (2024). Tracking keyword trends on google search with pytrends. Last visited on July 7th, 2024. Available at: https://towardsdatascience.com/tracking-keyword-trends-on-google-search-with-pytrends-cf97c43803f6.

[Lermann Henestrosa et al., 2023] Lermann Henestrosa, A., Greving, H., and Kimmerle, J. (2023). Automated journalism: The effects of ai authorship and evaluative information on the perception of a science journalism article. *Computers in Human Behavior*, 138:107445.

[Long, 2024] Long, B. (2024). Advantages of a great wysiwyg editor that takes your lms to the next level. Last visited on July 10th, 2024. Available at: https://www.tiny.cloud/blog/wysiwyg-html-editor-lms/.

[M, 2024] M, R. (2024). What is numpy used for in python? Last visited on July 11th, 2024. Available at: https://www.activestate.com/resources/quick-reads/what-is-numpy-used-for-in-python/.

[Manual, 2024] Manual, T. N. (2024). What is news? Last visited on July 11th, 2024. Available at: https://mediahelpingmedia.org/basics/what-is-news/.

[Muehmel, 2024] Muehmel, K. (2024). What is a large language model, the tech behind chatgpt? Last visited on July 6th, 2024. Available at: https://blog.dataiku.com/large-language-model-chatgpt.

[Nelson, 2017] Nelson, J. L. (Jan. 2017). Is 'fake news' a fake problem? - columbia journalism review. *Columbia Journalism Review*. Available at https://www.cjr.org/analysis/fake-news-facebook-audience-drudge-breitbart-study.php.

[NewsBlur, 2024] NewsBlur (2024). Newsblur. Last visited on June 12th, 2024. Available at: https://www.newsblur.com.

[of Encyclopædia Britannica, 2024a] of Encyclopædia Britannica, T. E. (2024a). news agency. Last visited on July 14th, 2024. Available at: https://www.britannica.com/topic/news-agency.

[of Encyclopædia Britannica, 2024b] of Encyclopædia Britannica, T. E. (2024b). newspaper. Last visited on July 14th, 2024. Available at: https://www.britannica.com/topic/newspaper.

[O'Grady, 2024] O'Grady, B. (2024). Javascript demystified: Why you should learn this essential language. Last visited on July 14th, 2024. Available at: https://codeinstitute.net/global/blog/what-is-javascript-and-why-should-i-learn-it/.

[OpenAI, 2024] OpenAI (2024). Chatgpt. Last visited on June 12th, 2024. Available at: https://chat.openai.com.

[Ouellette, 2024] Ouellette, A. (2024). What is bootstrap: A beginner's guide. Last visited on July 15th, 2024. Available at: https://careerfoundry.com/en/blog/web-development/what-is-bootstrap-a-beginners-guide/.

[Pol, 2024] Pol, T. (2024). Google trends: What it is & how to use the data for seo. Last visited on July 19th, 2024. Available at: https://www.semrush.com/blog/google-trends/.

[Rogers, 2024] Rogers, T. (2024). What is a breaking news story? Last visited on July 6th, 2024. Available at: https://www.thoughtco.com/what-is-a-breaking-news-story-2073757.

[Ronquillo, 2024] Ronquillo, A. (2024). Python's requests library (guide). Last visited on July 15th, 2024. Available at: https://realpython.com/python-requests/.

[S, 2024] S, S. (2024). What is pandas in python? everything you need to know. Last visited on July 17th, 2024. Available at: https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/.

[Scott, 2024] Scott, P. (2024). What is postgresql? everything you need to know. Last visited on July 19th, 2024. Available at: https://www.percona.com/blog/what-is-postgresql-used-for/.

[Signy, 2024] Signy, H. (2024). Writing an opinion piece. Last visited on July 5th, 2024. Available at: https://preventioncentre.org.au/resources/writing-an-opinion-piece/.

[Sirén-Heikel et al., 2023] Sirén-Heikel, S., Kjellman, M., and Lindén, C.-G. (2023). At the crossroads of logics: Automating newswork with artificial intelligence—(re)defining journalistic logics from the perspective of technologists. *Journal of the Association for Information Science and Technology*, 74(3):354–366.

[Small, 2024] Small, N. (2024). What does requests offer over urllib3 in 2022? Last visited on July 22th, 2024. Available at: https://medium.com/@technige/what-does-requests-offer-over-urllib3-in-2022-e6a38d9273d9.

[Sulcas, 2024] Sulcas, A. (2024). Beautiful soup tutorial - how to parse web data with python. Last visited on July 24th, 2024. Available at: https://oxylabs.io/blog/beautiful-soup-parsing-tutorial.

[Thorndyke, 2024] Thorndyke, K. (2024). What is css used for? Last visited on July 14th, 2024. Available at: https://www.codecademy.com/resources/blog/what-is-css-used-for/.

[Tuama, 2024] Tuama, D. O. (2024). Understanding libraries in python: A comprehensive guide. Last visited on July 5th, 2024. Available at: https://codeinstitute.net/global/blog/what-are-libraries-in-python/.

[Unadkat, 2024] Unadkat, J. (2024). Selenium webdriver tutorial : Getting started with test automation. Last visited on July 4th, 2024. Available at: ttps://www.browserstack.com/guide/selenium-webdriver-tutorial.

[Zhao et al., 2023] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2023). A survey of large language models. *arXiv*.