Politécnico
de Viseu

Escola Superior
de Tecnologia
e Gestão de Viseu

Trabalho efetuado sob a orientação de

Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Trabalho efetuado sob a orientação de

Agradecimentos

Ao Professor Doutor Filipe Cabral, meu orientador, pela manifestação de incondicional apoio e disponibilidade, pela compreensão por algumas dilações, pelo aconselhamento assertivo e pelo estímulo permanente, que muito contribuíram para aumentar o desafio e melhorar a profundidade e a clareza da investigação, pela sua amizade.

Aos meus pais e à minha irmã, Joana, pelo amor, carinho e atenção que sempre me deram.

Ao Ricardo, pela presença constante, carinho, incentivo e paciência.

Aos meus alunos dos anos letivos 2021/2022 e 2022/2023, que contribuíram para ser a profissional que hoje sou e que me acompanharam no decorrer deste projeto.

À Escola Superior de Tecnologia e Gestão de Viseu, seus docentes e funcionários, que desde 2016 quando ingressei na Licenciatura em Engenharia Informática, me acompanham neste meu percurso académico.

Agradeço a DEUS por todas as oportunidades concedidas a mim, pela força e tranquilidade nos momentos de fraqueza e dificuldades.

Abstract

Developing new services or improving existing ones is becoming more accessible with the evolution of Natural Language Processing (NLP) techniques. Chatbots are a known example of an NLP-based service; they can interact with humans using text messages or natural language. NLP grants, however, the development of other types of services based on natural languages, such as machine translation, email spam detection, information extraction, content summarization, and question answering. A current need, to develop smart cities projects, is a system that can match content (text) from a project offer description with the candidates description by finding common patterns in different textual descriptions. This project presents an implementation of an automated tool with AI and NLP to match needs and concrete ideas for innovation with the skills and offers of the business sector, including start-ups and entrepreneurs. In sentiment analysis, NLP can be harnessed to recognize and categorize the emotional tone conveyed in textual content, such as project collaborator reviews, customer reviews, or social media posts. The sentiment analysis component in this project establishes a tool for comprehending and categorizing sentiments, for candidates seeking engagement in smart cities projects.

Keywords: natural language processing, nlp, content matching, smart cities, stemming and lemmatization, bag-of-words, Bow, term frequency-inverse document frequency, TF-IDF, TF, IDF, stop words, cosine similarity, flask, sentiment analysis, polarity

Contents

List of Figures

List of Abbreviations

AI Artificial Intelligence

ATS Applicant Tracking Software

BoW Bag-of-Words

DL Deep Learning

EDP Engineering Design Process

GUI Graphical User Interface

HR Human Resources

HRM Human Resources Management

ICT Information and Communication Technologies

IDF Inverse Document Frequency

IDSS Intelligent Decision Support System

IoT Internet of Things

IR Information Retrieval

ML Machine Learning

NER Named Entity Recognition

NLP Natural Language Processing

NLTK Natural Language Toolkit

RD Research and Development

SVM Support Vector Machine

TDMs Term-Document Matrices

TF Term Frequency

TF-IDF Term Frequency - Inverse Document Frequency

VADER Valence-Aware Dictionary and Sentiment Reasoner

Chapter 1

Introduction

This section provides background information on using NLP for Information Retrieval (IR), content matching and sentiment analysis. The focus of this research is discussed and justified, the overall research aim is outlined, and research methods are outlined, as is the value of this research.

1.1 Background Information

Scientific and technological developments in the field of Information and Communication Technologies (ICT) are the main contributors to the sustainable growth of cities worldwide. Smart cities global market is growing quickly, in 2022 it had an estimated growth of 482 billion euros[25]. The evolution of 5G and the Internet of Things (IoT) technologies have contributed to improve the efficiency of urban services by addressing some of the main gaps in energy, mobility, security, privacy and environmental sustainability. However, there is still a journey ahead in this field.

One of the branches of AI and linguistics is NLP, whose function is to make computers understand the statements or words written or spoken in human language. Information extraction helps to collect information from machine-readable documents automatically, so it is a key step in NLP.

Organizations are constantly evolving into more complex environments due to global competition, creating the need to have partners and service providers that meet their needs and allow them to improve the quality of their services. Human resources (HR)

teams face various challenges, including the difficulty in identifying the most relevant partners, limited availability of data or resources to assess potential partners, and the lack of transparency or standardization in the selection process. The mismatch in expectations between urban needs/ideas and the business sector, limited knowledge or awareness of potential partners among urban stakeholders, and inefficient or ineffective communication channels can further complicate the process. One critical challenge in this context is the amount of time it takes to choose the ideal partner.

The consideration of AI models, such as ChatGPT, in the implementation of this tool represents a strategic incorporation of these innovations. This not only reflects significant technological progress but also opens new perspectives to optimize and enhance communication channels. By taking into account these advanced models, the tool seeks to capitalize on the advantages offered by these cutting-edge technologies, providing a more sophisticated approach aligned with the ongoing innovations in the field of AI.

1.2 Overall Research Aim

The "City Catalyst - Catalyst for Sustainable Cities" project aims to explore and develop innovative products, processes, and services that have significant potential to contribute to integrated urban management, efficiency, and innovation. This work intends to make specific contributions to the implementation and interoperability of urban platforms, with the ultimate goal of fostering sustainable urban development.

Start-ups can take advantage to leverage the need of established companies to maintain their place in the market while being more innovative and competitive. New companies can bring this innovation and maintain high-quality standards. However, it is crucial that the establishment of a partnership is quick and practical to keep up with the current market trends.

Figure 1.1 is a visual representation of the process and help stakeholders better understand how the contract with a startup or company will be developed and managed to create a smart waste management for a city.

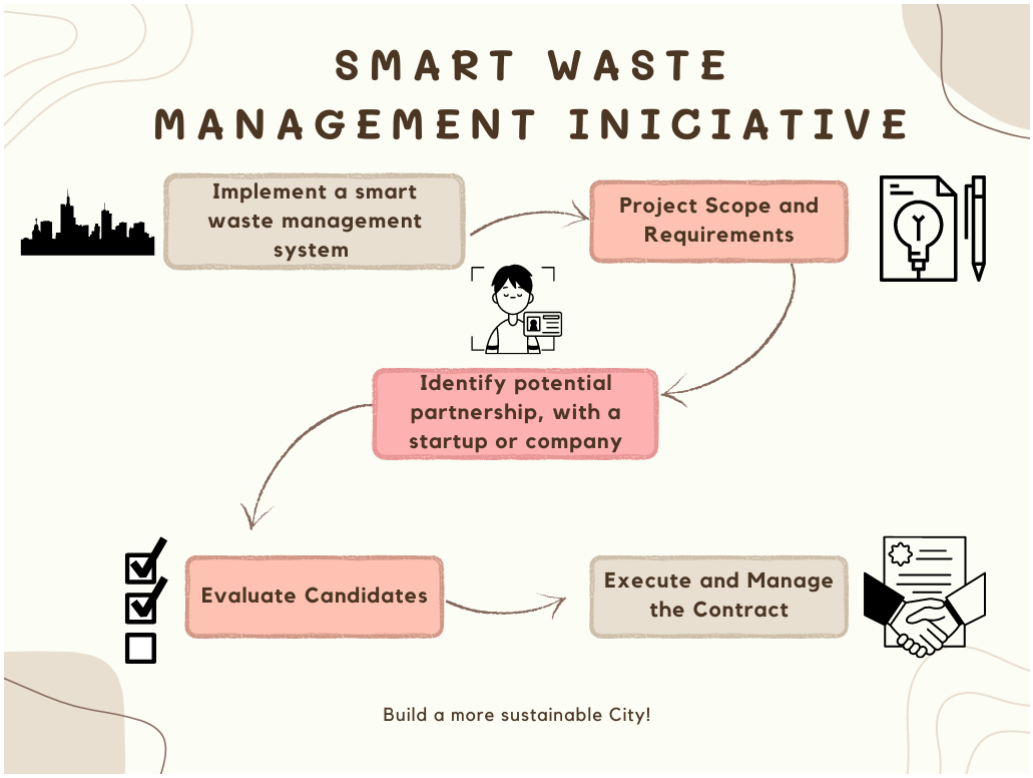

Figure 1.1: Example of a Smart City Project

This work aims to use AI and NLP techniques to implement an automated matching tool to connect needs and specific ideas for innovation within cities with the skills and offers of the business sector, including start-ups and entrepreneurs.

For candidates aspiring to contribute to smart cities projects, a sentiment analysis tool might be a valuable resource. NLP plays a pivotal role in deciphering and categorizing the emotional tone embedded in textual content, spanning from feedback given by project collaborators to customer reviews and social media posts. This analytical capability facilitates a nuanced comprehension of sentiment within the smart cities ecosystem. It allows candidates to gain valuable insights into the emotional dynamics associated with collaborative endeavors, the reception of the project, and the public perception of smart city initiatives.

1.3 Research Methods

As previously mentioned, the primary objective of this project is to create an innovative tool that will assist entities in finding suitable partners to collaborate on the development

of services that will make a significant contribution to society.

The methodology that best fits the needs of this projects is the Research and Development, in specific the Engineering Design Process (EDP).

EDP has a series of steps that can be followed to develop a solution to a problem. Since this project involves designing, building and testing an hypothetical solution this is the methodology that best suits.

1.4 Dissertation Structure

The project begins with a study of previous research and projects using NLP. After this analysis it will be possible to select techniques and approaches to solve the problem more effectively and efficiently.

The remainder of this work is organized as follows. Section 2 presents the state-of-the-art review, it is organized in five subsections to explore the core themes of this project. Section 3 presents the selected methodology for the project. Section 4 presents the steps of the development of the proposed tool and the results obtained. And to finish, Section 5, with the conclusions and proposed future work.

Chapter 2

State-of-the-Art

NLP is a sub-field of computer science and AI that focuses on the interaction between computers and human language. It involves developing algorithms and techniques to analyze, understand, and generate human language, and has a wide range of applications in fields such as machine translation, text classification, and conversation systems.

In the context of content matching and sentiment analysis, NLP plays a key role in understanding and analyzing language. To initiate a profitable partnership , NLP techniques can be used to extract key information from project descriptions or resumes, and to classify project postings or candidates based on their language and content. In sentiment analysis, NLP can be used to identify and classify the sentiment expressed in text, such as worker reviews, customer reviews or social media posts.

The purpose of this section is to provide an overview of the current state of research in NLP applied to content matching and sentiment analysis. This section will describe the most important and relevant studies, theories, and methods that have been developed in these areas, as well as any current debates or controversies. The section will be divided into several sections, including NLP and its techniques, sentiment analysis, and the intersection of these two areas.

2.1 Natural Language Processing - Concept and Techniques

The increasing number of internet-connected gadgets, websites, apps, and social media platforms has led to a significant surge in the amount of data gathered online in recent

years. This wealth of information, predominantly in text format, encompasses various sources such as emails, social media posts, papers, and recordings of spoken conversations. To make sense of this vast corpus of data, NLP techniques have been developed and employed. NLP involves the utilization of computer algorithms to understand and process human language, enabling machines to interpret, analyze, and derive insights from textual information. By leveraging NLP, organizations can unlock the potential of the data they acquire online, enabling them to make informed decisions, improve customer experiences, and drive innovation.

NLP can be applied to various aspects of the talent acquisition process. One area where NLP can be particularly useful is in resume parsing. This involves extracting important information from resumes, such as job titles, skills, and work experience. By using NLP algorithms, this information can be automatically extracted and organized, making it easier for HR teams and hiring managers to review and assess candidates. This not only saves time but also ensures that no relevant information is overlooked during the initial screening phase. In the next step of the evaluation process, candidate responses can be analysed, whether through interviews, assessments, or written submissions, NLP algorithms can assess language patterns, sentiment, and overall communication skills. This can provide valuable insights into a candidate's suitability for a role, helping recruiters make more informed decisions.

Collaborator feedback is a valuable resource for organizations, as it provides insights into the experiences, perspectives, and concerns of their workforce. Traditionally, analyzing employee feedback has been a time-consuming and labor-intensive task, requiring manual reading and categorization of large volumes of text. This is where NLP can make a significant difference. By applying NLP techniques to employee feedback, organizations can gain valuable insights in a more efficient and systematic manner. NLP algorithms can process and analyze large amounts of text data, allowing organizations to identify trends, patterns, and sentiments in the feedback. This can help uncover common themes and issues that might otherwise go unnoticed.

For example, NLP can be used to identify recurring topics in employee feedback, such as work-life balance, communication, leadership, and career development. By analyzing the frequency and sentiment associated with these topics, organizations can gain a better

understanding of the needs and concerns of their employees. This information can then be used to inform decision-making and drive improvements in the workplace.

2.1.1 NLP - Techniques for text Preprocessing

Text preprocessing is an important step in NLP that is performed on raw text data to clean, convert, and normalize it so that it can be effectively processed and analysed by NLP models.

There are several common text preprocessing techniques that are used in NLP, according to [12]:

- Text Normalization: involves converting text into a more standardized and convenient form. This includes tasks like tokenization (separating words or other units), emoticon and hashtag tokenization, word lemmatization (finding the common root of words), sentence segmentation (breaking text into sentences), and string comparison using metrics like edit distance.
- Regular Expressions (Regex): are a practical language used for specifying text search patterns. They allow for efficient searching and matching of text based on specified patterns. Regular expressions are widely used in computer science, programming languages, and text processing tools for tasks such as searching, matching, and extraction.
- Named Entity Recognition (NER): the task of identifying and classifying named entities in text. NER is closely related to tokenization, as it involves detecting and categorizing specific named entities within text.

A comprehensive introduction to NLP using the Python programming language is given in [5]. The book provides a detailed discussion of text preprocessing, which is an essential step in preparing text data for further NLP analysis. The book covers many of the standard techniques used in text preprocessing such as tokenization, stemming, and lemmatization, as well as stopword removal. It also covers more advanced techniques like regular expressions, text normalization, and string manipulation which are crucial in NLP preprocessing. Beyond text preprocessing, the book also covers several other important topics in NLP, such as:

- Part-of-speech tagging (POS): the process that involves automatically determining the grammatical category of each word within a sentence. By assigning the appropriate part-of-speech tag, such as noun, verb, adjective, or adverb, POS tagging helps to unravel the grammatical structure of text.

- Syntactic parsing: the process of determining the grammatical structure of a sentence. It goes beyond the basic task of POS tagging and delves into understanding the relationships between words in a sentence. This document aims to provide an overview of syntactic parsing, highlighting its significance and its complexities.

- Semantic analysis: the process that involves deciphering the meaning of text. This task is inherently complex as it necessitates comprehending the intricate relationships between words, phrases, and sentences, while also considering the broader context in which they are situated.

All these topics are related to each other, for example, syntactic parsing depends on POS tagging and Semantic analysis depends on syntactic parsing. The book covers these topics in a logical order, starting with the fundamental concepts and progressing to more advanced topics. Each chapter provides a good introduction to the topic and includes several examples to demonstrate how to use Natural Languege Toolkit (NLTK) to perform the different NLP tasks in Python.

NER is crucial in information extraction; it is used in machine translation, question answering, IR, and summarization. NER is a task that aims to identify text that mentions entities that are real-world objects that have a name, such as a person, organization, location, date, time, or product. The study [24] discusses the challenges and limitations in evaluating and comparing NER software. Lack of data and the inability to reproduce the evaluation presented obstacles for researchers attempting to replicate experiments in NER, making it challenging to judge the efficacy and completeness of the results. These limitations highlight the pressing need for more standardized and replicable evaluation methodologies to ensure reliable and meaningful comparisons among NER software solutions.

Considering the issue of analyzing large amounts of text data, the authors of [2] conducted a review of existing approaches for generating summaries of large texts using NLP.

They discussed structured-based approaches, which rely on the structure of the text to generate summaries, and semantic-based approaches, which rely on the meaning of the text to generate summaries. It was concluded that the summaries often fall short of expectations and may not align with the original document. No specific model has emerged as the best for generating summaries, indicating the need for further advancements.

The authors of the paper [22] evaluate the performance of different text preprocessing techniques on a dataset of tweets, using several Machine Learning (ML) algorithms such as Naive Bayes and Support Vector Machine (SVM). They reported the results of the evaluation, showing that text preprocessing techniques are crucial for improving the performance of sentiment analysis tasks on social media data.

The results of the evaluation showed that the text preprocessing techniques that are applied to social media data can significantly affect the performance of sentiment analysis tasks. They found that the combination of tokenization, stemming, and lemmatization with stopword removal and punctuation removal were the best preprocessing technique for sentiment analysis on social media data. They also found that the use of regular expressions, text normalization, and string manipulation techniques further improved the performance of the sentiment analysis task.

The authors also found that handling specific features of social media data during the preprocessing stage, such as emojis, hashtags, and URLs, can significantly improve the performance of sentiment analysis tasks on social media data. They also reported that, while stemming performed well, lemmatization was not effective in the choosen dataset, and that the use of a POS tagger did not improve the results.

"The Role of Text Pre-processing in Sentiment Analysis of Social Media Posts" by Waleed Aman, is a research paper which presents a detailed overview of preprocessing techniques used in sentiment analysis of social media posts. the paper describes the text cleaning and preprocessing stages and how these stages affect the accuracy of the sentiment analysis process.

The Named Entity Recognition (NER) has a crucial role in information extraction, it is used in machine translation, question answering, IR, and summarization [24].

NER is a task that aims to identify text that mention named entities and then classify them into predefined categories, for example, client, address, etc. In [17], a survey was

conducted to review recent studies about deep learning applied in NER systems. It was concluded that DL-based NER has as challenges, data annotation and informal text and unseen entities. The first challenge arises because this systems require big annotated data for training, but it is time consuming and expensive. The second challenge arises because it has to deal with user-generated texts.

2.2 Text Representation Techniques

In the field of NLP, text representation techniques play a crucial role in transforming raw textual data into a format that can be effectively processed by ML algorithms. Two widely used techniques in this regard are the bag-of-words (BoW) model and the Term Frequency-Inverse Document Frequency (TF-IDF) weighting scheme. These techniques provide a foundation for text analysis and have proven to be valuable tools in various NLP tasks, such as document classification, sentiment analysis, and IR.

2.2.1 Bag-of-Words Approach

The scientific article [23] discusses the use of the BoW method in computer vision and text classification. The BoW method counts the occurrences of each word or feature in a document or image, disregarding the order or grammar. It generates a histogram to represent the document or image. The BoW method is simpler than other classification methods and has shown high performance in text and image classification benchmarks.

The authors have found several challenges when using the BoW approach, those include:

- Viewpoint variation: When the same object is captured from different positions, it becomes difficult for the BoW approach to recognize and match the object accurately.
- Illumination: Changes in lighting conditions can affect the appearance of an object in an image, making it challenging for the BoW approach to handle variations in illumination.
- Deformation: Objects that undergo shape deformations, such as bending or stretching, pose difficulties for the BoW approach to correctly identify and classify them.

- Occlusion: When objects are partially or fully obscured by other objects in an image, the BoW approach may struggle to accurately detect and distinguish them.

- Background clutter: If the object of interest has a color similar to or blends with the background color, the BoW approach may face difficulties in differentiating the object from the background clutter.

- Intra-class variation: Objects belonging to the same class or category but having different subtypes or variations (e.g., different types of chairs) can pose challenges for the BoW approach to accurately classify and categorize them.

The researchers of [27] proposed a network BoW model for text analysis. The BoW model is a commonly used text representation method where each word in a document is counted independently, without considering the correlation among words. In contrast, the proposed model takes into account the high-level structural and semantic meaning of words by using a network representation. To evaluate the performance of the proposed model, the researchers applied it to text classification tasks on four different datasets. They compared the performance of the proposed model with seven other text representation methods. The results showed that the proposed model achieved the best performance with high efficiency. The network property called Eccentricity was found to provide the highest accuracy. They also investigated the influence of different network structures in the proposed model. They found that the dynamic network was more suitable for text classification compared to the static network and the hybrid network.

The work proposed in [29] was to incorporate fuzzy theory into the original BoW model to learn dense, robust, and effective representations for documents. The goal is to address the limitations of the BoW model, such as extreme sparsity, high dimensionality, and inability to capture semantics.

The main findings of this work were that the proposed FBoW and FBoWC models can capture more semantic information compared to the original BoW model. The FBoWC model, which uses word clusters instead of individual words as basis terms, is found to be more informative and less redundant than the FBoW model. The performance of the FBoWC mean and FBoWC max variants is more stable and robust across different

datasets. The document also mentions the use of fuzzy systems in text mining and the limitations of deep models in document modeling.

The article [7] work was to evaluate the usefulness of machine translation, specifically Google Translate, for comparative researchers using BoW text models. The researchers compared term-document matrices (TDMs) and topic model results from gold standard translated text and machine-translated text. They evaluated the results at both the document and corpus level.

The technique involved using machine translation, Google Translate, to convert non-English texts into English. Then the translated texts were preprocessed and turned into TDMs. They estimated topic models on these TDMs and compared the similarities of the TDMs, the topical prevalence at the document and corpus level, and the topical content.

The results showed that the TDMs of machine-translated and gold standard documents were highly similar, with minor differences across languages. The topic models also yielded highly similar results, with only small differences across languages.

2.2.2 TF-IDF Algorithm

TF-IDF stands for "term frequency-inverse document frequency". It is a numerical statistic intended to reflect how important a word is to a document in a corpus. The TF-IDF value increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. The TF-IDF algorithm can be used to weight words in text data, which is helpful in IR and text mining.

In NLP, a corpus is a large and structured set of texts. Corpora are used for statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory. Corpora are also used to train language models and other machine-learning models. In the context of the TF-IDF algorithm, the corpus refers to the set of all documents in which we want to compute the tf-idf values for each word.

The basic formula for TF-IDF is:

- TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document) (See Figure 2.1)

$$TF_{(w,j)} = \frac{\text{(Number of times term w appears in a document)}}{\text{(Total number of terms w in the document)}}$$

Figure 2.1: Term Frequency Formula

- IDF(t) = log(Total number of documents / Number of documents with term t in it) (See Figure 2.2)

$$IDF_{(w)} = \log \frac{\text{Total number of documents}}{\text{Number of documents with term w in it}}$$

Figure 2.2: Inverse Document Frequency Formula

- TF-IDF(t) = TF(t) * IDF(t)

The intuition behind this formula is that words that are common in all documents are not very informative and, thus should have a lower weight. Conversely, words that frequently appear in only a few documents are highly informative and should have a higher weight. The TF-IDF algorithm thus assigns a weight to each word in each document, where the weight is the product of the term's TF and IDF values.

The results of the TF-IDF formula can be analyzed in several ways. One common approach is to use the computed TF-IDF values to identify the most important words in a document or a set of documents. This can be done by sorting the words in descending order of TF-IDF value and selecting the top N words. These words are often considered the keywords or keyphrases of the document and can be used for tasks such as IR, text summarization, and document classification.

The article [28] presents the development of a recommendation system for Microsoft news data using TF-IDF and cosine similarity methods. The system suggests relevant news articles to users based on their preferences. The authors collected and preprocessed the data, used TF-IDF to extract features, and calculated cosine similarity scores to recommend articles. They evaluated the system's performance using precision, recall, and F1-score metrics. The results show that the proposed system outperforms similar systems in precision, recall, and F1-score. The authors also discuss the strengths and limitations of their system and suggest areas for future research.

The paper [3] proposes an automated question answering system that uses an improved TF-IDF and cosine similarity to provide precise and relevant answers to users' queries with

high confidence. The study compares different techniques for automatic question answering and finds that rule-based techniques are less effective due to natural language's lack of fixed patterns. The proposed approach pre-processes all repository questions and generates a matrix using the improved TF-IDF model to find the similarity of each user query. The system removes stop-words and applies lemmatization and POS tagging techniques for pre-processing. The empirical analysis-based results show that the proposed technique takes less than five seconds to respond to user queries with maximum similarity and attains up to 84 per cent accuracy.

In the scientific article [30], the authors propose a refined TF-IDF algorithm called TA TF-IDF to find hot terms based on time distribution information and user attention. They also introduce a method to generate new terms and combined terms using Chinese word segmentation. The authors then extract hot news based on the hot terms and group them into K-means clusters to detect hot topics. They propose using the AdjustTime value, AdjustAttention value, and AdjustIDF value to refine the TF-IDF algorithm and improve the accuracy of hot term identification. By combining these factors, their system effectively filters non-hot news and accurately identifies hot topics.

With the rapid evolution of the internet, more users are relying on internet services for their daily work, including booking hotels for travel. In order to provide users with more options for hotel recommendations, the study [18] explores the use of TF-IDF to determine the weight value of terms or documents frequency, and cosine similarity to extract similar values from sentiment datasets.

The collection and analysis of online reviews are major challenges in opinion mining. However, these data can provide useful information to help tourists make informed decisions and improve tourist-related services. The study proposes the use of TF-IDF and cosine similarity to find similar hotels based on user reviews and make recommendations to users.

The study demonstrates the potential of these techniques in providing personalized and valuable hotel recommendations for tourists. The findings suggest further research to enhance recommendation systems in the tourism industry.

2.3 Sentiment Analysis

Sentiment analysis is the process of automatically identifying whether a user-generated text expresses positive, negative or neutral opinion about an entity. Social media platforms caused an increase in the amount of user-generated data. Due to this growth organizations saw an opportunity in keeping track on customers reviews and opinions about their problems, but this task proved to be very challenging. Tweets can be used as a valuable source to extract public's opinion. The objective of [11] was to give a step-by-step detail about the process of sentiment analysis on Twitter data using ML, used Naive Bayes and Decision Tree Algorithms.

Emotions expressed in social media messages can greatly influence the spread of misinformation and online radicalization. It is important to accurately identify these emotions to make deductions from social media messages. [16] proposed to evaluate the performance of three publicly available word-emotion lexicons (NRC, Depeche Mood and EmoSenticNet) on a set of Facebook and Twitter messages. They designed and implemented an algorithm that applies NLP techniques through a number of heuristics that reflect the way humans naturally assess emotions in written texts. This work found notable differences in the performance of the lexicons and also took into account the emotion scores provided by human raters in a survey.

Twitter is a platform for people to share and express their views about, topics, happenings, products and other services. Tweets can be classified into different classes based on their relevance with the topic searched. To classify them there are some ML Algorithms (Naive Bayes, Baseline, Support Vector Machine, etc.) currently employed, these algorithms classify the tweets into positive and negative classes based on their sentiment. The proposed solution in [10] is an implementation of Naive Bayes using sentiment 140 training data using Twitter Database. Naive Bayes is used with SentiWordNet to improve accuracy of classification of tweets. To implement the proposed solution there are five steps:

1. Selection of training data – selected having in consideration the basis of the type of problem;

2. Preprocessing of training data – removing irrelevant information like URLs, user-

names, slang words, symbols, etc.;

3. Establish connection with Twitter database using Twitter API – recent tweets can be extracted for analysis purpose;

4. Various ML algorithms used for classification of tweets into different classes;

5. Results are displayed on the basis of polarities of tweets after their classification.

Results, with a training data set with 1.6 million tweets of sentiment 140 data set, after training system and test with small testing data set of 100 tweets, the system gives an efficiency of 58.40 per cent. This means that analysing human sentiments is not an easy job for machines, and that they need more exhausting training in order to have a better accuracy.

Published in 2021, [20] presents a study on the use of the TextBlob library for sentiment analysis. The study aims to show how sentiment analysis can be used to make decisions and how the TextBlob library can be used to perform sentiment analysis.

Tweets related to a specific topic were collected, and used the TextBlob library to perform sentiment analysis on the tweets. The study also used data visualization techniques to represent the results of the sentiment analysis.

The results of the study showed that the TextBlob library can be used to perform sentiment analysis with good accuracy and that sentiment analysis can be used to make decisions by analyzing the opinion of people on a specific topic. It also provides an understanding of how data visualization techniques can be used to represent the results of sentiment analysis.

The scientific article [1] focuses on using sentiment analysis to analyze emotional responses to the COVID-19 pandemic in Nigeria. The study collected 1,048,575 tweets using the hashtag 'COVID-19' from Twitter. The tweets were pre-processed and analyzed using TextBlob and VADER analyser for sentiment analysis.

TextBlob, a Python text processing package, calculates two key properties for each input sentence:

- Polarity: This property reveals the polarity of emotions conveyed in the sentence, ranging from -1 to 1. Negative sentiment corresponds to a value of -1, while positive sentiment corresponds to +1.

- Subjectivity: This property quantifies the degree of subjectivity in the sentence, indicating the speaker's personal emotions, beliefs, and opinions. The subjectivity score ranges from 0 to 1, with values closer to 0 indicating objectivity and reliance on factual information.

On the other hand, VADER (Valence-Aware Dictionary and Sentiment Reasoner) employs a lexicon-based approach for sentiment analysis. It examines lexical features, such as words, and categorizes them as positive or negative based on their semantic orientation.

VADER provides sentiment scores including:

- Positive, Negative, and Neutral Sentiment Probabilities: These scores represent the likelihood of a statement being positive, negative, or neutral, respectively.

- Compound Score: This aggregated sentiment score combines positive and negative scores, providing an overall sentiment intensity. The compound score ranges from -1 to 1.

The study's results showed that VADER sentiment analysis returned 39.8 per cent positive, 31.3 per cent neutral, and 28.9 per cent negative sentiments. In comparison, TextBlob analysis returned 46.0 per cent neutral, 36.7 per cent positive, and 17.3 per cent negative sentiments.

The researchers concluded that social media data, especially from Twitter, can play a crucial role in aiding organizations and governments to make informed decisions during the COVID-19 pandemic. By understanding public sentiment, authorities can address concerns and combat misinformation effectively.

The article [21], presents a study on sentiment analysis of comments on the professional social networking platform, LinkedIn. The aim was to examine the sentiment of comments on LinkedIn posts and identify the most frequent words used in the comments. There were collected comments on LinkedIn posts and used sentiment analysis techniques to classify the comments as positive, negative, or neutral. Text mining techniques were also implemented to identify the most frequent words used in the comments.

The results of the study showed that a majority of the comments were neutral, but there were also a significant number of positive and negative comments. The study also

found that the most frequent words used in the comments were related to the topic of the post and the industry of the users.

2.4 NLP and Sentiment Analysis applied to Talent Acquisition

This section will examine the various NLP techniques used for talent acquisition and sentiment analysis, the evaluation metrics used, and the results obtained. Additionally, it will explore the potential of NLP to improve the efficiency and effectiveness of hiring processes.

The article [19], presents a case study on the use of NLP to enhance recruitment processes. It aims to show how NLP can improve the efficiency and effectiveness of recruitment processes, particularly in the context of an internship campaign.

The study collected resumes and cover letters of candidates applying for an internship campaign, and used NLP techniques to extract relevant information and perform a semantic analysis of the resumes and cover letters. The study also used ML algorithms to classify the candidates according to their qualifications and skills. The results of the study showed that the use of NLP in the recruitment process improved the efficiency and effectiveness of the internship campaign by reducing the time and resources needed to review resumes and cover letters and by identifying the most qualified candidates.

The article [26], presents a study on the design of an internship recruitment platform that uses NLP based technologies. The authors proposed a recruitment platform that uses NLP techniques to extract relevant information from resumes and cover letters, and ML algorithms. The platform also includes a semantic search engine that allows employers to search for candidates based on specific qualifications and skills. It was presented a prototype of the platform and discussed the potential benefits of using NLP in recruitment processes, such as reducing the time and resources needed to review resumes and cover letters and identifying the most qualified candidates. The prototype could be used as a starting point for further research or development.

The scientific article [15] discusses the increasing importance of recruiting and talent acquisition within the field of Human Resource Management (HRM), particularly for

high-tech positions. With the rise of specialized companies in the knowledge economy, the selection and retention of skilled employees have become critical for organizational success. This has led to the development of various software tools and technologies, including artificial intelligence (AI), to improve efficiency in HR operations and recruitment processes.

The article categorizes the software tools used by recruiters and talent acquisition professionals into three categories: job aggregator software, candidate assessment software, and applicant tracking software (ATS) tools. Job aggregator software collects and organizes job postings from various websites, effectively attracting potential applicants. Candidate assessment software evaluates individuals based on preset criteria and measures, assisting in the selection process. ATS tools help recruiters manage the recruitment process, from creating job postings to on boarding selected candidates.

AI-based tools for candidate engagement utilize chatbots to interact with candidates, provide updates, and schedule interviews. AI-based tools for candidate selection combine pattern recognition and analysis with ML to assess candidates' facial expressions, voice, and tone during video interviews. The authors emphasize the need to incorporate human touch and core values alongside technological advancements in order to create an efficient and cost-effective approach to talent acquisition.

[8] explores the use of NLP techniques in data-driven HR to improve the recruiting process. The researchers developed a resume parser using NLP to analyze key recruitment parameters. They also used a pie chart representation for candidates in the algorithmic structure of the parser, making it a powerful tool for resume matching based on job criteria. To evaluate the accuracy of the resume parser, the researchers applied the firefly ranking algorithm. They achieved an overall accuracy of 94.19 per cent with their approach, indicating the robustness and accuracy of the results.

The article discusses the challenges faced in the traditional method of manually skimming and scanning through resumes, highlighting the error-prone nature of this approach. To overcome these challenges, the researchers used a three-level hierarchical structure to parse resumes, consisting of segments, blocks, and chunks. They utilized the structured nature of resumes to classify and extract information, achieving high accuracy in segmentation and identification rates of named entities.

The paper [4] describes an Intelligent Decision Support System (IDSS) for evaluating CVs using natural language processing techniques. The proposed system is designed to automate the process of CV evaluation, which is traditionally time-consuming and requires significant human effort. The system is based on algorithms that analyze the CVs and extract relevant information, such as the candidate's skills, education, experience, and achievements. The system then uses this information to recommend the candidate's suitability for a particular job.

The paper provides a detailed description of the algorithms used in the system, including the preprocessing steps, feature extraction, and classification models. The system is trained and evaluated using a data set of 300 CVs and achieves an accuracy of 94.3 per cent. The findings of the study suggest that the proposed approach is effective in screening and filtering CVs. The tool can provide decision makers with a shortlist of applicants who meet the criteria defined by the organization. This not only speeds up the hiring process but also reduces bias and inconsistency in decision making.

Ethics is an important consideration when applying NLP techniques to recruitment and sentiment analysis. The use of NLP in these areas has the potential to greatly improve efficiency and effectiveness, but it also raises ethical concerns related to privacy, bias, and discrimination.

In the area of recruitment, NLP can be used to analyze resumes and job applications to identify candidates who are a good fit for a position. However, there is a risk that NLP algorithms may perpetuate existing biases or discriminate against certain groups of people, such as those from marginalized backgrounds. It's essential to ensure that the data used to train the algorithm is diverse and that the algorithm is evaluated for any potential bias.

Sentiment analysis, which uses NLP to determine the emotional tone of text, is also subject to ethical concerns. For example, using sentiment analysis to evaluate social media posts for the purpose of making employment decisions can raise privacy concerns. Additionally, sentiment analysis algorithms may be subject to bias, resulting in inaccurate analysis, particularly when analyzing text from underrepresented groups.

Moreover, the use of NLP in these areas also brings up issues around data privacy, ownership, and control. When collecting data on individuals, it's important to obtain

informed consent, to be transparent about how the data will be used, and to take steps to protect the data.

One notable example of ethical concerns related to the use of NLP in recruitment is the case of Amazon's automated hiring tool. The company developed an algorithm to help screen job applicants, but it was discovered that the algorithm had a bias against female candidates because it was trained on resumes submitted to the company over a 10-year period, which were mostly from men. The algorithm also had difficulty assessing resumes that used words such as "women" or "female," which led to it downgrading resumes that included those words. As a result, Amazon had to discontinue using the algorithm in their recruitment process.

The article [14], presents an overview of the ethical issues that arise when using AI systems in hiring and provides a case study of Amazon's AI-based hiring tool. The aim of implementing AI in hiring is to save time and money by efficiently identifying top applicants. Amazon's system used NLP and ML to analyze resumes and rate candidates on a scale of 5 stars. However, the system was found to have a gender bias, downgrading resumes that included terms related to women or graduates from all-women's colleges.

The article highlights the challenges in adopting AI technologies, particularly related to technological, privacy, and ethical concerns. Privacy issues arise when training AI systems with personal data of applicants, raising questions about data consent. Ethical dilemmas include ensuring fairness, managing conflicting ideas, maintaining diversity, and preserving contextual integrity. The article discusses the different definitions of fairness in computer science and the difficulties in optimizing AI systems to meet these various definitions.

The company and engineers were responsible for designing and implementing the system, but they did not consider moral and ethical requirements or thoroughly test the system. Managers relied on the system without questioning its biases. Legal implications are also discussed, including violations of employment equality acts and personal data protection regulations.

The study emphasizes the importance of considering ethical, privacy, and legal implications when using AI systems in hiring. While AI can improve efficiency, it must be implemented carefully to avoid biases and the potential for discrimination.

The article [13], presents a systematic review of the literature on discrimination and fairness by algorithmic decision-making in the context of HR recruitment and development. The study aims to provide an overview of the current state of research on the use of algorithms in HR and to identify the main challenges and solutions related to discrimination and fairness.

The study found that the use of algorithms in HR can lead to a number of ethical issues such as bias, discrimination, and lack of transparency. The study also found that there are challenges in addressing these issues, such as the lack of data privacy and security, and the lack of interpret ability of the algorithms.

The study also reviewed solutions proposed in the literature such as the use of explainable AI, bias detection and correction algorithms, and the use of diverse data sets to train the algorithms.

This study provides a comprehensive overview of the current state of research on the use of algorithms in HR and the challenges and solutions related to discrimination and fairness, the authors of this study also provide recommendations for organizations and future research on the topic. Additionally, it highlights the importance of ensuring fairness and transparency in the use of algorithms in HR processes such as recruitment and development and the importance of addressing ethical issues related to discrimination and bias.

Another example is in case of a Japanese company that developed a recruitment system that uses facial recognition technology to evaluate job applicants. The company claimed that the technology can assess candidates' personalities and emotions, but it raised concerns about privacy and the potential for bias.

These cases highlight the importance of regularly evaluating NLP algorithms for potential bias and engaging in ongoing dialogue with stakeholders to ensure that the use of these techniques is fair and ethical. It's also essential for companies, organizations and Researchers to provide transparency about their data collection and decision-making process and are in compliance with laws and regulations.

[6] is a research paper that provided an overview of the types of bias that can occur in NLP systems and discusses strategies for detecting and mitigating bias in NLP systems.

The paper starts by defining bias in NLP and identifying the different sources of bias

in NLP systems, such as biased data, biased algorithms, and biased evaluation metrics. It also examines how bias in NLP can affect the performance and the decision-making process of the system.

The paper then provides an overview of the current state of research on bias in NLP, including a summary of the methods that have been proposed to detect and mitigate bias in NLP systems. These methods include techniques for identifying and removing bias from the data, such as data preprocessing and data augmentation, methods for detecting bias in algorithms, such as fairness metrics and bias detection methods, and methods for mitigating bias in the algorithm, such as debiasing methods and adversarial training.

The paper also highlights the importance of considering different types of bias and their potential impacts, and it suggests ways to identify the potential sources of bias in the data and algorithms.

The authors conclude by pointing out that bias in NLP is a complex problem that requires a multi-faceted approach, involving the examination of data, algorithms, and evaluation methods. They also stress the importance of considering the downstream impact of the bias and the potential harms for certain groups and to engage in ongoing dialogue with stakeholders to ensure that the use of these techniques is fair and ethical.

In conclusion, NLP techniques have the potential to greatly benefit recruitment and sentiment analysis, but it's crucial to be mindful of the ethical concerns and take steps to mitigate any potential negative impacts.

Chapter 3

Methodology

This chapter presents the approach that will be taken to carry out the project.The methodologies that will be used for the research, development, and implementation of the project are presented. Specifically, the chapter will describe how the Research and Development (RD) process, and the Engineering Design Process (EDP) will be employed to ensure that the project is completed in a systematic, efficient and effective manner.

The RD process is a systematic approach that encompasses various stages, each contributing to the development of new goods or services. It begins with the identification of potential areas for improvement or innovation within a particular industry or organization. This initial step involves exploring new ideas, concepts, and emerging technologies.

Basic research is focused on advancing knowledge and understanding of a particular field or topic. It is typically conducted in universities or research institutions and is often driven by curiosity or a desire to gain new insights.

Applied research, on the other hand, takes the knowledge and understanding gained from basic research and uses it to develop new products, processes or services. This is typically done in industry and government organizations, it's goal is to solve specific problems and improve the performance of existing products or create new ones.

Even though the RD process can be time-consuming and expensive, it can also result in important advancements and discoveries that can yield high returns on investment. It is one of the main forces behind innovation, which is essential in any setting where businesses compete.

The EDP is a part of the process of RD's applied research. In several disciplines, such

as engineering, product design, and software development, EDP is an organized method for resolving issues and developing new products or systems. It is frequently employed as a method of putting the findings of the research phase into practice by taking the knowledge and understanding acquired via fundamental research and converting it into a useful and practical design.

In RD process, the initial phase focus on gathering information and data relevant to the problem at hand, and identify the objectives and constraints of the project. The next step is generating solutions, this include the process of creativity and brainstorming where different solutions are proposed.

Then, by using the EDP, the project team can systematically and methodically evaluate the feasibility, cost, and performance of each solution, and select the best one. The EDP then helps to guide the development of detailed specifications for the chosen solution, including drawings and models. Finally, the EDP is used to build and test a prototype, implement and evaluate the solution in real-world setting and refine and improve it based on the feedback and evaluation.

This process is iterative, it may go back to certain steps of the process if needed to improve or revise the solution. Each of these steps helps to ensure that the final solution is well thought-out, feasible, and meets the needs of the project.

The first step of the EDP, defining the problem, can help to recognize the specific difficulties associated with selection within the context of urban platforms, including the requirement for speed and effectiveness in the selection process, as well as the desire to utilize the creative ideas and abilities provided by start-ups and entrepreneurs.

During the research phase, data will be gathered regarding the needs of urban platforms, their current state, and existing methods to attract talent.

The generate solutions step will include the process of brainstorming, which will allow to come up with different solutions to the problem of finding suitable partners for projects, including the use of AI and NLP techniques for content matching.

Evaluating solutions step, to assess the feasibility and performance of each solution and select the best one. This step will be important to ensure that the chosen solution is well thought-out, feasible, and meets the needs of the project.

Next step is developing a design, this step requires creating detailed specifications for

the chosen solution, including drawings and models.

Building and testing, a prototype of the solution will be developed and tested to determine its performance and identify any issues that need to be addressed.

Implementing and evaluating step, implement the solution in real-world setting and evaluate its performance.

Finally, refining and improving, based on the feedback and evaluation, refine and improve the solution.

By using EDP, it will be possible to systematically and methodically develop a solution that uses AI and NLP techniques to develop an efficient and effective content matching tool.

Chapter 4

Experimental Setup

As mentioned previously, organizations are constantly evolving into increasingly complex environments due to global competition, creating a need for partners and service providers who can meet their needs and help them improve the quality of their services. In this scenario, the content matching tool aims to help establish partnerships of strategic interest to respond to the innovation needs of these organizations.

The following sections will describe the steps to create the content matching tool.

4.1 Construction of the Data Set

The development of the tool requires a significant amount of data that is relevant and specific to the task, such as information about the needs and ideas of the business sector, as well as information about the skills and offers of start-ups and entrepreneurs.

Creating a specific dataset for the project would involve several steps, such as:

- Data collection: gathering relevant data from various sources, such as news articles, social media posts, and websites, that pertain to the needs and ideas of the business sector, as well as the skills and offers of start-ups and entrepreneurs.
- Data cleaning: removing any irrelevant or redundant information from the data, such as special characters, punctuation, digits, and stopwords, as well as converting text to lowercase and removing duplicate data.
- Data annotation: labeling the data with relevant categories or tags, such as the type

of need or idea, or the type of skill or offer, so that it can be used for training and evaluating the NLP models.

- Data preprocessing: applying the different text preprocessing techniques, such as tokenization, stemming, lemmatization, stopword removal, Named-entity recognition, part-of-speech tagging, so that the data is ready for the matching process.

Creating a specific dataset for the project required a significant amount of time and resources, but it is essential for the success of the project. A good dataset would ensure that the matching process is accurate and effective, which is the ultimate goal of the tool.

4.1.1 Using AI models to generate data

The expansion of the dataset was deemed necessary to facilitate the development of the tool. This imperative arose from the need for a substantial volume of pertinent and task-specific data. This encompassed comprehensive information pertaining to the requirements and conceptualizations within the business sector. Additionally, there was a requisite for detailed insights into the skill sets and propositions put forth by start-ups and entrepreneurs.

To fulfill this need, a larger dataset was curated, leveraging data generated through interactions with the ChatGPT model. This augmentation was instrumental in enhancing the diversity and richness of the dataset, ensuring a more robust foundation for the tool's development.

Projects Dataset has the following collumns:

- Project: A unique identifier for each project.
- Name: The name or title of the smart cities project.
- Description: A brief description of the project, outlining its goals and objectives.
- Technologies: Technologies associated with the project, providing insights into the tools and innovations involved.
- Ideal Partners: The types of partners or collaborators deemed ideal for the successful implementation of the project.

Candidates Dataset is organized as follows:

- Candidate Name: The name of the candidate company or startup.
- Industry: The industry or sector to which the candidate company belongs.
- Candidate Description: A description of the candidate company, providing information about its expertise, focus areas, and offerings.

These projects and candidates were hypothetical and created for testing purposes. The candidates were created with diverse industry backgrounds, company names, and company descriptions to simulate realistic variations.

The projects dataset comprises 100 distinct projects (Figure 4.1), while the candidate dataset includes a total of 300 candidates (Figure 4.2).

Project	Name	Descriptio	Technolog	Ideal Partners
1	Integrated	Implement	IoT sensor	Waste management companies with expertise in optimizing collection routes, Environmental NGOs focusing
2	Sustainabl	Developin	Smart met	Energy utility companies with experience in grid management, Renewable energy providers contributing to cl
3	Smart Wat	Implement	IoT sensor	Water utility companies with expertise in water infrastructure, Environmental NGOs focusing on water conse
4	Urban Air	Creating a	Air quality	Environmental agencies specializing in air quality management, Health organizations concerned with public w
5	Smart Park	Developin	IoT sensor	Local government agencies managing urban planning, Parking management companies with expertise in smar
6	Public Safe	Implement	Security ca	Local law enforcement agencies with a focus on public safety, Public safety organizations involved in emerge
7	Smart Buil	Integratin	Building au	Real estate developers committed to sustainable building practices, Energy efficiency companies specializing
8	Communit	Establishin	Wireless ir	Telecommunication companies providing internet services, Local government supporting community initiativ
9	Smart Stre	Implement	IoT sensor	Local government agencies focused on urban infrastructure, Energy efficiency companies specializing in smar
10	Green Urb	Creating a	Electric ve	Transportation authorities with experience in sustainable mobility, Sustainable mobility companies promotin

Figure 4.1: Projects Dataset

Company Name	Industry	Company Description
EnviroTech Solutions	Environme	EnviroTech Solutions is a leading provider of innovative environmental technologies. Our expertise includes the development of IoT sensors, machine learning algorithms, and data analytics platforms for optimized waste col
RenewableGrid Syster	Energy Tec	RenewableGrid Systems specializes in the development of smart grid solutions to optimize energy distribution. Our focus is on smart meters, IoT devices, machine learning algorithms, and energy storage solutions for promot
AquaSense Technolog	Water Ma	AquaSense Technologies is dedicated to implementing intelligent water management systems. We excel in IoT sensors, data analytics platforms, and machine learning algorithms to monitor water usage, detect leaks, and en
AirGuard Innovations	Air Quality	AirGuard Innovations is a pioneer in creating a network of sensors for real-time air quality monitoring. Our expertise lies in air quality sensors, IoT devices, and data analytics platforms to improve urban air quality.
ParkSmart Solutions	Smart Park	ParkSmart Solutions is at the forefront of developing smart parking solutions. Our focus includes IoT sensors, mobile apps, and data analytics platforms to optimize parking space utilization and reduce traffic congestion.
SafeCity Surveillance	Public Safe	SafeCity Surveillance specializes in comprehensive surveillance systems using AI and analytics for improved public safety. Our solutions include security cameras, video analytics, and machine learning algorithms.
EcoBuild Energy Mana	Smart Buil	EcoBuild Energy Management integrates IoT devices and sensors to optimize energy usage in commercial and residential buildings. Our expertise includes building automation systems, IoT devices, and energy management s
CommunityConnect V	Wireless C	CommunityConnect Wi-Fi is dedicated to establishing community-wide Wi-Fi networks for enhanced digital connectivity. Our solutions involve wireless infrastructure, internet service providers, and collaboration with comm
GreenLight Street Ligh	Smart Stre	GreenLight Street Lighting focuses on implementing energy-efficient and sensor-controlled street lighting. Our solutions include IoT sensors, LED lighting, and centralized control systems for improved safety and reduced ener
UrbanMobility Solutic	Urban Mo	UrbanMobility Solutions is committed to creating a sustainable and integrated urban mobility system. Our expertise includes electric vehicles, smart traffic management, and public transportation systems for efficient city co
HealthTech Innovatio	Health Tec	HealthTech Innovations focuses on deploying wearable devices and IoT sensors to monitor community health and provide early intervention. Our solutions include wearable devices, health monitoring apps, and data analytic
CivicEngage Platform	Civic Techi	CivicEngage Platform is dedicated to developing a digital platform to facilitate citizen engagement, feedback, and participation in urban planning. Our solutions involve mobile apps, web platforms, and community engageme
EduTech Solutions	Education	EduTech Solutions specializes in implementing technology in educational institutions for smart classrooms, remote learning, and interactive education. Our expertise includes interactive displays, online learning platforms, an
AgriInnovate Technol	Agricultura	AgriInnovate Technologies is focused on integrating IoT devices and sensors in agriculture for precision farming and improved crop yield. Our solutions include IoT sensors, agricultural drones, and data analytics platforms.
BioDiversity Guardian	Environme	BioDiversity Guardians is dedicated to creating initiatives and technologies to protect and promote urban biodiversity. Our expertise includes green spaces, wildlife monitoring systems, and conservation programs.
TourismTech Solution	Tourism Te	TourismTech Solutions is committed to developing a smart tourism infrastructure to enhance the visitor experience and promote sustainable tourism. Our solutions involve tourism apps, visitor analytics, and smart signage.
ResilientCommunities	Communit	ResilientCommunities focuses on implementing measures and technologies to enhance community resilience against natural disasters and emergencies. Our solutions include disaster monitoring systems, emergency respons
RetailTech Innovators	Retail Tec	RetailTech Innovators specialize in integrating technology in retail areas for enhanced customer experiences, inventory management, and efficient operations. Our expertise includes RFID technology, smart shelves, and cust
GreenRoof Technolog	Green Buil	GreenRoof Technologies promotes the installation of green roofs on buildings for energy efficiency, biodiversity, and improved air quality. Our solutions involve green roof technology, eco-friendly construction, and urban pl
DigitalInclusion Initiat	Digital Incl	DigitalInclusion Initiatives is dedicated to implementing initiatives to bridge the digital divide and ensure access to digital technologies for all community members. Our solutions include digital literacy programs, community c
SmartWaste Network	Waste Ma	SmartWaste Networks aims to expand the use of IoT-enabled waste bins to optimize waste collection routes and promote recycling. Our solutions include IoT sensors, smart waste bins, and data analytics platforms.
SolarCommunity Pow	Renewable	SolarCommunity Power is dedicated to implementing community-based solar power projects to promote renewable energy use at the local level. Our expertise includes solar panels, energy storage solutions, and community
CleanWater Tech	Water Tec	CleanWater Tech focuses on deploying technology to ensure clean water access in urban areas through water purification and distribution systems. Our solutions involve water purification technology, IoT sensors, and data
UrbanGreen Monitori	Green Spa	UrbanGreen Monitoring utilizes technology to monitor and manage urban green spaces for biodiversity conservation and community well-being. Our solutions include green space monitoring systems, IoT sensors, and comm
NoiseGuard Systems	Noise Mar	NoiseGuard Systems is dedicated to implementing a system to monitor and manage urban noise levels for an improved quality of life. Our solutions include noise monitoring sensors, data analytics platforms, and noise reduc
CommuteHub Solutio	Transporta	CommuteHub Solutions is focused on developing integrated transportation hubs with digital services, connectivity, and sustainable commuting options. Our expertise includes digital kiosks, e-mobility solutions, and commute

Figure 4.2: Candidates Dataset

4.2 Development of the solution - Content Matching Tool for City Improvement

This section presents a comprehensive overview of the solution, including text preprocessing techniques, text representation using TF-IDF, GUI development, and the potential inclusion of sentiment analysis. These steps are fundamental to develop an efficient and user-friendly Content Matching tool for bussiness.

4.2.1 Text Preprocessing

The goal of preprocessing is to prepare the text data for analysis by cleaning, normalizing, and structuring it.

There were studied several approaches to perform text preprocessing. The choosen techniques were tokenization, stopwords removal, lemmatization and lowercasing, using the NLTK library. (See Figure 4.3)

```
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
from nltk.corpus import wordnet
```

Figure 4.3: NLTK Library - Text Preprocessing

The first step in preprocessing is tokenization, which involves breaking down the text into individual words/punctuation, known as tokens. This is done using the "word tokenize" function. For example, the sentence "Waste management companies with expertise in optimizing collection routes, Environmental NGOs focusing on sustainable waste practices, Technology companies specializing in IoT and data analytics." is going to be divided in 26 tokens, some of them are in Figure 4.4.

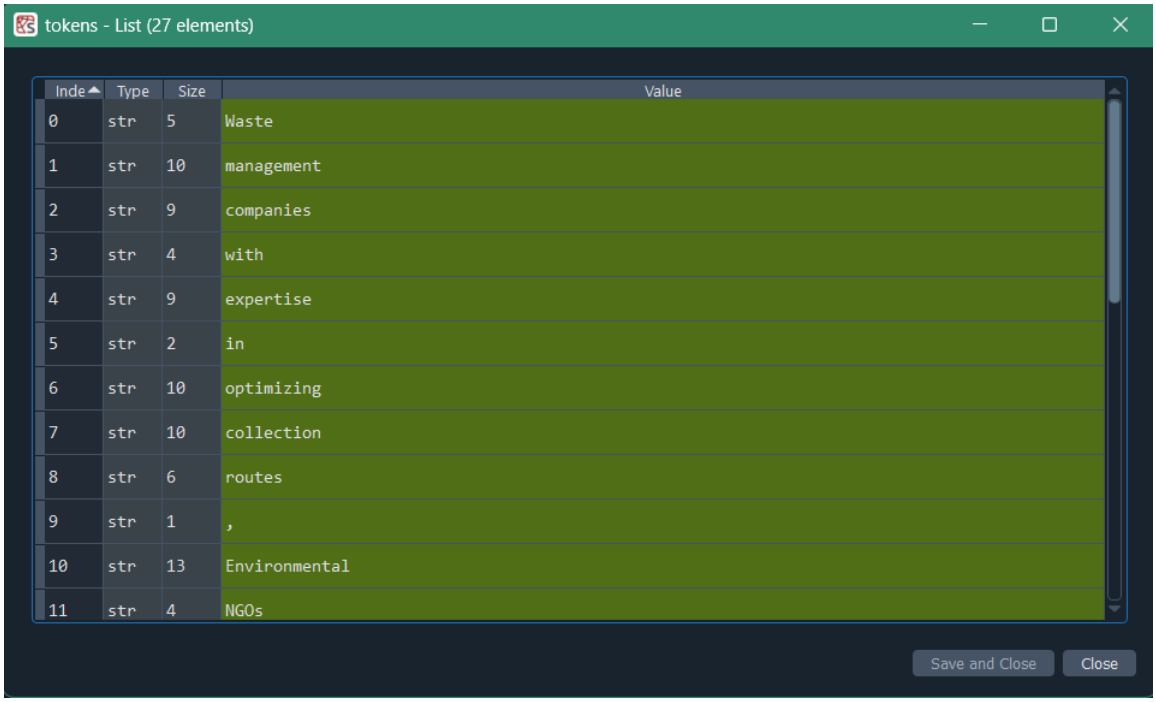

Figure 4.4: Text after performing tokenization

Next, the text is cleaned by removing stop words, commonly used words that do not provide much meaning, such as "and" and "the", using the nltk.corpus.stopwords module.

Another important preprocessing step is lemmatization, which is the process of reducing words to their base or root form, known as lemma. This is done using the nltk.stem.WordNetLemmatizer() function and it helps to reduce the dimensionality of the data and improve the efficiency of the algorithm. Additionally, lowercasing is performed and punctuation is removed from the text to keep consistency and improve the efficiency of the algorithm.

Figure 4.5 has the implementation of the method that was described.

```
def preprocess(doc):
    stopset = set(stopwords.words('english'))
    lemmatizer = WordNetLemmatizer()
    tokens = nltk.word_tokenize(doc)
    lemmas = [lemmatizer.lemmatize(token.lower()) for token in tokens if token.lower() not in stopset and len(token) > 2]
    return lemmas
```

Figure 4.5: Text Preprocessing Method

4.2.2 Text Representation using TF-IDF - English Languge

TF-IDF is a statistical measure that is used to evaluate how important a word is to a document in a collection of documents. It is a product of two statistics, term frequency

(TF) and inverse document frequency (IDF). The TF component measures the number of times a word appears in the document, while the IDF component measures the rarity of the word across the entire collection of documents. A document refers to a unit of text that is treated as a single entity for analysis.

The resulting TF-IDF score for a word in a document represents the importance of that word to the document, with higher scores indicating that the word is more important to the document. By creating a numerical representation of the text using TF-IDF, words that are important to the document are given more weight and words that are not important are given less weight.

Cosine Similarity as an Evaluation Metric

The Cosine Similarity is a measure of similarity between two non-zero vectors of an inner product space. It is calculated by taking the dot product of the vectors and dividing it by the product of their magnitudes. In the context of text analysis, the TF-IDF vectors of two documents are used to calculate the cosine similarity between the documents. The cosine similarity score ranges from -1 to 1, with 1 indicating that the documents are identical and -1 indicating that the documents are completely dissimilar. A cosine similarity score of -1 in text analysis implies a strong contrast or opposition in the content of the two documents. This could occur when the terms and features present in one document are, to a large extent, the opposite of those in the other document.

In this way, the code uses the TF-IDF vector representation of the text and Cosine Similarity as a metric to determine the similarity between two inputted documents. The results are presented in a numerical format, which allows for easy comparison of the similarity between the two documents, as shown in Figure 4.6.

Then, it calculates the TF-IDF vectors for each document, which represent the relative importance of each word in the document (See Figures 4.7 and 4.8).

First, it checks if the similarity value is greater than 0. If it is, then the documents have some overlap in content. It then checks if the similarity value is equal to 1, in which case the documents are identical. If so, it prints a message indicating this. If the similarity value is not equal to 1, the code then checks if the similarity value is greater than 0.5. If so, it prints a message indicating that the documents are quite similar. Otherwise, it prints

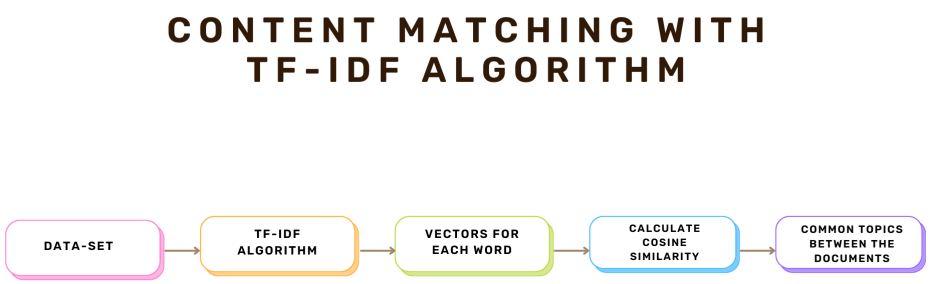

Figure 4.6: TF-IDF with Cosine Similarity to compare documents

```
from sklearn.feature_extraction.text import TfidfVectorizer
import pandas as pd
from scipy.spatial.distance import cosine
```

Figure 4.7: Libraries used for Text Representation

```
# Calculate TF-IDF vectors for each document
vectorizer = TfidfVectorizer()
vectors = vectorizer.fit_transform([documentA, documentB])
feature_names = vectorizer.get_feature_names_out()
dense = vectors.todense()
tfidf_vectors = dense.tolist()
```

Figure 4.8: TF-IDF vector representation in Python

a message indicating that the documents are somewhat similar. If the similarity value is equal to 0, then the documents have no overlap in content. Finally, if the similarity value is negative, it means that the documents have opposite content. Figure 4.9 presents how this algorithms works.

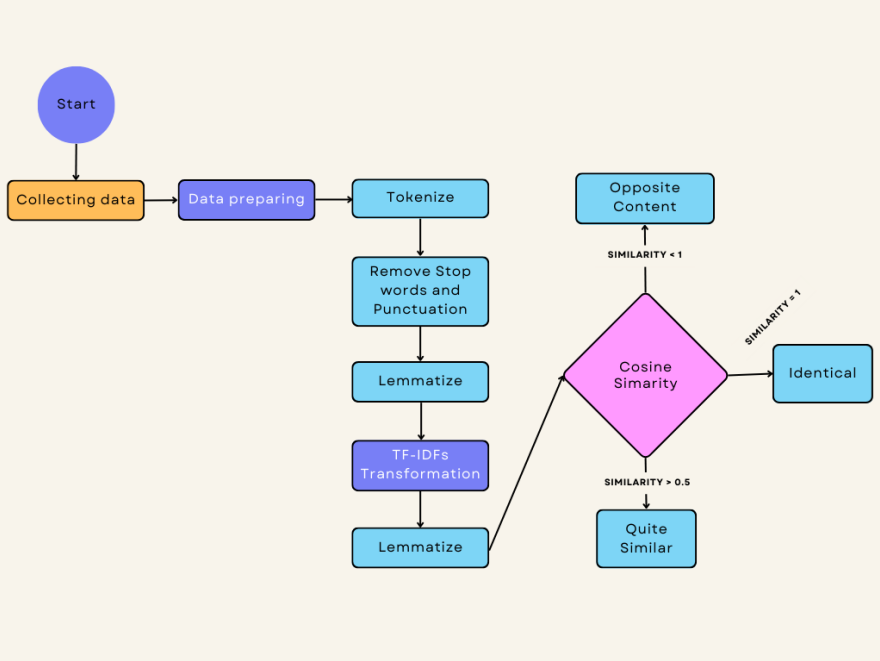

Figure 4.9: Algorithm for the Content Matching Tool

The code calculates the cosine similarity between the two vectors to determine the degree of overlap in content between the two documents. It prints a message indicating the similarity between the two documents based on the degree of overlap. It also identifies the common lemmas between the two documents and prints out the relevant topics for each common lemma using the WordNet library. (See Figure 4.10)

Finally, the code calculates the term frequency (TF) for each word and merges it with the IDF scores.

4.2.3 Algorithm for Portuguese Language

Text preprocessing plays a crucial role in preparing text data for analysis. For English text, standard techniques such as tokenization, removal of stop words, punctuation, and lemmatization using the WordNetLemmatizer, where sufficient. These steps help to normalize the text and reduce noise, enabling more accurate analysis.

However, as the Portuguese language is more complex than English, there was a need for the tool to incorporate specific preprocessing techniques to handle Portuguese text effectively.

Using the previous code as starting point, a few changes were applied so the tool works

```python
# Calculate cosine similarity between vectors
similarity = 1 - cosine(df1.values[0], df2.values[0])
print("Similarity: {:.3f}".format(similarity))

if similarity > 0:
    print("The documents have some overlap in content.")
    if similarity == 1:
        print("The documents are identical.")
    elif similarity > 0.5:
        print("The documents are quite similar.")
    else:
        print("The documents are somewhat similar.")
elif similarity == 0:
    print("The documents have no overlap in content.")
else:
    print("The documents have opposite content.")

# Get the lemmatized words for each document
lemmas_a = preprocess(documentA)
lemmas_b = preprocess(documentB)

# Find the common lemmas between the two lists
common_lemmas = set(lemmas_a).intersection(set(lemmas_b))
```

Figure 4.10: Cosine Similarity in Python

with Portuguese language.

The language used for stop words has been changed from English to Portuguese.

The method used to tokenize the text is the same, but the "word tokenize" method from nltk in this case it's used for Portuguese text, 4.11.

```python
# Pré-processar os documentos, fazendo tokenização, removendo stop words, pontuações e aplicando lematização
def preprocess(doc):
    stopset = set(stopwords.words('portuguese'))
    lemmatizer = WordNetLemmatizer()
    tokens = nltk.word_tokenize(doc)
    lemmas = [lemmatizer.lemmatize(token.lower()) for token in tokens if token.lower() not in stopset and len(token) > 2]
    return " ".join(lemmas)
```

Figure 4.11: Text Preprocessing for Portuguese Language

This tailored approach to text preprocessing is crucial for optimizing the accuracy and relevance of the content matching tool when working with Portuguese textual data.

The subsequent steps in the algorithm, including TF-IDF representation and cosine similarity calculation, remain unchanged, providing a robust and language-specific solution for matching content in Portuguese documents.

This adaptation enhances the tool's capability to effectively analyze and match content in both English and Portuguese, extending its utility and relevance in multilingual contexts.

The final implementation of the content matching tool demonstrates its versatility in accommodating different languages and its potential to serve as a valuable resource for businesses operating in diverse linguistic environments.

4.2.4 Graphic User Interface

The development of a graphical user interface (GUI) that can be used by HR teams to easily navigate and interact with the tool. To make the tool more accessible and user-friendly, allowing recruiters to quickly and efficiently identify the most promising candidates for a given position.

Flask is a lightweight web framework for Python, known for its simplicity and minimalistic design. It is considered a microframework because it does not impose any specific tools or libraries on developers. Instead, Flask is built on top of the Werkzeug WSGI toolkit and uses the Jinja2 template engine.

One of the main advantages of Flask is its straightforward and easy-to-use API, which allows developers to quickly build web applications. The framework prioritizes simplicity and adheres to the "Keep It Simple, Stupid" (KISS) principle. By minimizing unnecessary complexity, Flask enables developers to focus on the core functionality of their applications while keeping the codebase clean and maintainable.

Routing is another essential feature provided by Flask. It simplifies the process of handling different URLs within the application. Developers can easily define routes and associate them with specific functions that handle the corresponding requests. This flexibility makes it effortless to create various pages and endpoints for the web application.

A GET request is used to retrieve data from a server. In Flask, handling a GET request is quite simple. It is used the '@app.route' decorator to define a route for a specific URL. The POST method is used to send data to a server to create or update a resource. In Flask, it is possible to define a route that responds to POST requests in a similar way to the GET method. (See Figure 4.12)

The home() function is the handler for the home route. It handles both GET and

```
@app.route('/', methods=['GET', 'POST'])
```

Figure 4.12: Route for the Home Page

POST requests. If the request method is GET, the home template ('index.html' - Figure 4.13) is rendered without a warning message. If the request method is POST, the function retrieves the documents from the request form data.

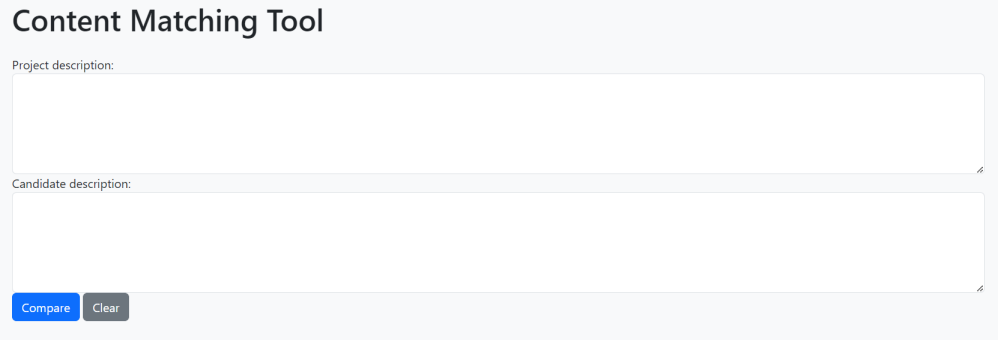

Figure 4.13: index.html - Home Template Rendered

The code checks if both documents are provided. If either document is missing, the home template is rendered with a warning message. (Figure 4.14)

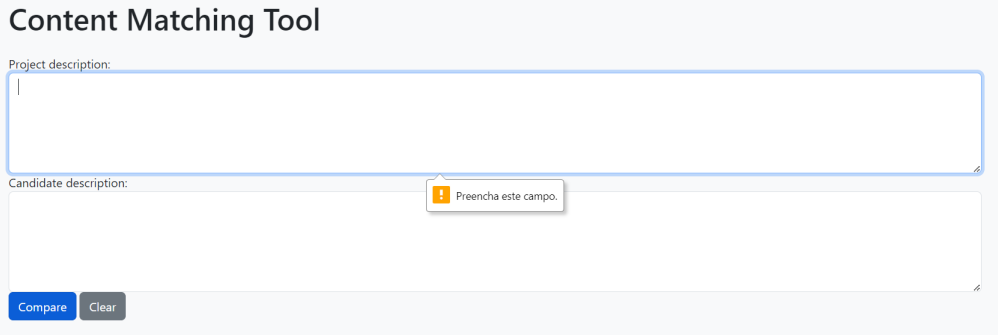

Figure 4.14: Warning Message - Empty Document

The methods used to do the content matching between the documents were described in subsection 4.2.2.

The result template ('result.html') is rendered with the response message and the common lemmas. In section 4.3 it will be analysed how the GUI is working.

4.2.5 Sentiment Analysis

For candidates seeking engagement in smart cities projects, leveraging sentiment analysis becomes crucial in gaining comprehensive project insights. Notably, sentiment analysis, powered by NLP, goes beyond deciphering emotional tones in various textual sources—it serves as a multifaceted tool. Firstly, it allows candidates to assess the compatibility of the project by analyzing sentiments in collaborator reviews, providing a nuanced understanding of the collaborative environment and its alignment with professional objectives.

Secondly, this tool aids in identifying potential challenges within the project. By scrutinizing sentiments in project-related discussions, candidates can uncover issues such as communication gaps or resource constraints, allowing for a proactive assessment of the project's feasibility and potential impact on job satisfaction.

Thirdly, candidates can employ sentiment analysis to evaluate the quality of project deliverables and the project's standing in the professional community. Analyzing comments and discussions related to the project's outputs provides valuable insights into the project's reputation and its ability to deliver high-quality results. This information assists candidates in making informed decisions about contributing to projects with positive public perception and a track record of successful outcomes.

Feeling Extraction using APIs - Twitter and LinkedIn

To perform sentiment analysis on social media data, an API-based script was planned to scrape posts from a specified platform, retrieving content related to a specific topic of interest. This approach woul allow collecting relevant textual data for further analysis and sentiment classification.

In the realm of partnerships, social media platforms can prove to be invaluable for potential future partners. These platforms offer a wealth of information that enables partners to gain insights into the entity that has proposed an offer and assess its alignment with their expectations, beliefs, policies, and values. By examining the entity's online presence, including their posts, engagements, and interactions with their audience, potential partners can gauge their reputation, credibility, and the consistency of their messaging. Social media platforms also provide an avenue for observing public sentiment and reac-

tions towards the entity, allowing future partners to assess how the proposed partnership might be perceived by the wider audience. Furthermore, by analyzing the entity's engagement with their existing partners, stakeholders, and customers on social media, potential partners can gain a deeper understanding of their collaborative approach and the level of satisfaction among their existing network.

One of the key advantages of using the Twitter API for sentiment analysis is the ability to access real-time data. This is particularly useful when studying rapidly evolving topics or events. Researchers and developers can monitor the Twitter stream in real-time, allowing them to capture and analyze the most recent tweets related to their chosen topic of interest. This real-time access is crucial for staying up-to-date with the latest public sentiment and for accurately capturing the dynamic nature of social media discussions.

To perform sentiment analysis, with access to the Twitter API, implies several step as shown in Figure 4.15.

- Set up Credentials: Obtain the necessary credentials (API key, API secret key, Access token, and Access token secret) from the Twitter Developer platform to authenticate and access the Twitter API.

- Access the Twitter API: Utilize a suitable library or programming language (e.g., Tweepy for Python) to connect and interact with the Twitter API.

- Authenticate with the Twitter API: Use the obtained credentials to authenticate the application or script with the Twitter API to enable data retrieval.

- Define Search Query: Specify the search query parameters to define the scope and topic of the tweets to analyze. This includes keywords, hashtags, usernames, language, location, and date range.

- Fetch Query: Execute the search query using the Twitter API to retrieve relevant tweets based on the specified parameters. Retrieve the tweet text, user information, and other relevant data.

- Preprocess the Tweets: Clean and preprocess the retrieved tweets by removing unnecessary elements like URLs, special characters, and stopwords. Perform tasks like

tokenization, stemming, and removing punctuation to prepare the text for sentiment analysis.

- Perform sentiment analysis: Apply a sentiment analysis algorithm or model to the preprocessed tweets to determine the sentiment polarity (positive, negative, or neutral) associated with each tweet.

- Filter and Analyze the Results: Filter the analyzed tweets based on sentiment labels or confidence scores. Perform further analysis, such as aggregating sentiment by user, topic, or timeframe, to gain insights into sentiment trends.

- Present the Results: Visualize and present the sentiment analysis results using suitable charts, graphs, or reports. Communicate the findings, including sentiment distribution, sentiment changes over time, and any significant patterns or trends observed.

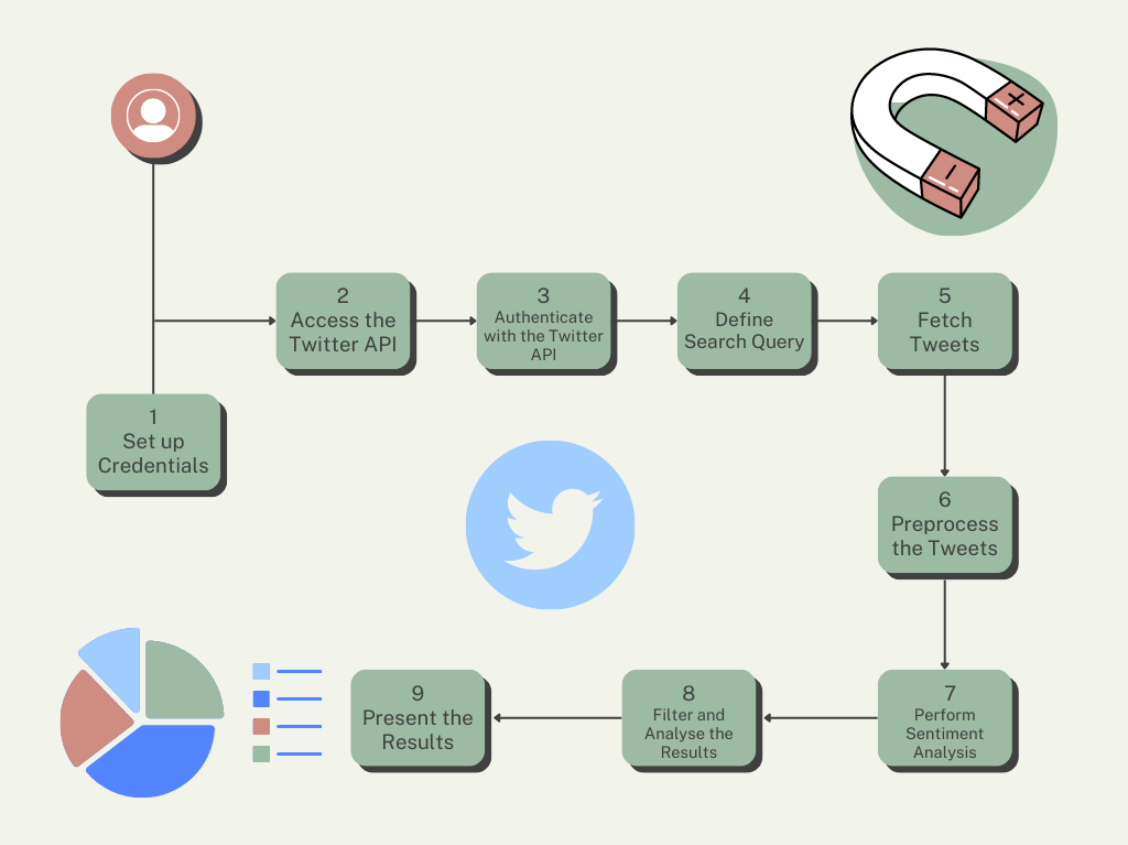

Figure 4.15: Algorithm for Sentiment Analysis with Twitter API

Using LinkedIn to extract sentiment about companies can be more effective than using Twitter for a few reasons:

- Audience: LinkedIn is a professional social media platform where people share their work experiences, skills, and connect with other professionals. Most of the users of LinkedIn are individuals with professional profiles and they share their opinions mainly related to their workplace, job opportunities, and other professional-related topics. In contrast, Twitter is a more general social media platform where people share a wide range of topics.

- Professionalism: LinkedIn is a more professional platform, and as such, users are more likely to express their opinions in a more constructive and formal manner. They may be more willing to give detailed feedback and express their opinions in a way that is less likely to be influenced by emotions or bias. On the other hand, Twitter users might be more inclined to express their opinions in a more casual, informal, and often more emotional way.

- Company focus: LinkedIn is a platform where individuals connect with other people, companies and organizations. Therefore, it is possible for a company to create their own page and collect feedback from their followers and customers. In contrast, on Twitter, it is possible to find feedback about a company, but it is mixed in with a wide range of other topics and comments.

- Quantity vs Quality: LinkedIn has less number of users than Twitter, which makes the amount of data to be analyzed to be lower. However, the data generated on LinkedIn tend to be more relevant, high-quality, and informative when compared to Twitter as LinkedIn users tend to be more engaged with the platform and they usually express their thoughts and opinions in a more professional manner. In contrast, Twitter users are more likely to post in a more casual and informal way, and the volume of tweets can be overwhelming to analyze, making it difficult to extract meaningful insights. This makes LinkedIn a more effective platform for extracting sentiment about a company as it tends to provide a more targeted and relevant data set.

To extract sentiment from comments on LinkedIn, a similar approach can be used as with Twitter. However, in this case, a developer account on LinkedIn is required. It is possible to compare the steps with the previously presented Twitter API in Figure 4.16.

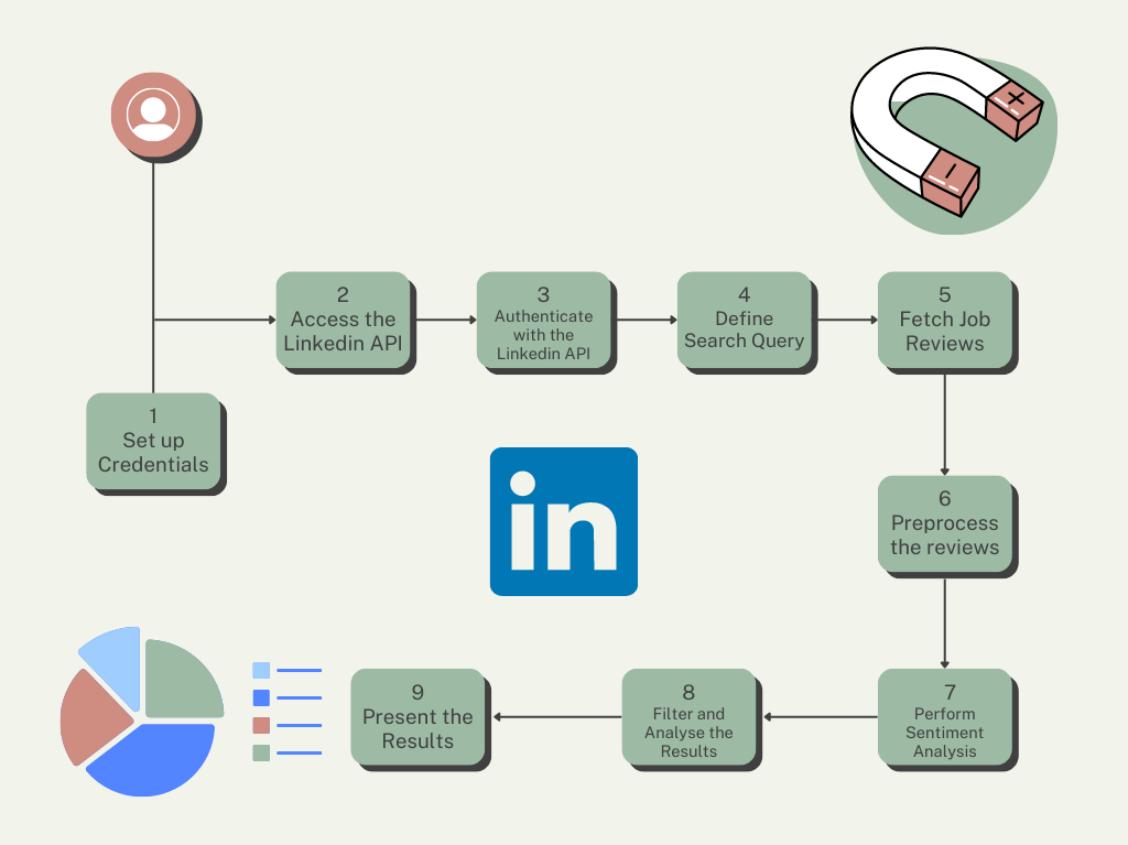

Figure 4.16: Algorithm for Sentiment Analysis with LinkedIn API

Algorithm for Sentiment Analysis

In this initial stage of the project, a pre-existing dataset was chosen over collecting data from Twitter or LinkedIn using their APIs. This decision was made due to the requirement of having an application to use the APIs. Additionally, the free version of the APIs has limitations on the amount and type of data that can be collected.

When using web scraping tools to collect data from websites like Twitter and LinkedIn, it is important to be aware of the legal implications. Web scraping without permission from the website owner is generally considered a violation of their terms of service and may result in legal consequences such as cease and desist letters, account suspension, or even lawsuits. In addition, the data obtained through web scraping may not always be

accurate or up to date, as websites can change their structure or content at any time.

Web scraping refers to the automated process of extracting data from websites. It involves accessing the HTML code of a webpage and extracting the desired information, such as text, images, or links. While web scraping can be a powerful tool for data collection and analysis, it is essential to understand the legal considerations involved.

To avoid legal issues and ensure ethical data collection practices, using a pre-existing dataset was the selected approach. It also saves time and effort that would be required to collect and clean the data. The dataset chosen, Glassdoor Job Reviews [9], contains over 1.5 million reviews of companies, making it a rich source of information. In contrast, collecting data from social media platforms would require extensive filtering to remove irrelevant data and ensure data quality.

The dataset includes reviews of various companies with columns representing the review date, job details, location, reviewer status, and feedback. Reviews are categorized into sub-sections, covering Career Opportunities, Compensation and Benefits, Culture and Values, Senior Management, and Work/Life Balance. Additionally, employees can provide recommendations for the firm, CEO, and outlook.

Numeric ratings are assigned based on recommendation levels: "v" for Positive, "r" for Mild, "x" for Negative, and "o" for No Opinion.

To perform sentiment analysis using a dataset instead of retrieving information using an API has different steps. It was necessary to choose the suitable libraries. Figure 4.17 has the libraries used in the developed algorithm, and those include:

- Pandas is a powerful data manipulation and analysis library that provides data structures like dataframes, which are commonly used for handling structured data.
- TextBlob is a powerful library for NLP tasks, including sentiment analysis. It provides an intuitive API for text processing and sentiment polarity calculation.
- Matplotlib is a popular data visualization library in Python. The pyplot module provides a simple interface for creating various types of plots, including pie charts and line plots. It is employed to visualize the sentiment analysis results using a pie chart, allowing for a clear representation of the distribution of positive, negative, and neutral sentiments among the reviews.

- The re library is imported to work with regular expressions. Regular expressions are patterns used to match and manipulate text data, which can be helpful for filtering and processing strings.

- NLTK: In this algorithm, NLTK is used to tokenize the text and remove stopwords during the text preprocessing step.

```
import pandas as pd
from textblob import TextBlob
import matplotlib.pyplot as plt
import re
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

Figure 4.17: Python Libraries - Sentiment Analysis

As mentioned previously text preprocessing has a major role when working with text, in this algorithm it involves techniques such as:

- Lowercasing: The reviews' text is converted to lowercase letters. This normalization step ensures that words with different cases are treated as the same and avoids redundancy in the analysis. For example, "Good" and "good" would be considered identical after lowercasing.

- Special Character and Number Removal: The re library is used to remove special characters and numbers from the text. Regular expressions are employed to match and remove any characters that are not letters or whitespace. This step helps eliminate noise from the text data, focusing solely on the words' content and structure.

- Tokenization: Tokenization is the process of splitting the text into individual tokens or words. The nltk library's wordtokenize function is utilized to tokenize the reviews. Tokenization allows for analyzing and processing individual words separately, which is essential for sentiment analysis.

- Stopword Removal: Stopwords are common words (e.g., "the," "and," "is") that often do not contribute much to the sentiment or meaning of the text. The nltk library's stopwords module provides a set of predefined stopwords for different languages. In this algorithm, English stopwords are employed to filter out these common words and reduce noise in the sentiment analysis.

- Joining Tokens: After removing the stopwords, the remaining tokens are joined back together to form a clean review. Joining the tokens helps maintain the context and structure of the original text, facilitating accurate sentiment analysis.

The code loads the dataset containing the reviews using a function from pandas. The user is prompted to enter the name of the firm they want to search for. The code filters the dataset to retrieve reviews that contain the specified firm's name.

Sentiment analysis is performed using the TextBlob library. (Figure 4.18)

```
polarity_scores = []
for review in clean_reviews:
    blob = TextBlob(review)
    polarity_scores.append(blob.sentiment.polarity)
```

Figure 4.18: TextBlob to perform Sentiment Analysis

When using TextBlob to analyze the sentiment of a text, numeric values for both polarity and subjectivity are provided. The polarity value indicates the degree of negativity or positivity within a sentence. This allows to determine whether a sentence leans towards a negative or positive sentiment. On the other hand, subjectivity refers to the extent to which a text is objective or subjective.

The TextBlob library lacks a dedicated built-in model for sentiment analysis. Instead, it relies on a default pre-trained model known as "Pattern" for tasks like sentiment analysis. "Pattern" functions as a text mining module and provides various capabilities, including sentiment analysis.

TextBlob employs a sentiment-calculating algorithm, the "Pattern" model, that rates each word in its lexicon. This algorithm assigns a sentiment score to each word, indicating whether it is positive, negative, or neutral. These sentiment scores are then used to calculate the overall sentiment of a text.

When analyzing the sentiment of a single word using TextBlob, the technique used is called "averaging". This technique involves applying it to the polarity values in order to calculate a polarity score for each individual word. This same procedure is then applied to every single word in a given text, resulting in a combined polarity score for larger texts.

The process begins by assigning a polarity value to each word in the text, indicating whether it conveys a positive, negative, or neutral sentiment. These polarity values are

then averaged to derive an overall polarity score for each individual word.

This process is iterated for every word in the text. The polarity score is calculated for each word, and these scores are subsequently combined to generate a final polarity score for the entire text. This aggregated polarity score represents the overall sentiment conveyed by the larger text.

The code generates a pie chart using the methods in Figure 4.30 to visualize the sentiment analysis results. The pie chart shows the distribution of positive, negative, and neutral sentiments based on the polarity scores. The pie chart is plotted using plt.pie() with the specified labels, colors, and formatting options.

```
pos_count = sum(1 for score in polarity_scores if score > 0)
neg_count = sum(1 for score in polarity_scores if score < 0)
neu_count = sum(1 for score in polarity_scores if score == 0)
labels = ['Positive', 'Negative', 'Neutral']
values = [pos_count, neg_count, neu_count]
colors = ['#00ff00', '#ff0000', '#cccccc']
plt.pie(values, labels=labels, colors=colors, autopct='%1.1f%%')
plt.title("Sentiment Analysis of Reviews for {}".format(firm))
plt.show()
```

Figure 4.19: Sentiment Analysis Results Plot

In the end, the number of reviews for the specified firm using string formatting is displayed.

4.3 Results

The following section presents a comprehensive analysis of the results obtained through the implementation of this project.

4.3.1 Content Matching Tool

As cities around the world continue to face obstacles related to population growth, urbanization, and sustainability, many are turning to smart city solutions to improve the lives of their citizens and optimize city operations. To achieve this transformation, partnerships between cities and private companies, startups, academic institutions, and other stakeholders are essential. One area where these partnerships can be particularly valuable is in the development of smart transportation systems and infrastructure such as smart buildings, streetlights, and waste management systems.

The algorithm presented helps to solve the difficulty of finding partners by using TF-IDF vector representation of text and Cosine Similarity as a metric to determine the similarity between two documents. This allows the code to compare the similarity between the needs and offers from potential partners in a numerical format, making it easier to prioritize which partnerships to analyze first. This project applies TF-IDF and Cosine Similarity to the task of matching potential partners, which is a different application than recommendation systems that use these techniques for personalized recommendations.

The tool created is capable of matching the needs with the skills and offers. However, it is important to note that even with the implementation of this automated tool, HR teams are still crucial to find a good match for the partnership. While the tool can help filter and prioritize corporate profiles, it cannot replace the human touch needed for recruitment. A human touch is still required to ensure that the process of choosing a partner is fair, objective, and aligned with the company's or organization's values and culture.

In Figure 4.20, it is possible to analyse how the algorithm helps to match the content between the proposal and the potential partner.

To test the system, using the data in the candidates dataset, three candidates were choosen for a project in the projects dataset, "Smart Healthcare for Enhanced Patient Care".

Figures 4.21 and 4.22, show how the system would handle a description that it is not similar to the project description. Although the algorithm finds some overlap in content, when the common words are analysed it is possible to conclude that they are general words and don't relate directly with the nature of the project. The HR team should exclude this candidate.

The description of a company that has the requirements to develop the project is tested in Figures 4.23 and 4.24, By looking at the common words, the conclusion is that this company is a suitable candidate to have in consideration by the HR Team, and should be the first in the list. Then a third candidate will be tested, only after having the results the HR can organize the final list of suitable candidates.

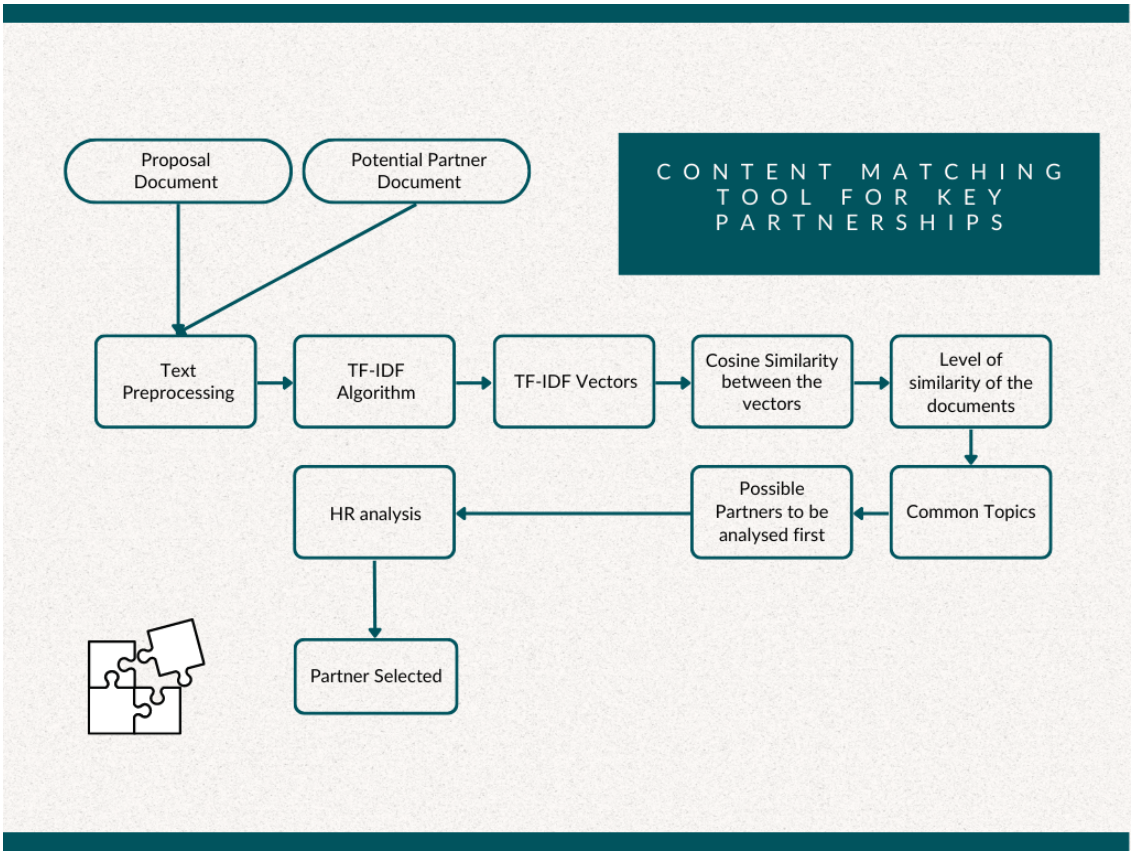

Figure 4.20: Content Matching Tool

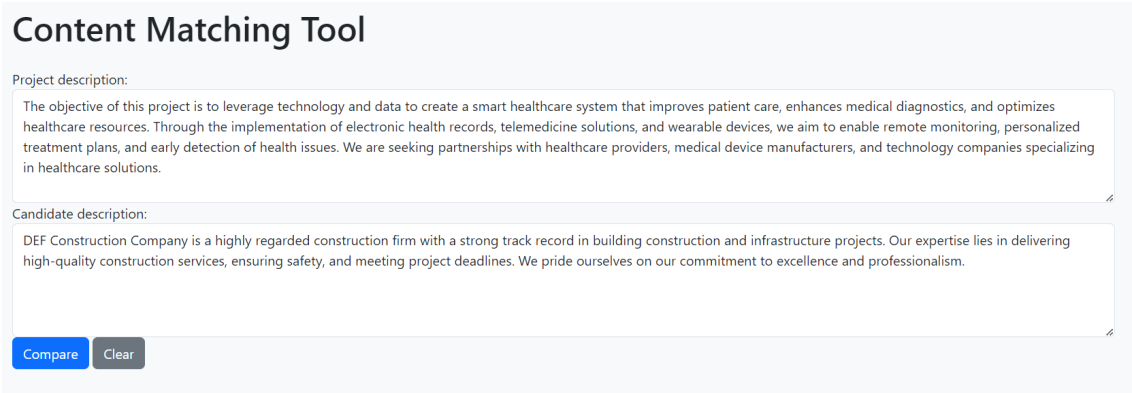

Figure 4.21: Analysis - Candidate 1

Finally, the third candidate, Figures 4.25 and 4.26, have the description of this candidate, and the result of the comparison. This candidate might be a good partner for the project, although there are more common words when compared to candidate two, the cosine similarity value is a bit lower.

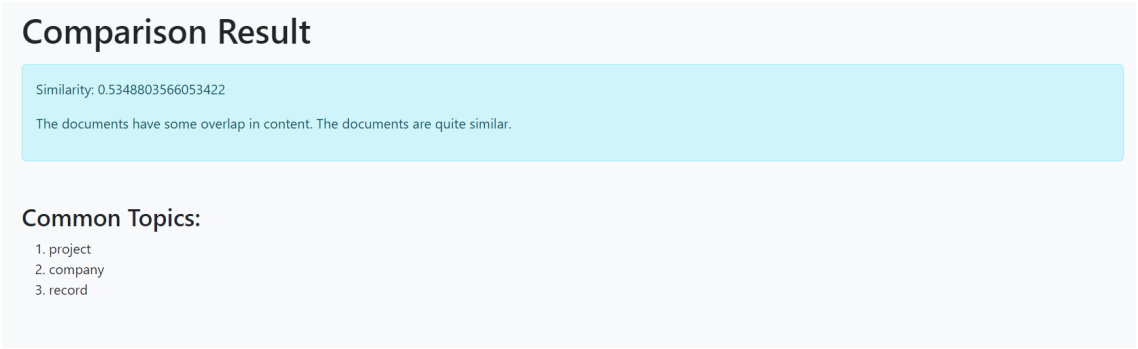

Figure 4.22: Result - Candidate 1

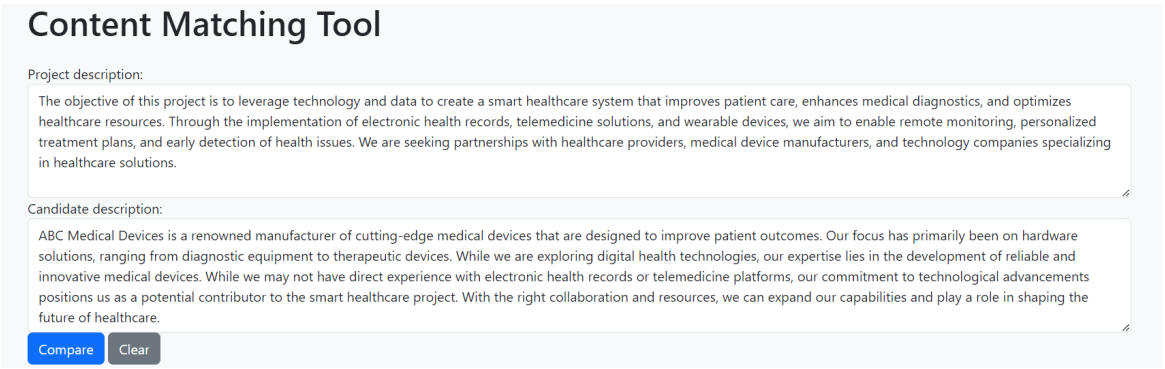

Figure 4.23: Analysis - Candidate 2

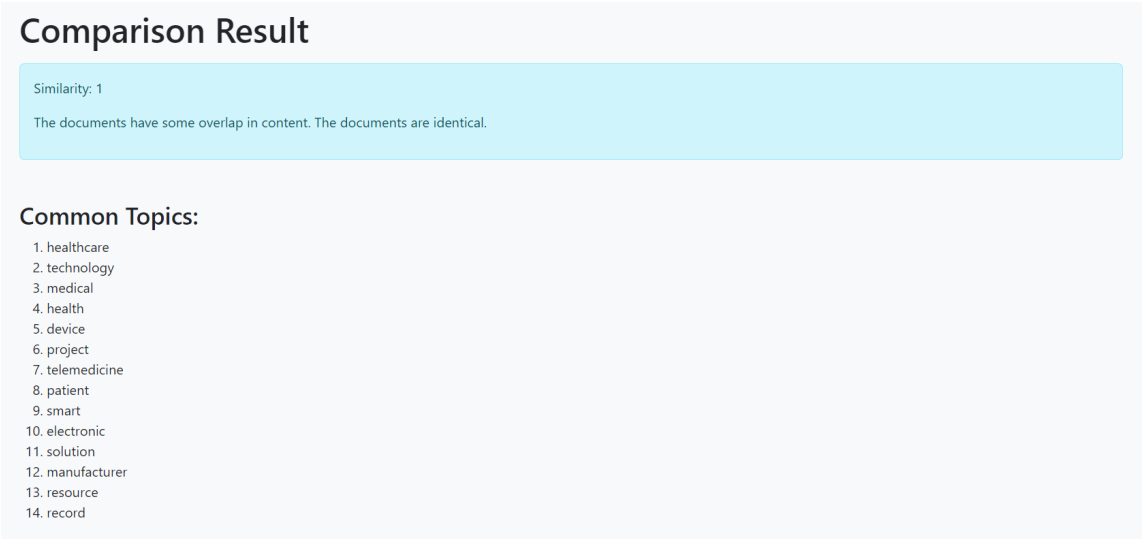

Figure 4.24: Result - Candidate 2

With the automated matching tool, the initial screening process can be made more efficient, allowing HR teams to focus their attention on the most promising candidates. By using NLP techniques for content matching, the system can quickly sift through large volumes of texts from the potential partners and identify those that most closely match the

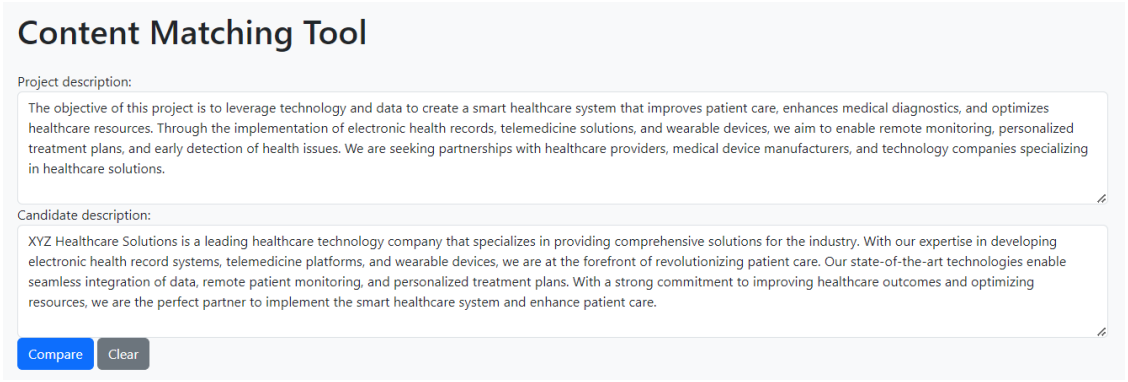

Figure 4.25: Analysis - Candidate 3

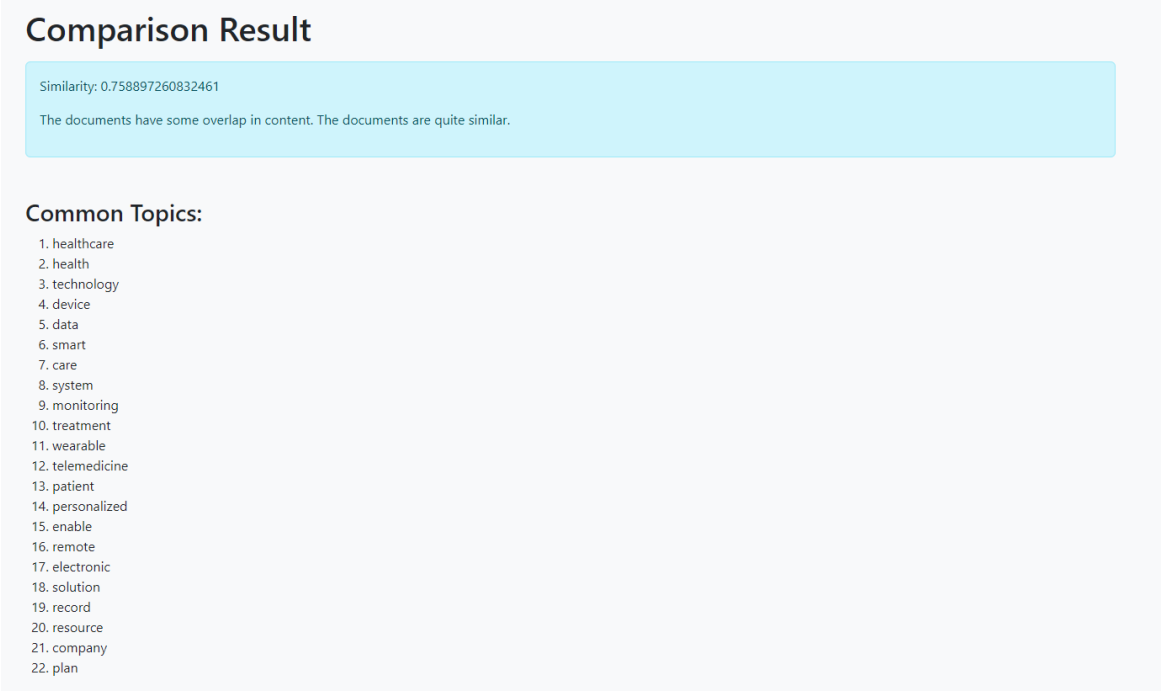

Figure 4.26: Result - Candidate 3

project requirements. This can significantly reduce the time and effort spent on manual screening.

The example shown proved that the HR team is still needed to evaluate the candidates and organize their list to prioritize interviews of the most adequate candidates.

Although there are already tools available, such as ChatGPT, that can effectively perform content matching, certain concerns regarding privacy and accuracy may arise. These concerns may lead companies to reconsider this approach and opt for alternative solutions.

Privacy concerns can emerge due to the nature of content matching, which involves

analyzing and comparing sensitive information from different sources. Companies understand the importance of protecting sensitive data, especially when it involves personal or confidential information. The potential risks of sharing this data with external tools or platforms may include unauthorized access, data breaches, or even misuse of the information. As a result, many companies are cautious and hesitant to share such data, preferring to keep it within their own systems where they can exercise greater control and ensure its confidentiality.

In terms of privacy, it's important to note that candidates for projects may also have concerns about sharing their personal or company data, especially if there is a possibility that this information could end up on a public platform accessible to everyone.

Candidates may hesitate to disclose sensitive information, such as proprietary knowledge or confidential business strategies, fearing potential risks associated with exposing their data to a wide audience. They may prefer to maintain control over the dissemination of their information and restrict its access to trusted parties.

While AI models like ChatGPT have proven to be effective, there are concerns regarding their accuracy. The algorithms used in these tools may generate false positives or miss relevant matches, which can result in incorrect content recommendations. Inaccurate content matching can lead to user dissatisfaction, miscommunication, and even legal implications. Companies relying solely on these tools should consider the potential risks associated with inaccuracies and seek more reliable alternatives.

In order to evaluate the algorithm's performance, a comparison was made between the algorithm's predictions and real-world correspondences. A supplementary dataset (ground truth df) was created to contain information about the actual matches between projects and candidates. This dataset includes columns such as "Project," "Actual Candidate," and "Correspondent," where "Correspondent" is marked as "True" if the candidate is a suitable match for the project, and "False" otherwise.

For each project, three candidate entries were selected by ChatGPT, each with a different level of relevance or suitability to the project.

The process involves loading the ground truth dataset, merging it with the algorithm's results based on the project names, and calculating accuracy by comparing the predicted candidates with the actual corresponding candidates. The resulting accuracy value pro-

vides a measure of how well the algorithm aligns with the real-world matches, which has proven that ChatGPT does some incorrect content recommendations, Figure 4.27.

```
   ...: merged_df = pd.merge(results_df, ground_truth_df,
left_on='Project Number', right_on='Projeto', how='inner')
   ...:
   ...: # Calcular a precisão
   ...: correct_matches = merged_df[merged_df['Best Candidate'] ==
merged_df['Candidato']]
   ...: accuracy = len(correct_matches) / len(merged_df)
   ...:
   ...: # Imprimir a precisão
   ...: print(f"Accuracy: {accuracy:.2%}")
Accuracy: 10.00%
```

Figure 4.27: Accuracy for ChatGPT matching candidates

4.3.2 Sentiment Analysis

In this project, the primary objective was to gather tweets related to a specific topic. The initial preference was to utilize the Twitter API for accessing real-time and comprehensive tweet data. However, the implementation of this choice faced a significant obstacle due to the specific requirements for obtaining the necessary credentials to access the Twitter API.

Access to the Twitter API requires the development of a dedicated application and the acquisition of API keys, tokens, and secrets. Unfortunately, these credentials are not universally available and are typically provided only under specific conditions. These conditions often involve a detailed application process and meeting specific criteria set by Twitter.

At the outset of this project, the requisite application for API access had not been developed, hindering the ability to satisfy the criteria for obtaining API credentials. Moreover, considering the limitations of the free version of the Twitter API, including constraints on data volume and type, it became clear that alternative approaches needed to be explored.

Recognizing the challenges and constraints associated with obtaining API credentials, a pragmatic decision was made to opt for an alternative method. Instead of real-time data collection through the Twitter API, the project utilized a pre-existing dataset related to the specific topic of interest. This approach not only circumvented the immediate barriers posed by the API access requirements but also allowed for a more straightforward and

efficient initiation of the project.

While the use of the Twitter API remains a valuable option for future stages of the project, the decision to employ a pre-existing dataset at the project's outset was driven by the need to overcome the initial hurdles associated with API access. This approach ensured that the project could commence without delays and provided a solid foundation for subsequent phases of analysis and exploration.

The section of the code in Figure 4.28 was developed to assess the accuracy of the developed sentiment analysis algorithm. The process involves:

- Creating Sentiment Labels: Based on the calculated sentiment polarity scores, sentiment labels are assigned to each review. A label of 1 is assigned for positive sentiment, -1 for negative sentiment, and 0 for neutral sentiment. These labels are then added as a new column ('sentiment') to the DataFrame "firm reviews".

- Splitting Data into Training and Testing Sets: The dataset is divided into training and testing sets using the "train test split" function from scikit-learn. This allows the algorithm to be trained on one subset of the data and evaluated on another.

- Vectorizing Text Data using TF-IDF: The text data is converted into numerical features using the TF-IDF vectorizer. This step is essential for training a machine learning model.

- Training the Naive Bayes Classifier: A Multinomial Naive Bayes classifier is instantiated and trained using the vectorized training data, which is 80 per cent of the reviews for a certain company.

- Predicting Sentiments for Test Data: The trained classifier is used to predict sentiments for the test data, which is 20 per cent of the reviews for a certain company.

- Calculating Accuracy: The accuracy of the sentiment analysis is calculated by comparing the predicted sentiments with the actual sentiments in the test set. The accuracy is then printed.

- Calculating the Confusion Matrix: The confusion matrix is calculated using the "confusion matrix" function. It provides detailed information about the model's perfor-

mance, showing correct and incorrect predictions for each sentiment class (positive, negative, neutral).

```
# Create labels for sentiment: 1 for positive, 0 for neutral, -1 for negative
sentiment_labels = [1 if score > 0 else -1 if score < 0 else 0 for score in polarity_scores]
firm_reviews['sentiment'] = sentiment_labels

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(firm_reviews['review'], firm_reviews['sentiment'], test_size=0.2, rando

# Vectorize the text data using TF-IDF
vectorizer = TfidfVectorizer()
X_train_vectorized = vectorizer.fit_transform(X_train)
X_test_vectorized = vectorizer.transform(X_test)

# Train Naive Bayes classifier
classifier = MultinomialNB()
classifier.fit(X_train_vectorized, y_train)

# Predict sentiments for test data
predictions = classifier.predict(X_test_vectorized)

# Calculate accuracy
accuracy = accuracy_score(y_test, predictions)
print(f"Sentiment Analysis Accuracy for {firm}: {accuracy:.2f}")

# Calculate confusion matrix
confusion_mat = confusion_matrix(y_test, predictions)
print("Confusion Matrix:")
print(confusion_mat)
```

Figure 4.28: Evaluating the algorithm accuracy

When the algorithm analyzes the whole dataset for a certain company, is this case Aldi, it is notable that the positive reviews are more than negative and neutral combined. In 931 reviews (Figure 4.29) 58,5 per cent are positive reviews (Figure 4.30).

```
Number of reviews for Aldi: 931
```

Figure 4.29: Reviews for "Aldi"

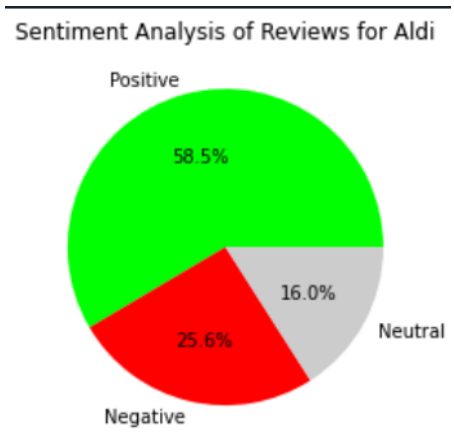

Figure 4.30: Pie Chart of Sentiment of Reviews for "Aldi"

Figure 4.31 represents the ratio of correctly classified instances (reviews) to the total instances in the test dataset. Essentially, the model achieved an 82 per cent accuracy in predicting the sentiment of reviews as positive, negative, or neutral.

The suitability of this accuracy is based on the nature of the sentiment analysis task, including factors such as class balance (the proportion of positive, negative, and neutral reviews).

The confusion matrix (Figure 4.31) has three classes (sentiments): negative, neutral, and positive. The matrix shows the number of instances for each combination of actual and predicted classes.

The rows are identified as follows: the first row represents the actual negative class, the second row the actual neutral class, and the third row the actual positive class.

The columns correspond as follows: the first column represents the predicted negative class, the second column the predicted neutral class, and the third column the predicted positive class.

Breaking down the values:

- True Negatives (TN): 0 - This signifies the count of reviews accurately predicted as negative among those that are actually negative. In this instance, there are zero true negatives (instances predicted as negative that are indeed negative).
- False Positives (FP): 0 - Representing the count of reviews that are actually negative but inaccurately predicted as positive. No false positives were observed.
- False Negatives (FN): 8 - Denoting the count of reviews actually neutral (or negative) but inaccurately predicted as positive. There are 8 instances of false negatives.
- True Positives (TP): 153 - This indicates the count of reviews accurately predicted as positive among those that are genuinely positive. There are 153 true positives.

Considering True Negatives and False Positives: For the negative class, both true negatives and false positives are zero, indicating the model's absence of accurate predictions for negative reviews. Considering False Negatives and True Positives: Regarding the positive class, there are 8 false negatives and 153 true positives. This implies the model's partial success in predicting positive reviews but also missing 8 positive instances.

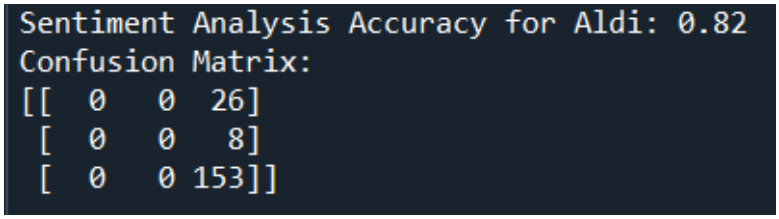

Figure 4.31: Accuracy for Sentiment Predicted for Reviews of Aldi

After analyzing the distribution of reviews in the dataset and testing it across different companies, it was determined that the vast majority of reviews for these companies are positive. This prevalence of positive sentiment in the dataset contributes to the high accuracy observed in the sentiment analysis results. The model excels in correctly predicting positive sentiments, given their significant representation in the dataset. It's important to acknowledge that the high accuracy is influenced by the skewed distribution of sentiment classes.

Measuring sentiment accurately from textual phrases is a complex task due to the inherent subjectivity and context-dependency of language. Sentiment analysis faces challenges such as sarcasm, nuanced expressions, and varying interpretations. Sarcasm, for instance, poses a considerable challenge as it involves the use of words to convey a meaning opposite to their literal interpretation. An example of sarcasm could be a statement like "Great, another flat tire," where the word "great" is used ironically to express frustration. Nuanced expressions, such as subtle positive or negative connotations, further complicate sentiment analysis, as these expressions may not be easily discernible without a deep understanding of the context.

Moreover, varying interpretations of sentiment exist among individuals, making it difficult to establish universal criteria for what constitutes positive, negative, or neutral sentiment. Cultural and individual differences contribute to diverse perspectives on language, leading to subjective interpretations of sentiment. Additionally, sentiment can be context-dependent, where the meaning of a phrase may change based on the surrounding text or the broader context of a conversation.

These multifaceted challenges underscore the intricacy of sentiment analysis and highlight the need for advanced NLP techniques to capture the subtleties of human expression accurately. While sentiment analysis algorithms continue to evolve, achieving consistently

high accuracy remains a complex endeavor due to the dynamic and nuanced nature of language.

Chapter 5

Conclusion

The implementation of an automated matching tool using AI and NLP techniques can help make the recruitment process more efficient, while also ensuring that HR teams continue to play a crucial role in the process of making new partnerships. The tool can help filter and sort corporate profiles, allowing HR teams to focus their attention on the most promising candidates.

In addition to private companies, partnerships with academic institutions can also be valuable in developing and implementing smart city solutions. Collaborating with universities and research institutions can provide cities with access to the latest research, ideas, and innovations in the field of smart cities.

As a final point, the need for cities to evolve to smart cities is becoming increasingly important, and key partnerships with private companies, startups, academic institutions, and other stakeholders are essential to achieving this transformation. By working together, cities and their partners can create innovative solutions that address the challenges of urbanization and improve the quality of life for citizens.

5.1 Future Work

In future work, it is planned to further develop the GUI by adding additional functionalities. One of the proposed enhancements is to enable the inclusion of multiple candidate texts for comparison with the project description, resulting in a prioritized list of potential partners for interviews.

Additionally, a similar tool will be developed to allow potential partners to assess and identify projects that align closely with their interests. This tool will provide valuable insights for potential partners to determine which projects they should consider applying for.

The provided code currently relies on similarity measures, such as cosine similarity, without explicitly utilizing a machine learning algorithm. While these techniques offer a foundation for matching projects and candidates based on textual information, there is room for improvement by incorporating machine learning algorithms.

In future iterations, enhancing the system could involve implementing machine learning models that learn from labeled data to make more accurate predictions. By training a model on historical matches between projects and candidates, the system can potentially identify more complex patterns and relationships, leading to improved accuracy in matching.

Integrating machine learning algorithms could enable the system to adapt and generalize better to a broader range of scenarios, ultimately enhancing its effectiveness in making accurate project-candidate matches.

Regarding sentiment analysis, once the application is deployed in the market, it will be possible to leverage Twitter and LinkedIn APIs for data collection. This expanded dataset will contribute to improved accuracy in sentiment analysis, thereby enhancing the understanding of sentiment expressed in the texts. Considering the existing class imbalance, a comprehensive examination of precision, recall, and F1 score for each sentiment class becomes imperative while obtaining new data for sentiment analysis. It is crucial to recognize the increased likelihood of encountering imbalances among sentiment classes during the acquisition of new data for sentiment analysis.

These future developments will expand the capabilities of the content matching tool, providing users with more comprehensive information for decision-making in the partner selection process. Through the utilization of APIs and enhanced data collection, increased precision and effectiveness are expected in matching projects with potential partners.

Bibliography

[1] Odeyinka Abiola et al. "Sentiment analysis of COVID-19 tweets from selected hashtags in Nigeria using VADER and Text Blob analyser". In: *Journal of Electrical Systems and Information Technology* 10.1 (2023), pp. 1–20.

[2] Surabhi Adhikari et al. "Nlp based machine learning approaches for text summarization". In: *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE. 2020, pp. 535–538.

[3] Muzamil Ahmed et al. "Automated Question Answering based on Improved TF-IDF and Cosine Similarity". In: *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE. 2022, pp. 1–6.

[4] Hejab Alfawareh and Shaidah Jusoh. "Intelligent decision support system for CV evaluation based on natural language processing". In: *International Journal of Advanced and Applied Sciences* 6.4 (2019), pp. 1–8.

[5] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[6] Kai-Wei Chang, Vinod Prabhakaran, and Vicente Ordonez. "Bias and fairness in natural language processing". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): Tutorial Abstracts*. 2019.

[7] Erik De Vries, Martijn Schoonvelde, and Gijs Schumacher. "No longer lost in translation: Evidence that Google Translate works for comparative bag-of-words text applications". In: *Political Analysis* 26.4 (2018), pp. 417–430.

[8] Gerard Deepak, Varun Teja, and A Santhanavijayan. “A novel firefly driven scheme for resume parsing and matching based on entity linking paradigm”. In: *Journal of Discrete Mathematical Sciences and Cryptography* 23.1 (2020), pp. 157–165.

[9] Dg. *Glassdoor job reviews*. July 2023. URL: `https://www.kaggle.com/datasets/davidgauthier/glassdoor-job-reviews/data`.

[10] Ankur Goel, Jyoti Gautam, and Sitesh Kumar. “Real time sentiment analysis of tweets using Naive Bayes”. In: *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*. IEEE. 2016, pp. 257–261.

[11] Anuja P Jain and Padma Dandannavar. “Application of machine learning techniques to sentiment analysis”. In: *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*. IEEE. 2016, pp. 628–632.

[12] Daniel Jurafsky and James H Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*.

[13] Alina Köchling and Marius Claus Wehner. “Discriminated by an algorithm: a systematic review of discrimination and fairness by algorithmic decision-making in the context of HR recruitment and HR development”. In: *Business Research* 13.3 (2020), pp. 795–848.

[14] Akhil Alfons Kodiyan. “An overview of ethical issues in using AI systems in hiring with a case study of Amazon’s AI based hiring tool”. In: *Researchgate Preprint* (2019), pp. 1–19.

[15] Swatee B Kulkarni and Xiangdong Che. “Intelligent software tools for recruiting”. In: *Journal of International Technology and Information Management* 28.2 (2019), pp. 2–16.

[16] Ema Kušen et al. “Identifying emotions in social media: comparison of word-emotion lexicons”. In: *2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. IEEE. 2017, pp. 132–137.

[17] Jing Li et al. “A survey on deep learning for named entity recognition”. In: *IEEE Transactions on Knowledge and Data Engineering* (2020).

[18] Ram Krishn Mishra, Siddhaling Urolagin, et al. “A Sentiment analysis-based hotel recommendation using TF-IDF Approach”. In: *2019 international conference on computational intelligence and knowledge economy (ICCIKE)*. IEEE. 2019, pp. 811–815.

[19] Adrian Pasat, Andrei Birdici, and Iulia Pop. “AN INTERNSHIP CAMPAIGN CASE STUDY SHOWING RESULTS OF ENHANCED RECRUITMENT PROCESSES USING NLP”. In: *The International Scientific Conference eLearning and Software for Education*. Vol. 2. ” Carol I” National Defence University. 2021, pp. 222–231.

[20] Bishwo Prakash Pokharel. “Twitter sentiment analysis during covid-19 outbreak in nepal”. In: *Available at SSRN 3624719* (2020).

[21] R Poonguzhali et al. “Sentiment analysis on linkedin comments”. In: *International Journal of Engineering Research & Technology IJERT (ICONNECT)* 6.7 (2018), p. 415.

[22] Saurav Pradha, Malka N Halgamuge, and Nguyen Tran Quoc Vinh. “Effective text data preprocessing technique for sentiment analysis in social media data”. In: *2019 11th international conference on knowledge and systems engineering (KSE)*. IEEE. 2019, pp. 1–8.

[23] Wisam A Qader, Musa M Ameen, and Bilal I Ahmed. “An overview of bag of words; importance, implementation, applications, and challenges”. In: *2019 international engineering conference (IEC)*. IEEE. 2019, pp. 200–204.

[24] Xavier Schmitt et al. “A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate”. In: *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE. 2019, pp. 338–343.

[25] “Smart Cities Market Size, share: 2022 - 2027”. In: URL: https://www.marketsandmarkets.com/Market-Reports/smart-cities-market-542.html?gclid=EAIaIQobChMIwLbdrcSF_AIVx4TVCh3gPAZ3EAAYASAAEgIMYvD_BwE.

[26] George Suciu et al. “Design of an internship recruitment platform employing nlp based technologies”. In: *ECAI 2018-International Conference, June 2018*. 2018, pp. 1–6.

[27] Dongyang Yan et al. “Network-based bag-of-words model for text classification”. In: *IEEE Access* 8 (2020), pp. 82641–82652.

[28] Gisela Yunanda, Dade Nurjanah, and Selly Meliana. “Recommendation system from microsoft news data using TF-IDF and cosine similarity methods”. In: *Building of Informatics, Technology and Science (BITS)* 4.1 (2022), pp. 277–284.

[29] Rui Zhao and Kezhi Mao. “Fuzzy bag-of-words model for document representation”. In: *IEEE transactions on fuzzy systems* 26.2 (2017), pp. 794–804.

[30] Zhiliang Zhu et al. “Hot topic detection based on a refined TF-IDF algorithm”. In: *IEEE access* 7 (2019), pp. 26996–27007.

Attachments

Content Matching for City Improvement

Margarida Rodrigues
Department of Informatics
Polytechnic of Viseu,
Viseu, Portugal
estgv16061@alunos.estgv.ipv.pt

Filipe Pinto
Altice Labs, Rua Eng.º José Ferreira Pinto Basto,
Aveiro, Portugal
CISeD – Research Centre in Digital Services, Polytechnic of Viseu,
Viseu, Portugal
filipe-c-pinto@alticelabs.com

***Abstract*—Developing new services or improving existing ones is becoming more accessible with the evolution of NLP techniques. Chatbots are a known example of an NLP-based service; they can interact with humans using text messages or natural language. NLP grants, however, the development of other types of services based on natural languages, such as machine translation, email spam detection, information extraction, content summarization, and question answering. A current need for employers and job seekers is a system that can match content (text) from a project offers description with the job seeker's report to promote discovery between demand and supply by finding common patterns in different textual descriptions. This paper presents an implementation of an automated tool with AI and NLP to match needs and concrete ideas for innovation with the skills and offers of the business sector, including start-ups and entrepreneurs.**

***Index Terms*—natural language processing, nlp, content matching, stemming and lemmatization, term frequency-inverse document frequency, TF-IDF, TF, IDF, stop words, cosine similarity, smart cities**

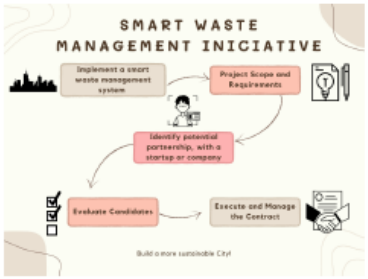

Fig. 1. Example of a Smart City Project

I. Introduction

This section provides background information on using NLP for Information Retrieval (IR) and content matching. The focus of this research is discussed and justified, and the overall research aim is outlined.

A. Background Information

Scientific and technological developments in Information and Communication Technologies (ICT) are the main contributors to the sustainable growth of cities worldwide. Smart cities' global market is expanding rapidly; in 2022, it had an estimated increase of 482 billion euros [1]. The evolution of 5G and the Internet of Things (IoT) technologies have contributed to improving the efficiency of urban services by addressing some of the main gaps in energy, mobility, security, privacy and environmental sustainability. However, there is still a journey ahead in this field.

One of the branches of AI and linguistics is NLP, whose function is to make computers understand the statements or words written or spoken in human language. Information extraction helps to collect information from machine-readable documents automatically, so it is a key step in NLP.

Organizations are constantly evolving into more complex environments due to global competition, creating the need to have partners and service providers that meet their needs and allow them to improve the quality of their services. Human resources teams face various challenges, including the difficulty in identifying the most relevant partners, limited availability of data or resources to assess potential partners, and the lack of transparency or standardization in the selection process. The mismatch in expectations between urban needs/ideas and the business sector, limited knowledge or awareness of potential partners among urban stakeholders, and inefficient or ineffective communication channels can further complicate the process. One critical challenge in this context is the amount of time it takes to choose the ideal partner.

B. Overall Research Aim

Cities require and aim to promote the development of new products, processes and services with high potential, contributing to integrated urban management and efficient and innovation-catalysing based on specific contributions to the implementation and interoperability of urban platforms.

Start-ups can take advantage to leverage the need of established companies to maintain their place in the market while being more innovative and competitive. New companies can bring this innovation and maintain high-quality standards. However, it is crucial that the establishment of a partnership is quick and practical to keep up with the current market trends.

Figure 1 is a visual representation of the process and help stakeholders better understand how the contract with a startup or company will be developed and managed.

This work aims to use AI and NLP techniques to implement an automated matching tool to connect needs and specific

ideas for innovation within cities with the skills and offers of the business sector, including start-ups and entrepreneurs. By improving the efficiency and accuracy of the matching process, human resources teams can focus on the important task of ensuring the process is fair, objective, and aligned with the organization's values and culture.

C. Paper Structure

The paper begins with a study of previous research and projects using NLP. After this analysis, it will be possible to select techniques and approaches to solve the problem more effectively and efficiently. According to a preliminary selection of techniques, this paper explains how they are used in the project and how they were implemented in previous works.

The remainder of this work is organized as follows. Section II, presents the current state of research in NLP applied to content matching. The practical work is described on section III. Section IV shows the final product preview.Finally, in Section V conclusions are drawn, followed by the introduction of future work guidelines in section VI.

II. Related Works

NLP is a sub-field of computer science and AI that focuses on studying the interactions between computers and human language. It involves devising algorithms and techniques to analyze, understand, and generate human language, and its practical applications span various domains, including machine translation, text classification, and conversational agents.

In content matching, NLP is crucial in understanding and analyzing language. NLP techniques can extract critical information from job descriptions or resumes and categorize job postings or candidates based on their language and content.

This section provides an overview of the current state of research in NLP applied to content matching. This section will describe the most important and relevant studies, theories, and methods developed in these areas and any current debates or controversies. It is divided into several sections, covering various aspects of NLP, including an introduction to NLP and its methodologies, followed by its use in content mathing contexts; specific techniques such as the TF-IDF algorithm are explored and also the ethical implications of applying NLP to make decisions about new partnerships.

A. Introduction to NLP and its Techniques

The amount of data collected online has increased significantly in recent years due to the proliferation of internet-connected devices, websites and apps, and social media platforms. This data, often in text, can include various information such as emails, social media posts, documents, and transcripts of spoken conversations. Text data is typically unstructured and is analyzed using NLP techniques, which involve using computer algorithms to understand and interpret the meaning of the text.

NLP can be applied to various aspects of the recruitment process, including resume parsing, job posting analysis, and candidate evaluation. For example, NLP can be used to extract important information from resumes, such as job titles, skills, and work experience, which can help streamline the process of sorting and reviewing resumes. NLP can also be used to analyze job postings and identify key skills and qualifications that are required for a particular role. In addition, NLP can be used to evaluate candidates by analyzing their responses to questions or their writing samples, and to identify their fit for a particular role based on their language and communication skills.

NLP can also be used in the analysis of employee feedback, both during and after the recruitment process. For example, NLP can be used to identify trends and patterns in employee feedback and to identify areas for improvement in the workplace. This can help organizations to better understand the needs and concerns of their employees and to create a more positive and productive work environment.

The Named Entity Recognition (NER) is crucial in information extraction; it is used in machine translation, question answering, IR, and summarization. NER is a task that aims to identify text that mentions entities that are real-world objects that have a name, such as a person, organization, location, date, time, or product. The study [2] aimed to evaluate the performance of these libraries in terms of precision, recall, and F1 score on two data sets: CoNLL-2003 and OntoNotes 5.0. Precision measures the proportion of correctly identified entities among all entities identified by the system. In contrast, recall measures the proportion of correctly identified entities among all entities in the text. F1 score is the harmonic mean of precision and recall, and provides an overall measure of the system's performance. The study found that SpaCy and StanfordNLP performed the best on both data-sets, with F1 scores above 90 percent. NLTK and OpenNLP had lower F1 scores, around 80 percent, while Gate had the lowest F1 score of around 70 percent. The authors also noted that the performance of the NER libraries varied depending on the type of entity and the data-set size.

B. NLP for Content Matching in business: Tools and Applications

This section will examine the various NLP techniques used in business, the evaluation metrics used, and the results obtained. Additionally, it will explore the potential of NLP to improve the efficiency and effectiveness of hiring processes in real-world scenarios.

The article [3] presents a case study on using NLP to enhance recruitment processes. It aims to show how NLP can improve the efficiency and effectiveness of recruitment processes, particularly in the context of an internship campaign.

The study collected resumes and cover letters of candidates applying for an internship campaign and used NLP techniques to extract relevant information and perform a semantic analysis of the resumes and cover letters. The study's results showed that using NLP in the hiring process improved the efficiency and effectiveness of the internship campaign by reducing the time and resources needed to review resumes and cover letters and by identifying the most qualified candidates.

The article [4] discusses the use of AI in three critical areas of recruitment and selection: candidate identification, candidate engagement, and candidate selection. To identify candidates, AI-based tools use supervised learning algorithms to match job postings with suitable candidates from resume databases and social platforms. To engage candidates, NLP and chat-bots interact with candidates, answer their questions, and schedule interviews. Lastly, AI-based tools for candidate selection use pattern recognition and analysis to measure facial expressions, voice, and tone during video interviews and compare the data with other candidates or successful employees. It was concluded that the benefits of using these tools are automation of repetitive tasks, cost savings, and prevention of recruiter burn-out. However, further research is needed to analyze the potential for AI tools to replace recruiters and talent acquisition professionals and to validate the impact of company size and industry type on the implementation of AI-based tools.

The paper [5] describes an Intelligent Decision Support System (IDSS) for evaluating CVs using natural language processing techniques. The proposed system is designed to automate the process of CV evaluation, which is traditionally time-consuming and requires significant human effort. The system is based on algorithms that analyze the CVs and extract relevant information, such as the candidate's skills, education, experience, and achievements. The system then uses this information to recommend the candidate's suitability for a particular job.

The paper provides a detailed description of the algorithms used in the system, including the preprocessing steps, feature extraction, and classification models. The system is trained and evaluated using a data set of 300 CVs and achieves an accuracy of 94.3 per cent. The authors conclude that the proposed system can significantly reduce the time and effort required for CV evaluation while also improving the accuracy and consistency of the evaluation process.

Ethics is an important consideration when applying NLP techniques in terms of strategical partnerships between entities. The use of NLP in this area has the potential to improve efficiency and effectiveness significantly, but it also raises ethical concerns related to privacy, bias, and discrimination.

In a process of finding the best partner for a project, NLP can be used to analyze resumes and job applications to identify candidates who are a good fit. However, there is a risk that NLP algorithms may perpetuate existing biases or discriminate against certain groups of people, such as those from marginalized backgrounds. It's essential to ensure that the data used to train the algorithm is diverse and that the algorithm is evaluated for any potential bias. When collecting data on individuals, it's important to obtain informed consent, to be transparent about how the data will be used, and to take steps to protect the data.

One notable example of ethical concerns related to the use of NLP in hiring is the case of Amazon's automated hiring tool. The company developed an algorithm to help screen job applicants. Still, it was discovered that the algorithm was biased against female candidates because it was trained on resumes submitted to the company over a 10-year period, mostly from men. As a result, Amazon had to discontinue using the algorithm in its recruitment process.

$$TF_{(w,j)} = \frac{\text{(Number of times term w appears in a document)}}{\text{(Total number of terms w in the document)}}$$

Fig. 2. Term Frequency Formula

$$IDF_{(w)} = \log \frac{\text{Total number of documents}}{\text{Number of documents with term w in it}}$$

Fig. 3. Inverse Document Frequency Formula

The article [6] presents an overview of the ethical issues that arise when using Artificial Intelligence (AI) systems in hiring and provides a case study of Amazon's AI-based hiring tool. The study found that using AI in hiring can lead to a number of ethical issues such as bias, discrimination, and lack of transparency. The study also found that there are challenges in addressing these issues, such as the lack of data privacy and security, and the lack of interpretability of the AI systems.

In conclusion, NLP techniques can significantly improve the process of finding a good match for a project, but it's crucial to be mindful of ethical concerns and take steps to mitigate any potential negative impacts.

C. Understanding Text with TF-IDF Algorithm

TF-IDF stands for "term frequency-inverse document frequency". It is a numerical statistic intended to reflect how important a word is to a document in a corpus. The TF-IDF value increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. The TF-IDF algorithm can be used to weight words in text data, which is helpful in IR and text mining.

In NLP, a corpus is a large and structured set of texts. Corpora are used for statistical analysis and hypothesis testing, checking occurrences or validating linguistic rules within a specific language territory. Corpora are also used to train language models and other machine-learning models. In the context of the TF-IDF algorithm, the corpus refers to the set of all documents in which we want to compute the tf-idf values for each word.

The basic formula for TF-IDF is:

- TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document) (See Figure 2)
- IDF(t) = log(Total number of documents / Number of documents with term t in it) (See Figure 3)
- TF-IDF(t) = TF(t) * IDF(t)

The intuition behind this formula is that words that are common in all documents are not very informative and, thus should have a lower weight. Conversely, words that frequently appear in only a few documents are highly informative and

should have a higher weight. The TF-IDF algorithm thus assigns a weight to each word in each document, where the weight is the product of the term's TF and IDF values.

The results of the TF-IDF formula can be analyzed in several ways. One common approach is to use the computed TF-IDF values to identify the most important words in a document or a set of documents. This can be done by sorting the words in descending order of TF-IDF value and selecting the top N words. These words are often considered the keywords or keyphrases of the document and can be used for tasks such as IR, text summarization, and document classification.

The article [7] presents the development of a recommendation system for Microsoft news data using TF-IDF and cosine similarity methods. The system suggests relevant news articles to users based on their preferences. The authors collected and preprocessed the data, used TF-IDF to extract features, and calculated cosine similarity scores to recommend articles. They evaluated the system's performance using precision, recall, and F1-score metrics. The results show that the proposed system outperforms similar systems in precision, recall, and F1-score. The authors also discuss the strengths and limitations of their system and suggest areas for future research.

The paper [8] proposes an automated question answering system that uses an improved TF-IDF and cosine similarity to provide precise and relevant answers to users' queries with high confidence. The study compares different techniques for automatic question answering and finds that rule-based techniques are less effective due to natural language's lack of fixed patterns. The proposed approach pre-processes all repository questions and generates a matrix using the improved TF-IDF model to find the similarity of each user query. The system removes stop-words and applies lemmatization and POS tagging techniques for pre-processing. The empirical analysis-based results show that the proposed technique takes less than five seconds to respond to user queries with maximum similarity and attains up to 84 per cent accuracy.

III. Experimental Setup

A. Construction of the Data Set

A specific data set is likely needed for an automated matching tool. The data collection is necessary to provide a format of information that is going to be used when establishing partnerships of strategic interest, since online there was no data-set that answered these requisites.

Creating a specific data-set for the project would involve several steps, such as:

- Data collection: After consulting a range of sources, including academic papers, industry reports, and case studies of possible smart city projects, it was possible to produce data that contains these projects requirements in terms of technology, logistics and human resources. This data was then used to create a comprehensive data-set that was later used for the development of the solution.
- Data cleaning: This step involved removing any irrelevant or redundant information from the data, such as special characters, punctuation, digits, and stopwords, as well as converting text to lowercase and removing duplicate data.
- Data annotation: This step involved labeling the data with relevant categories or tags, such as the type of need or idea, or the type of skill or offer, so that it can be used for training and evaluating the NLP models.
- Data preprocessing: This step involved applying the different text preprocessing techniques, such as tokenization, stemming, lemmatization, stopword removal, Named-entity recognition, part-of-speech tagging, so that the data is ready for the matching process.

Creating a specific dataset for the project required a significant amount of time and resources, but it is essential for the success of the project. A good dataset would ensure that the matching process is accurate and effective, which is the ultimate goal of the tool.

B. Development of the solution - Content Matching tool for recruitment

1) Text Preprocessing: The goal of preprocessing is to prepare the text data for analysis by cleaning, normalizing, and structuring it. There were studied several approaches to perform text preprocessing. One of the choosen libraries was the Natural Language Toolkit (NLTK), which is a popular Python library used for NLP tasks such as text preprocessing. It provides a range of tools and resources for tasks such as tokenization, stemming, lemmatization, and part-of-speech tagging, among others.

The first step in preprocessing is tokenization, which involves breaking down the text into individual words, known as tokens. Next, the text is cleaned by removing stop words, commonly used words that do not provide much meaning, such as "and" and "the".

Another important preprocessing step is lemmatization, which is the process of reducing words to their base or root form, known as lemma. Additionally, punctuation and lower casing are removed from the text to keep consistency and improve the efficiency of the algorithm.

2) Algorithm for Content Matching: TF-IDF is a statistical measure that is used to evaluate how important a word is to a document in a collection of documents. It is a product of two statistics, term frequency (TF) and inverse document frequency (IDF). The TF component measures the number of times a word appears in the document, while the IDF component measures the rarity of the word across the entire collection of documents.

The resulting TF-IDF score for a word in a document represents the importance of that word to the document, with higher scores indicating that the word is more important to the document. By creating a numerical representation of the text using TF-IDF, words that are important to the document are given more weight and words that are not important are given less weight.

The Cosine Similarity is a measure of similarity between two non-zero vectors of an inner product space. It is calculated by taking the dot product of the vectors and dividing it

Fig. 4. TF-IDF with Cosine Similarity to compare documents

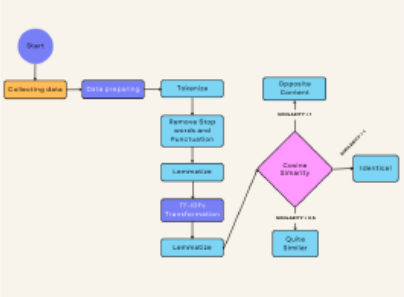

Fig. 5. Algorithm for the Content Matching Tool

by the product of their magnitudes. In the context of text analysis, the TF-IDF vectors of two documents are used to calculate the cosine similarity between the documents. The cosine similarity score ranges from -1 to 1, with 1 indicating that the documents are identical and -1 indicating that the documents are completely dissimilar.

In this way, the code uses the TF-IDF vector representation of the text and Cosine Similarity as a metric to determine the similarity between two inputted documents. The results are presented in a numerical format, which allows for easy comparison of the similarity between the two documents, as shown in Figure 4.

Then, it calculates the TF-IDF vectors for each document, which represent the relative importance of each word in the document. The code calculates the cosine similarity between the two vectors to determine the degree of overlap in content between the two documents. It prints a message indicating the similarity between the two documents based on the degree of overlap. It also identifies the common lemmas between the two documents and prints out the relevant topics for each common lemma using the WordNet library.

First, it checks if the similarity value is greater than 0. If it is, then the documents have some overlap in content. It then checks if the similarity value is equal to 1, in which case the documents are identical. If so, it prints a message indicating this. If the similarity value is not equal to 1, the code then checks if the similarity value is greater than 0.5. If so, it prints a message indicating that the documents are quite similar. Otherwise, it prints a message indicating that the documents are somewhat similar. If the similarity value is equal to 0, then the documents have no overlap in content. Finally, if the similarity value is negative, it means that the documents have opposite content. Figure 5 presents how this algorithms works

The code saves the results of the TF-IDF calculations to an Excel file for later use. Finally, the code calculates the term frequency (TF) for each word and merges it with the IDF scores. The resulting data-frame is also saved to the same Excel file.

IV. Results

As cities around the world continue to face obstacles related to population growth, urbanization, and sustainability, many are turning to smart city solutions to improve the lives of their citizens and optimize city operations. To achieve this transformation, partnerships between cities and private companies, startups, academic institutions, and other stakeholders are essential. One area where these partnerships can be particularly valuable is in the development of smart transportation systems and infrastructure such as smart buildings, streetlights, and waste management systems.

The algorithm presented helps to solve the difficulty of finding partners by using TF-IDF vector representation of text and Cosine Similarity as a metric to determine the similarity between two documents. This allows the code to compare the similarity between the needs and offers from potential partners in a numerical format, making it easier to prioritize which partnerships to analyze first. This project applies TF-IDF and Cosine Similarity to the task of matching potential partners, which is a different application than recommendation systems that use these techniques for personalized recommendations.

The tool created is capable of matching the needs with the skills and offers. However, it is important to note that even with the implementation of this automated tool, human resources teams are still crucial to find a good match for the partnership. While the tool can help filter and prioritize corporate profiles, it cannot replace the human touch needed for recruitment. A human touch is still required to ensure that the process of choosing a partner is fair, objective, and aligned with the company's or organization's values and culture.

One example of where partnerships can be particularly valuable is in the development of smart transportation systems in cities. Partnering with a technology company or startup that is specialized in autonomous vehicles can help the city implement a fleet of self-driving cars and improve the efficiency and safety of the city's transportation network.

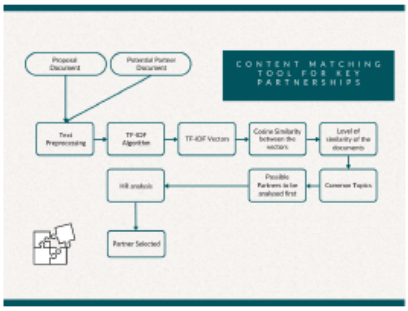

Fig. 6. Content Matching Tool

Similarly, partnering with an energy company that specializes in renewable energy can help the city develop a sustainable energy infrastructure and reduce its carbon footprint.

In addition to private companies, partnerships with academic institutions can also be valuable in developing and implementing smart city solutions. Collaborating with universities and research institutions can provide cities with access to the latest research, ideas, and innovations in the field of smart cities.

In Figure 6, it is possible to analyse how the algorithm helps to match the content between the proposal and the potential partner.

With the automated matching tool, the initial screening process can be made more efficient, allowing human resources teams to focus their attention on the most promising candidates. By using NLP techniques for content matching, the system can quickly sift through large volumes of texts from the potential partners and identify those that most closely match the project requirements. This can significantly reduce the time and effort spent on manual screening.

V. Conclusion

The implementation of an automated matching tool using AI and NLP techniques can help make the recruitment process more efficient, while also ensuring that human resources teams continue to play a crucial role in the process of making new partnerships. The tool can help filter and sort corporate profiles, allowing human resources teams to focus their attention on the most promising candidates.

As a final point, the need for cities to evolve to smart cities is becoming increasingly important, and key partnerships with private companies, startups, academic institutions, and other stakeholders are essential to achieving this transformation. By working together, cities and their partners can create innovative solutions that address the challenges of urbanization and improve the quality of life for citizens.

VI. Future Work

Although there were presented some conclusions about the performance of this tool for content matching, this study is not over.

One potential area for improvement is the development of a graphical user interface (GUI) that can be used by human resources teams to easily navigate and interact with the tool. This would make the tool more accessible and user-friendly, allowing recruiters to quickly and efficiently identify the most promising candidates for a given position.

Another potential area for future work is the integration of sentiment analysis into the tool. This could involve analyzing the sentiment of both the employer and the candidate to determine how well they might work together and what to expect from their partnership. For example, analyzing the sentiment of a company's social media posts or comments or other online activity could provide insights into their work style, which could help make more informed decisions when accepting a job or partnership.

VII. Acknowledgements

This work is supported by the European Regional Development Fund (ERDF), through the Incentive System to Research and Technological development, within the Portugal2020 Competitiveness and Internationalization Operational Program, in the framework of the project "City Catalyst Catalyst for Smart Cities" (POCI-01-0247-FEDER-046119).

This work is also funded by National Funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project Ref. UIDB/05583/2020. Furthermore, we would like to thank the Research Centre in Digital Services (CISeD) and the Polytechnic of Viseu for their support.

References

[1] "Smart cities market size, share: 2022 - 2027." [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/smart-cities-market-542.html

[2] X. Schmitt, S. Kubler, J. Robert, M. Papadakis, and Y. LeTraon, "A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2019, pp. 338–343.

[3] A. Pasat, A. Birdici, and I. Pop, "An internship campaign case study showing results of enhanced recruitment processes using nlp," in *The International Scientific Conference eLearning and Software for Education*, vol. 2. " Carol I" National Defence University, 2021, pp. 222–231.

[4] S. B. Kulkarni and X. Che, "Intelligent software tools for recruiting," *Journal of International Technology and Information Management*, vol. 28, no. 2, pp. 2–16, 2019.

[5] H. Alfawareh and S. Jusoh, "Intelligent decision support system for cv evaluation based on natural language processing," *International Journal of Advanced and Applied Sciences*, vol. 6, no. 4, pp. 1–8, 2019.

[6] A. A. Kodiyan, "An overview of ethical issues in using ai systems in hiring with a case study of amazon's ai based hiring tool," *Researchgate Preprint*, pp. 1–19, 2019.

[7] G. Yunanda, D. Nurjanah, and S. Meliana, "Recommendation system from microsoft news data using tf-idf and cosine similarity methods," *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 1, pp. 277–284, 2022.

[8] M. Ahmed, H. U. Khan, S. Iqbal, and Q. Althebyan, "Automated question answering based on improved tf-idf and cosine similarity," in *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2022, pp. 1–6.