PAPER NAME

Omolayo Clement 1-5 Full Project.docx

WORD COUNT	CHARACTER COUNT
8903 Words	51116 Characters
PAGE COUNT	FILE SIZE
48 Pages	3.2MB
SUBMISSION DATE	REPORT DATE
Jan 17, 2023 11:16 AM GMT+1	Jan 17, 2023 11:17 AM GMT+1

63% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 57% Internet database
- 17% Publications database
- Crossref database
- Crossref Posted Content database
- 46% Submitted Works database

A

PROJECT REPORT ON

AUTOMATED RESUME RANKING AND INTERVIEW SCHEDULER APPLICATION USING NATURAL LANGUAGE PROCESSING.

BY

OMOLAYO CLEMENT OLUWASHOLA
IFT/16/0156

SUPERVISED BY:
DR. O.K BOYINBODE

SUBMITTED TO:
THE DEPARTMENT OF INFORMATION TECHNOLOGY,
SCHOOL OF COMPUTING,
FEDERAL UNIVERSITY OF TECHNOLOGY AKURE,
ONDO STATE, NIGERIA.

IN PARTIAL FULFILLMENT OF REQUIREMENTS FOR THE AWARD OF BACHELOR OF TECHNOLOGY (B. TECH) IN INFORMATION TECHNOLOGY

JAN, 2023.

CERTIFICATION

This work has not been presented elsewhere for the award of any degree as for any other purpose.

…………………………………. ……......………………………

OMOLAYO CLEMENT Date

IFT/16/0156

This is to certify that this project work was carried out by **Omolayo Clement** with matriculation number **IFT/16/0156** of the Department of Information Technology, Federal University of Technology, Akure, Nigeria.

…………………………………. ……......………………………

DR. O.K BOYINBODE Date

Project Supervisor

DEDICATION

This project report is dedicated to Almighty God, my sustainer, the beginning and the end for the grace to start and end well.

ACKNOWLEDGEMENT

My profound gratitude goes to Almighty God for his divine help, support, guidance, and wisdom throughout the period of my first degree; to Him alone be the glory, honor, and adoration forever.

ABSTRACT

Recruiting job seekers to fit a particular job profile is a task crucial to most of the companies. Due to increasing growth in online recruitment, traditional hiring methods are becoming less efficient. The traditional methods typically entail a time-consuming process of manually sorting through applicants, reviewing resumes, and then producing a shortlist of suitable candidates for interviews. Job searching has improved in this technological era by becoming both more intelligent and more accessible. The vast majority of resumes and CVs that recruiters receive are not always organized. A lot of work has been put into the job search process. The selection of a candidate based solely on their resume, however, has not yet been fully automated.

Furthermore, recruitment of job seekers becomes more challenging when lots of resumes are uploaded for the same employment position. Most times, recruiters are not able to decide on which resume is the best-fit for that position.

This paper presents an effective resume ranking and interview scheduling system to help recruiters find the best jobseekers for a given position using text mining and machine learning tools. When jobseekers upload their resumes, the system ranks them according to the company's requirements for the employment positions. The ranking can be used by the recruiter to get the most preferable candidates, and it can also help schedule interviews and appointments after successful screening.

TABLE OF CONTENT

1. INTRODUCTION

2. LITERATURE REVIEW

CHAPTER ONE

1.1 BACKGROUND OF STUDY

In order to hire the right job seeker for the right position, recruiters must be able to screen resumes correctly. Resume screening is the process of determining whether a jobseeker is qualified for a role based on his or her skills, education, and other information captured on their resume. The importance of any good recruitment strategy lies in efficient and effective resume ranking. The objective of resume ranking is to locate the most qualified jobseeker for a job.In the present system, the jobseeker has to fill out every piece of information regarding their resume in a manual form, which takes a large amount of time, and then they are not satisfied with the job that the present system prefers according to their skills.

The system will act as a handshake between two entities. i.e., the recruiter who prefers the best possible jobseeker and the jobseeker who prefers the best possible job according to his or her skills and expertise. This system is an automated resume screening software using NLP and text mining. This text processing powered resume screening software goes beyond keywords and screens resumes contextually. After resume screening, the software ranks candidates based on the recruiter's job requirements automatically. This ranking is relative. The software uses natural language processing and text mining to match and rank candidates automatically.

1.2 RESEARCH MOTIVATION

The current recruitment processes are very tedious and time consuming which forces the candidates to fill out all their skills and information manually. And the recruiting team requires more manual processing to scrutinize the resumes of the candidates. That motivated the development of a solution that is less tedious and fully automated.

1.3 OBJECTIVE OF THIS RESEARCH

The primary goals of this resume ranking system are to improve upon the current resume ranking systems and make them more adaptable for both entities.

1) Jobseekers seeking jobs

2) Recruiters who are hiring the jobseekers

1.3.1 JOBSEEKERS SEEKING JOBS:

Many applicants are occasionally hired for positions that are unrelated to their skill set and abilities. This will make sure that the candidate is only hired for relevant positions.

1.3.2. THE CLIENT COMPANY, WHO IS HIRING THE CANDIDATES:

Every organization always wants to work with the best minds in their field. This system would help the organization make the best possible candidate list according to their given constraints and requirements for that particular vacancy. This kind of approach will help the hiring sector improve and make it more efficient as the relevant personnel get the relevant jobs. So there would be no regrets for both the entities, client company and the hired candidate—and Hence, satisfaction will be achieved.

1.4 METHODOLOGY

The system architecture consists of two modules:

1. The Frontend
2. The Backend Logic

1. The Frontend:
1. React Application Frontend
2. REStful API.

2.The Backend Logic
1. Parser System.
2. Candidate Skill Set Database
3. The resume ranking algorithm
4. Interview Schedler

1.5 EXPECTED CONTRIBUTION TO KNOWLEDGE

At the end of this research paper, I would have contributed a relatively improved automated resume ranking software and interview scheduler using natural language processing and text mining technologies.

1.6 ORGANIZATION OF THE PROJECT

Chapter 2 presents an overall literature review on automated resume ranking software and interview schedulers using natural language processing and text mining technologies; advantages and challenges. It also reviews the relationship between different preexisting essay grading systems, their limitations, and their applications.

Chapter three presents the methodologies used in architecting the whole resume ranking system and interview scheduler from start to finish.

Chapter four shows how the whole system is implemented using natural language processing, text mining, sendgrid, and the Django Python framework.

Chapter five consists of the conclusion and recommendation.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 RESUME

A curriculum vitae (CV), also known as a résumé or resume (or, alternatively, a resumé), is a document that a person creates and uses to highlight their educational background, professional experience, and accomplishments. Although there are many different uses for resumes, they are most frequently employed to find new jobs (Wikipedia, 2021).

A typical résumé includes a "summary" of relevant educational background and work history. The résumé is typically one of the first things a potential employer sees about a job seeker, along with a cover letter and occasionally an application for employment, and is usually used to screen applicants, frequently followed by an interview. Sehgal, M. K. (2008).

2.2 ONLINE RÉSUMÉS

As the search for employment has become more electronic, it is common for employers to only accept résumés electronically, either out of practicality or preference. This has changed much about the manner in which résumés are written, read, and processed. Some career experts are pointing out that today, a paper-based résumé is the exception rather than the rule (Garone, Liz 2014).

Many employers and hiring managers now find candidates' résumés through search engines, which makes it more important for candidates to use appropriate keywords when writing a résumé. Larger employers use applicant tracking systems to search, filter, and manage high volumes of résumés. Job ads may direct applicants to email a résumé to a company or visit its website and submit a résumé in an electronic format.

Many employers, as well as hiring companies acting on their behalf, insist on receiving resumes in a specific file format. Some only accept resumes formatted in HTML, PDF, or plain ASCII text, while others only accept Microsoft Word documents.

The use of natural language processors to parse electronic resumes is another factor to take into account. Some elements of the resume's content may be correctly interpreted by résumé parsers, but not others. The best resume parsers are very accurate when it comes to location, names, and titles but less so when it comes to skills, industries, and other less structured or quickly changing data. Résumés written in a standard format are more likely to be correctly interpreted by résumé parsers, which may make the candidate more findable.

One advantage for the employers to online résumés is the significant cost saving compared to traditional hiring methods (Retrieved 8 March 2017). Another is that potential employers no longer have to sort through massive stacks of paper. AI-tools can be used to test résumé templates. (Zwan, Gwen van der 2019).

2.3. WHAT IS A RESUME RANKING?

Resume ranking is the process of using artificial intelligence technology and machine learning tools to test resumes and suggest areas for improvement. Others charge a small fee or require a subscription, while some resume rankers are free. Users can upload an existing resume into a resume check. They might occasionally create a new resume for review using the tool.

2.4 HOW TO RANK RESUMES

A multi-step process of both automated and human resume reviews makes up resume analysis. The goal is to efficiently extract important information about job applicants from resumes. Some larger organizations may receive thousands of resumes every month. To reduce the cost of resume reviews, they scan incoming resumes into resume management software. The software then performs a keyword, employment history, educational history, and years of experience search on the scanned text. After ranking resumes according to the requirements for the position, the system displays a list of the most likely candidates to users. A human resources representative may never see those resumes that are blatantly unfit.

Once the automated resume analysis (if any) is complete, the next step is to examine the remaining resumes by hand. Since some resumes are professionally prepared, it can be difficult to ascertain from the information presented whether someone is the genuine article, or has stitched together a sketchy background into a polished presentation. Noted below are some techniques used to pluck the best resumes. (Indeed Editorial Team, May 2022).

2.5 BENEFITS OF USING A RESUME RANKER

Here are some of the benefits of using a resume ranker before submitting your resume to an employer:

2.5.1 THEY IMPROVE YOUR CHANCES OF PASSING THROUGH APPLICANT TRACKING SYSTEMS

In order for resumes to be accepted by applicant tracking systems (ATS), people should use resume checker tools. These systems are widely used by businesses to select resumes from the best applicants. Making the changes that resume checkers advise can improve the likelihood that an employer or recruiter will review your resume.

2.5.2 THEY HELP YOU LOOK MORE PROFESSIONAL

A resume check can reveal grammar and spelling mistakes that you might have missed. These mistakes can be fixed after a resume tester has pointed them out. A flawless resume will give you a more credible and intelligent appearance.

2.5.3 IT CAN HELP YOU PROMOTE YOURSELF

A resume checker does more than just point out mistakes. Many of its suggestions can help you improve parts of your testing resume that are free of errors. For instance, Resume Check might advise including statistics to describe your experience. Additionally, it can assist you in writing more succinctly and with stronger, actionable language. Your resume will stand out and appear more distinctive and appealing if you adhere to these suggestions.

2.5.4 IT CAN IMPROVE YOUR WRITTEN COMMUNICATION SKILLS

Your ability to write clearly and effectively is necessary for proofreading. For instance, you can only fix spelling mistakes when you are aware that you have done so. A resume checker flags mistakes that you might have missed. Your understanding of English will improve, and you'll become a better writer and proofreader as a result of seeing these errors highlighted. (Indeed Editorial Team. May 2022).

2.6 WHAT IS AI IN RESUME SCREENING?

Throughout the hiring process, artificial intelligence (AI) can be used to assist hiring teams in speeding up the hiring process, saving time and money, and reducing some of the undetected human bias that prevents unconventional candidates from receiving a fair evaluation.

AI tools typically target three stages of the hiring process:

- Sourcing: finding and connecting with talent quickly.
- Screening: quickly deriving the best applicants.
- Interviewing: facilitate remote hiring and save time.

AI screening tools include resume parsing, behavioral analysis, and skill evaluation.
Artificial intelligence (AI) resume screening is the process of sorting through resumes and applications to advance the most qualified candidates to the next stage of the hiring process. Artificial intelligence (AI) resume screening tools aim to speed up the laborious process of sifting through resumes to find qualified candidates.

2.7 HOW COMPANIES USE AI RESUME SCREENING TOOLS

As part of the conventional hiring process, hiring managers invest a significant amount of time in reviewing resumes. According to some reports, the time it takes to screen resumes for just one hire can reach 23 hours.

The manual process of selecting candidates was time-consuming in the past. A hiring manager would read through a stack of resumes or applications carefully in order to identify the best candidates for a skill assessment or phone interview. The hiring manager will likely have to sift through about 250 resumes for a particular position.

The rise of high-volume hiring has virtually eliminated the possibility of managing the resume screening process manually. Additionally, as more recruiters become aware of how unconscious bias affects their hiring performance, businesses are turning to machine learning and recruitment software to assist with fair candidate screening.

2.8 HOW DO RESUME SCREENING TOOLS WORK?

The information in PDF or Word files is parsed by resume screening tools using machine learning algorithms. To use the parsing tool, the applicant essentially uploads their resume. Then, using artificial intelligence, each document is scanned to extract data pertinent to the hiring manager's requirements, such as the candidate's experience, qualifications, and skills..

Resume screening tools typically fall into one of three categories:

1. **Keyword-based:** Artificial intelligence searches the text for keywords, phrases, and patterns to classify candidates using a keyword-based approach.
2. **Grammar-based:** Machine learning algorithms analyze words and phrases on the CV to determine the meaning of each sentence using a set of predefined grammatical rules.
3. **Statistical:** Numerical models analyze the data on a resume by identifying patterns like addresses, timelines, and the definition of particular words.

The most sophisticated techniques for screening resumes use statistical tools. Hiring managers have the option to create their own custom search criteria for each category to ensure that qualification indicators are set up to find the best candidates.

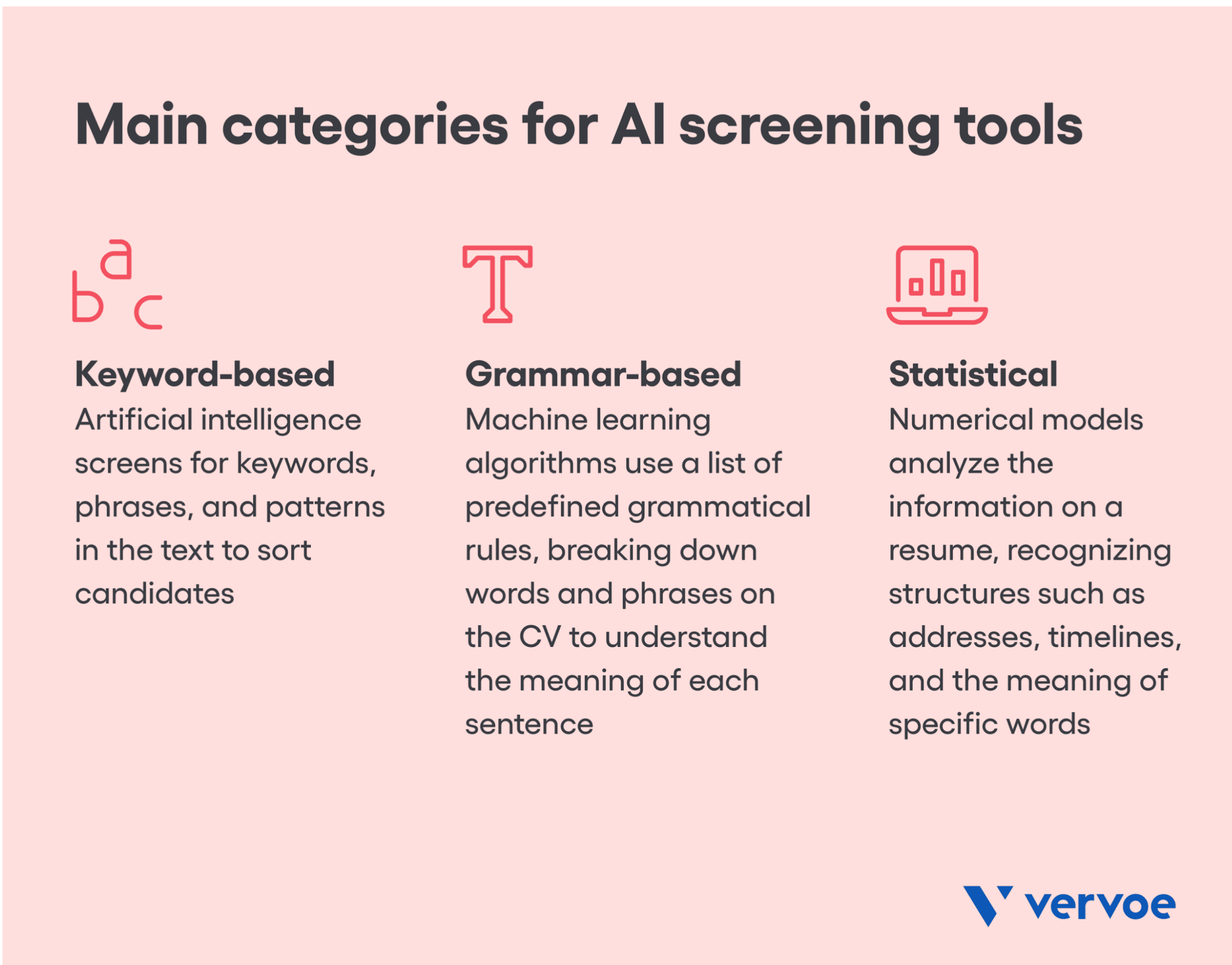

Fig 1.0 Main categories AI screening tools fall into (Emily Heaslip, 2022)

2.9 EXPECTATIONS OF USING AI TOOLS IN SCREENING IMPROVE CANDIDATE SHORTLIST

Resume screening has historically been one of the first steps in the traditional hiring process. As a result, this step often results in a candidate shortlist based on relevant information matching the job description. This shortlist then determines who is eligible for scheduling interviews, completing a skill assessment, or meeting with human recruiters.

Unfortunately, eye-tracking research indicates that recruiters review each resume for about seven seconds, or less if there are more resumes than usual. Simply put, seven seconds is not enough time to fully assess someone's capabilities, which may cause recruiters to choose candidates based on arbitrary (or biased) standards. To create a more consistent, high-quality candidate shortlist, AI screening tools use pre-set screening criteria that are consistent across all resumes.

Reduce unconscious bias.

Artificial intelligence (AI) in recruitment can be a double-edged sword. When applied ethically, artificial intelligence is a tool for good that produces potent, impartial data. It is also true, though, that bias in hiring brought about by humans in the data set can affect machine learning.

It's crucial to remove names (ethnic bias), locations (geographic or socioeconomic bias), genders (gender bias), dates of birth (age bias), and other information from resumes before screening them. This guarantees that new candidates are evaluated according to their qualifications rather than other heuristics. AI tools built into applicant tracking systems can frequently clean data in addition to parsing resumes.

2.9.1 HIRE AT SCALE

AI tools help level the playing field for small businesses to compete with larger enterprises. Merchants leveraging AI can evaluate applicants at scale, in days, not weeks, without compromising quality. These screening tools can also replace not only manually reading resumes but also phone screens, which serve virtually the same purpose as a resume review. For companies managing multiple open positions at once, this double-ended benefit can help hiring managers and recruiters spend time on the candidates most likely to continue through the application process (without screening anyone out).

2.9.2 RISKS OF USING AI TOOLS IN SCREENING

The use of AI in screening carries some risks, as was already mentioned. AI tools have the potential to duplicate some of the biases that recruiters bring to the resume review process if they are not used properly. In reality, AI systems are only as good as the data that was used to train them by the individuals who created them. Machine learning tools pick up the biases of the recruiters who initially screened those candidates when hiring teams use historical data to train them, such as data from the company's internal database.

Additionally, AI can only parse the data that is already available, so it isn't a perfect solution. Unfortunately, resumes rarely give a candidate's true abilities justice. In addition to having gender and racial biases, they also place more emphasis on style than on content. Because resume parsing tools can only evaluate a candidate based on the information presented, if a candidate omits important information or, worse yet, lies about their qualifications, the AI solution will incorrectly attribute their qualifications.

2.10 TIPS AND TRICKS FOR USING AI IN SCREENING

Although AI can help recruiters save time and effort, it does require some setup and training. Here are some pointers for incorporating AI into your resume screening procedure.

Set the right criteria

One method of implementing blind hiring is data cleaning. Another crucial step is to configure the AI screening tool to look for the appropriate criteria when scanning the resume. Most AI resume screening tools use proxies such as college degrees to measure work ethic or productivity. But, these proxies result in many well-qualified candidates being excluded from consideration.

“Workers are excluded from consideration due to variables such as the lack of a college degree or a gap in their employment history. While employers may infer that applicants who have those attributes are undeserving of consideration, applying an ‘affirmative’ logic would seem a more logical approach for seeking talent,” according to the Harvard Business School.

Ultimately, you should look for ways to screen candidates in, rather than screen them out.

2.11 DEFINE THE ROLE OF A RESUME SCREENING TOOL

A resume screening tool should only be used in certain circumstances. The initial screening of resumes is just the beginning of the hiring process and should be regarded as such. The true benefit of a resume screening system is to ensure that spam applications and those who are genuinely unqualified are removed as soon as possible, allowing recruiters to concentrate on applicants who have a chance of being hired.

Consider AI as a tool rather than a total solution as you consider your use of AI in screening. It can expedite the screening process and lessen the amount of manual work required, allowing hiring managers to spend more time personally interviewing and screening candidates.

2.12 DEFINITION NATURAL LANGUAGE PROCESSING (NLP)

The capability of a computer program to comprehend spoken and written natural language is known as "natural language processing" (NLP). It is a component of artificial intelligence (AI).

NLP has been around for more than 50 years and has linguistic roots. It has a variety of real-world applications in a number of fields, including medical research, search engines, and business intelligence.

2.12.1 HOW DOES NATURAL LANGUAGE PROCESSING WORK?

Thanks to natural language processing (NLP), computers can now comprehend natural language just like people do thanks to NLP. Natural language processing employs artificial intelligence to take real-world input, process it, and make sense of it in a way that a computer can comprehend, regardless of whether the language is spoken or written. Computers have reading programs and microphones to collect audio, just as humans have various sensors like ears to hear and eyes to see. Computers have a program to process their respective inputs, just as humans have a brain to do so. The input is eventually translated into computer-readable code during processing.

The development of algorithms and data preprocessing are the two main stages of natural language processing.

Preparing and "cleaning" text data so that computers can analyze it is known as data preprocessing. Preprocessing prepares data for use and highlights text features that an algorithm can use.

There are several ways this can be done, including:

Tokenization: This is when text is broken down into smaller units to work with.

Stop word removal. This is when common words are removed from text so unique words that offer the most information about the text remain.

Lemmatization and stemming: This is when words are reduced to their root forms to process.

Part-of-speech tagging. This is when words are marked based on the part-of speech they are -- such as nouns, verbs and adjectives.

Once the data has been preprocessed, an algorithm is developed to process it. There are many different natural language processing algorithms, but two main types are commonly used:

2.12.2 RULES-BASED SYSTEM.

The linguistic rules in this system were thoughtfully created. The use of this strategy dates back to the early stages of the development of natural language processing.

2.12.3 MACHINE LEARNING-BASED SYSTEM.

Algorithms for machine learning employ statistical techniques. They are fed training data to help them learn how to perform tasks, and as more training data is processed, they modify their techniques. Using a combination of machine learning, deep learning and neural networks, natural language processing algorithms hone their own rules through repeated processing and learning.

2.12.4 WHY IS NATURAL LANGUAGE PROCESSING IMPORTANT?

Businesses use massive quantities of unstructured, text-heavy data and need a way to efficiently process it. A lot of the information created online and stored in databases is natural human language, and until recently, businesses could not effectively analyze this data. This is where natural language processing is useful.

Consider the following two sentences: "Cloud computing insurance should be part of every service-level agreement" and "A good SLA ensures an easier night's sleep -- even in the cloud" to see the benefit of natural language processing. If a user uses natural language processing to conduct a search, the software will understand that cloud computing is a thing, that cloud is a shorthand for cloud computing, and that SLA is an acronym for service-level agreement used in the business world.

TECHNIQUES AND METHODS OF NATURAL LANGUAGE PROCESSING

Syntax and semantic analysis are two main techniques used with natural language processing.

The placement of words in a sentence to ensure proper grammar is known as syntax. NLP employs syntax to evaluate a language's meaning based on grammatical rules. Syntax techniques include:

Parsing. This is the grammatical analysis of a sentence. Example: A natural language processing algorithm is fed the sentence, "The dog barked." Parsing involves breaking this sentence into parts of speech -- i.e., dog = noun, bark = verb. This is useful for more complex downstream processing tasks.

Word segmentation: This is the process of extracting word forms from a string of text. An individual scans a handwritten document into a computer, for instance. The algorithm would be able to examine the page and identify that white spaces separate the words.

Sentence breaking: In lengthy texts, this establishes sentence boundaries. Example: A natural language processing algorithm is fed the text, "The dog barked. I woke up." The algorithm can recognize the period that splits up the sentences using sentence breaking.

Morphological segmentation: As a result, words are split up into units known as morphemes. As an illustration, the algorithm would convert the word untestably into [[un[[test]able]]ly, where "un," "test," "able," and "ly" are all recognized as morphemes. Speech recognition and machine translation both benefit greatly from this.

Stemming: This divides words with inflection in them into root forms. Example: In the sentence, "The dog barked," the algorithm would be able to recognize the root of the word "barked" is "bark." This would be useful if a user was analyzing a text for all instances of the word bark, as well as all of its conjugations. The algorithm can see that they are essentially the same word even though the letters are different.

Semantics involves the use of and meaning behind words. Natural language processing applies algorithms to understand the meaning and structure of sentences. Semantics techniques include:

Word sense disambiguation: This uses context to determine a word's meaning. Example: Think about the phrase "The pig is in the pen." There are various meanings for the word pen. This approach enables an algorithm to recognize that the word "pen" in this context refers to a fenced-in area rather than a writing tool.

Named entity recognition: This determines words that can be categorized into groups. Example: An algorithm using this method could analyze a news article and identify all mentions of a certain company or product. Using the semantics of the text, it would be able to differentiate between entities that are visually the same. For instance, in the sentence, "Daniel McDonald's son went to McDonald's and ordered a Happy Meal," the algorithm could recognize the two instances of "McDonald's" as two separate entities -- one a restaurant and one a person.

Utilizing a database to ascertain the meanings of words, natural language generation creates new text. Example: By associating specific words and phrases with aspects of the data in the BI platform, an algorithm could automatically write a summary of findings from the BI platform. Another illustration would be the automatic creation of news articles or tweets based on a specific body of training text.

Deep learning, a branch of AI that looks for and exploits patterns in data to enhance a program's comprehension, is the foundation of current approaches to natural language processing. Building this kind of big data set is one of the main challenges in natural language processing because deep learning models need enormous amounts of labeled data for the algorithm to train on and find pertinent correlations.

A more rules-based approach was used in earlier attempts at natural language processing, in which simpler machine learning algorithms were instructed what words and phrases to look for in text and were given specific responses when those words or phrases appeared. Deep learning, however, is a more adaptable and intuitive method that trains algorithms to recognize speakers' intentions from a large body of data, much like a young child would.

Three tools used commonly for natural language processing include Natural Language Toolkit (NLTK), Gensim and Intel natural language processing Architect. NLTK is an open source Python module with data sets and tutorials. Gensim is a Python library for topic modeling and document indexing. Intel NLP Architect is another Python library for deep learning topologies and techniques. (TechTargetNetwork, 2021).

2.13 REVIEW OF RELATED WORKS

There are three main extraction methods to deal with resumes in previous research, including keyword search based method,

rule-based method, and
semantic-based method.

In order to achieve the goal of job matching with a keyword search approach, it requires a different mode because it is difficult to extract the fine points of a resume. Many rule-based extraction methods provide the resume text as a web page and then extract detailed truths based on the DOM tree structure, inspired by the method of extracting the news web page. Researchers approach the task of extracting a resume as a semantic-based object extraction problem for the earlier types of methods. To predict the tags for each line's segments, some researchers use text classification techniques or sequence labeling processes. However, the majority of these techniques heavily rely on labeled data and the hierarchical structure information found in resume text. In reality, data that have been labeled and annotated by a human expert are frequently used in the learning of text extraction models. Additionally, labeling the data is more expensive the more knowledge and time it requires.

Uldis Bojars and John G. Breslin (2014) presented an RDF ontology that uses an RDF data model to model a resume. With its voluminous collection of classes and properties, resume RDF describes resume information. (Uldis Bojars 2004) further prolonged FOAF with resume information for an even more upgraded description of information. In 2002 and 2003, Turney and Littman offered a scheme which would infer the semantic orientation or evaluative character of a word from its huge hundred billion-word corpus corpora taking into thought the semantic associations with the other words, stated as paradigms by him. (Turney, P.D., Littman, M., 2002) and (Turney, P.D., Littman, M., 2003). (Ujjal Marjit et al.) proposed a different approach for recovering resume information by utilizing the concept of linked data, which enables the web to share data with various sources and discover various types of information. A deductive model that determined a match between a job seeker's talents and the skills required by the recruiter was used to develop an ontology-based approach that would match the talents of job seekers.

Kopparapu of the TCS Innovations lab created yet another system to automate the extraction of resume information. (Zhi Xiang Jing et al.) provided an online Chinese resume parser that used rule-based and statistical algorithms to extract data from a resume. (Zhi Xiang Jing et al.) provided an online Chinese resume parser that used rule-based and statistical algorithms to extract data from a resume. The largest resume parser system was created by (Zhang Chuang et al.) who worked on a resume document block analysis based on pattern matching and multi-level information identification. (Elik et al.) wanted to create a system that would turn a resume into an ontological structural model, making it easier to analyze resumes in both Turkish and English. (Di Wu et al.) succeeded to extract information from resumes more excellently by the idea of ontology using WordNet for similarity calculation. Numerous research articles were found on information extraction from resumes (Yu et al.) offered a resume information extraction methodology with cascaded hybrid model. Chandola et al. established an online resume parsing system using text analytics. (Kopparapu et al). presented a technique for automatically extracting useful data from unstructured resumes and boasted high precision and recall.

(Zhi Xiang et al.), (Zhang Chuang et al.), and (Celik Duygu et al.) have all made additional attempts to extract information from resumes. To improve the hiring process, (Mayuri Verma et al.) suggested using a Cluster based Ranking Index (CBR) to rank the resumes and find the best candidate. There haven't been many attempts to create a company recommender system in the past. As a Company Recommender, (YixinCai et al.) implemented a multinomial sorting system based on features. However, many studies on recommender systems in other areas or disciplines have been found.

(Douglas Eck, Thierry Bertin-Mahieux, and Paul Lamere) created a meta-learning algorithm-based music auto-tagging system. (Claudio Biancalana) devised a strategy for using neural networks to make movie recommendations. To produce the best products possible, (Yukun Cao and YunfengLi) planned a fuzzy based system for consumer electronics. By creating contextual data about the users and articles, (Lihong Li, Wei Chu, John Langford, and Robert E. Schapire) created a personalized news article approval system. The authors (Rafael Sotelo, Yolanda Blanco-Fernandez, Martin LópezNores, Alberto Gil-Solla, and José J. Pazos-Arias) proposed a TV program recommendation methodology that created personalized TV schedules for user groups based on TV Anytime descriptions of TV contents and semantic reasoning techniques.

According to Text Analytics, text mining is a rapidly developing field that is defined as a statistical machine learning methodology for converting unstructured raw data into structured data that can then be further classified, mined, and trained for high-quality feature information. Natural language processing (NLP), which includes pattern recognition, information extraction, data mining, and parsing techniques, is referred to as text mining (Manning, Christopher D., and HinrichSchütze, 1999). The remainder of the paper has been organized in the sequence of the steps taken during this project.

Even though there are many other websites that already exist and offer advanced services like searching by keywords, domain, location, etc., their search does not take the skill level of a particular candidate into account. For instance, if a company is looking for a candidate who can work in C, they can easily look for candidates whose resumes mention C. But how will they be able to tell whether a particular applicant is proficient in C? We incorporate additional data, such as the projects the candidate has worked on as well as the project description, into our pattern.

The candidate will provide this information as input, and we will classify the candidate into different levels of expertise by looking at the text that was used. Therefore, if a company wants to hire someone, his resume will be considered if he has a high level of C language proficiency. In order to categorize information, a knowledge base of many keywords will be planned. This will also help in the ranking of numerous resumes to convey which one is better or worse than the other, giving the applying nominees a chance to present themselves in the finest possible way.

2.14 CONCLUSION ON REVIEWS OF THESE LITERATURE

The conclusion of this literature review on resume ranking is that resume ranking is an effective way for employers to identify the best applicants for job openings. Resume ranking algorithms are able to quickly identify top candidates based on a variety of criteria, such as work experience, skills, and education. Furthermore, these algorithms are able to assess resumes more objectively and accurately than human reviewers, and can be used to streamline the recruitment process. Additionally, resume ranking algorithms can be used to identify applicants from underrepresented groups, as well as those from different cultural and economic backgrounds. Finally, resume ranking algorithms can be improved and refined to better assess the qualifications of applicants and provide employers with more accurate and useful results.

However, there are still some stones left unturned in making the recruitment process seamless and highly effective. Most resume ranking systems do not have HR aids like interview or appointment schedules for candidates who scale through the resume screening process. I will therefore be fully working on this improvement to improve automated resume ranking in the recruitment world.

CHAPTER THREE

3.0 SYSTEM METHODOLOGY AND DESIGN

3.1 SYSTEM OVERVIEW

In this chapter, the system overview, system architecture, system methodology, and the software tools used in the automated resume ranking application are discussed.

3.2 SYSTEM ARCHITECTURE

The NLP based evaluation system's architecture as shown above in Figure 3.1, consists of the

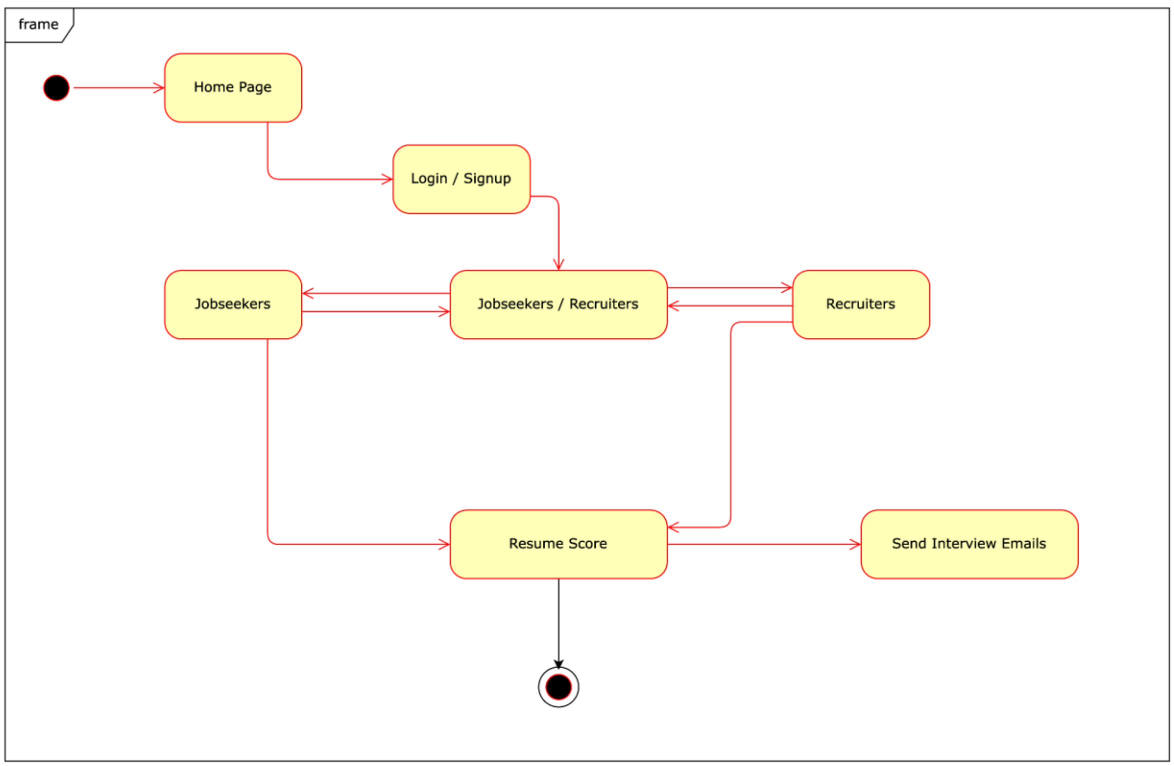

Figure 3.1 . The resume ranking state diagram..

3.3 SOFTWARE TOOLS

In the development of this project, the tools that were used include:

1. Jupyter Notebook.

2. Numpy.

3. NLTK

4. Matplotlib

3.3.1 JUPYTER NOTEBOOK

The jupyter notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. Its uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

Jupyter notebook (formerly ipython notebooks) is a web-based interactive computational environment for creating jupyter notebook documents. The term “notebook” can colloquially make reference to many different entities, mainly the jupyter web application, jupyter python web server, or jupyter document format depending on context.

According to the official website of jupyter, project jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.

Jupyter book is an open-source project for building books and documents from computational material. It allows the user to construct the content in a mixture of markdown, an extended version of markdown called myst, math & equations using mathjax, jupyter notebooks, restructuredtext, the output of running jupyter notebooks at build time. Multiple output formats can be produced (currently single files, multipage html web pages and pdf files).

Now that we have a brief understanding of the concept of project jupyter and the jupyter notebooks, and we have established that these notebooks are quite revolutionary in the modern ages, let us proceed to understand the more in-depth details of this amazing interactive environment in the next sections. (Data World, 2022).

3.3.2. NUMPY

Numpy in python is a library that is used to work with arrays and was created in 2005 by travis oliphant. Numpy library in python has functions for working in domain of fourier transform, linear algebra, and matrices. Python numpy is an open-source project that can be used freely. Numpy stands for numerical python. Operations using numpy. Using numpy, a developer can perform the following operations. Mathematical and logical operations on arrays.

Fourier transforms and routines for shape manipulation. Operations related to linear algebra. Numpy has in-built functions for linear algebra and random number generation.(Sunil Sharanappa. 2022).

3.3.2.1 WHY USE NUMPY ?

In Python we have lists that serve the purpose of arrays, but they are slow to process.NumPy aims to provide an array object that is up to 50x faster that traditional Python lists.
The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
Arrays are very frequently used in data science, where speed and resources are very important.
Numpy's most useful feature is the n dimension array object (Nd array).(Sunil Sharanappa. 2022)

3.3.2.2 MAIN BENEFITS OF NUMPY ARRAY ARE:

1) Less Memory
2) Fast
3) Convenient

3.3 NATURAL LANGUAGE TOOLKIT (NLTK)

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

Natural Language Processing with Python provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been updated for Python 3 and NLTK 3. (The original Python 2 version is still available at https://www.nltk.org/book_1ed.)

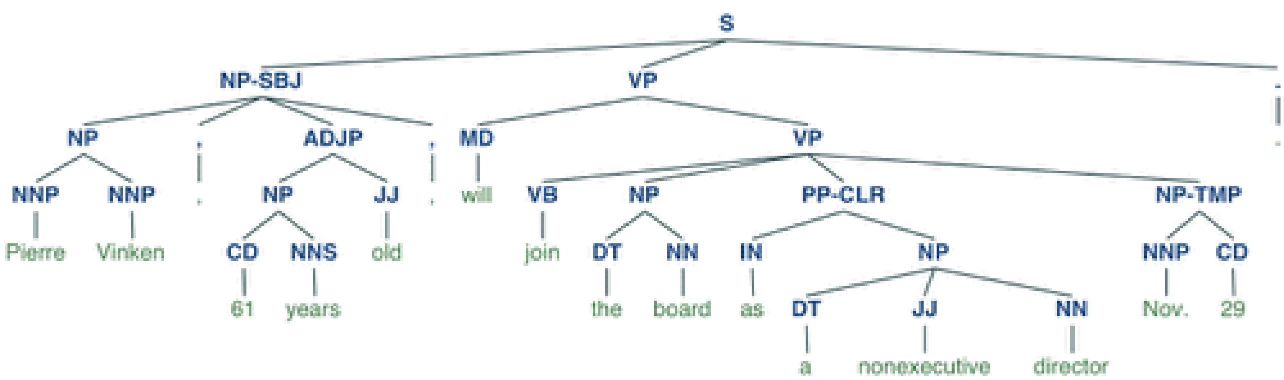

Fig 3.2 Parse tree generated with NLTK.(wikipedia 2022)

3.4 MATH PLOT LIB

Since python is widely used in machine learning, resources like numpy and matplotlib are often useful in modeling machine learning technologies. The idea is that programmers access these libraries for key tasks inside of a broader python environment, and integrate the results with all of the other elements and features of a machine learning program, a neural network or some other advanced machine. The utility of numpy and matplotlib have to do with numbers — the utility of matplotlib specifically has to do with visual plotting tools. So in a sense, these resources are more analytical than generative. However, all of this infrastructure works together to allow the machine learning programs to produce results that are useful to human handlers.

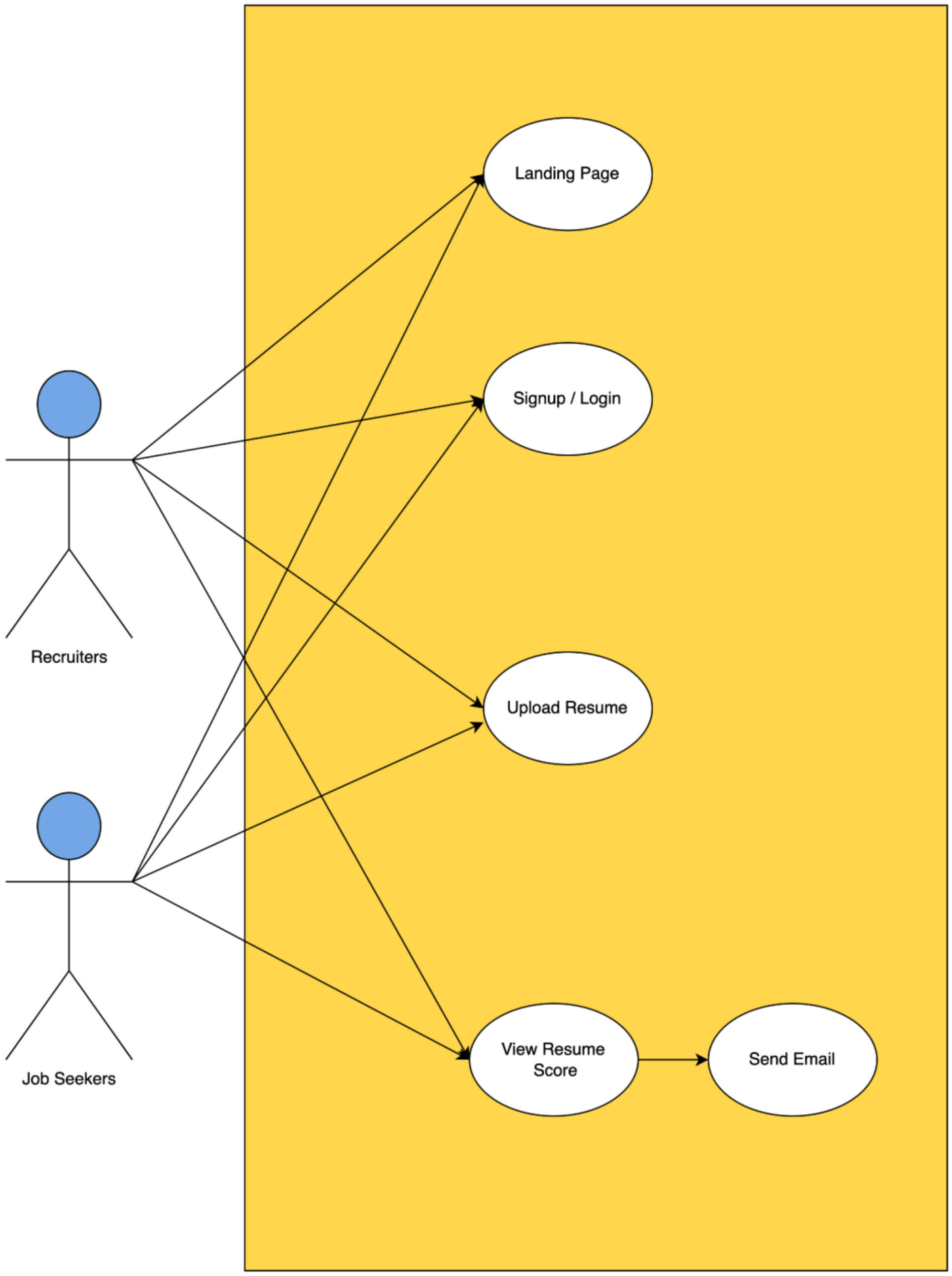

Fig. 3.3.
UML Use Case Diagram of Resume Ranking system

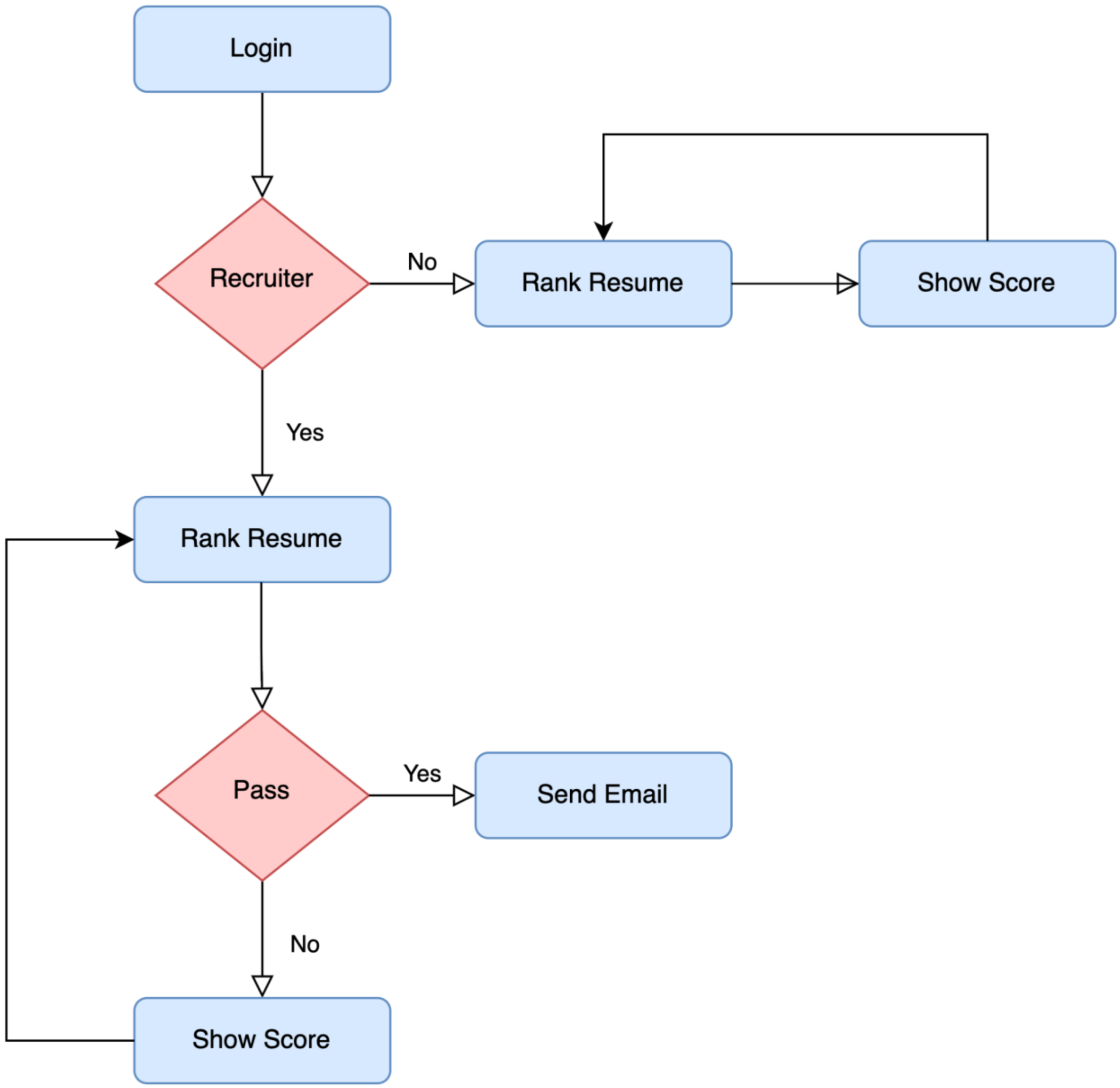

Fig. 3.4. UML Use Case Diagram of Resume Ranking system

CHAPTER FOUR

4.0 IMPLEMENTATION AND RESULT

4.1. INTRODUCTION

In this section, the code implementation of the automated resume ranking system using text mining, and natural language processing technologies used for analysis and automation were considered. The implementation of the code for the web interface and the connecting Application Programming Interface (API) for the models is also considered and discussed. Lastly, the results obtained from using the automated system are presented using the web interface and an attempt is made to analyze the result.

4.1.1 THE MINIMUM HARDWARE REQUIREMENTS OF THE SYSTEM ARE:

a. 64-bit processors
b. 4gb RAM
c. 128Gb Hard Drive

4.1.2 THE MINIMUM SOFTWARE REQUIREMENTS OF THE SYSTEM ARE:

a. Any Operating System
b. Web browser
c. Apache Server

4.2 CODE IMPLEMENTATION

4.2.1 APPROACH:

I want to create a Python program that will return the percentage % match between a resume and a job description. Also, I will create a word cloud using the Job description so that we get a clear view of all the important keywords.

Install & Import Libraries:

First, we are going to import all the libraries required for this project.

Now, Resumes do not have a fixed file format, and hence they can be in any file format such as .pdf or .doc, or .docx. So our first challenge is to read the resume and convert it to plain text. For this, we can use two Python modules: pdfminer and doc2text. These modules help extract text from.pdf or .doc, or .docx file formats.

```
pip install pdfminer
pip install docx2txt
```

Let’s import all the libraries required for this project.

```
import io
from pdfminer.converter import TextConverter
from pdfminer.pdfinterp import PDFPageInterpreter
from pdfminer.pdfinterp import PDFResourceManager
from pdfminer.pdfpage import PDFPage
#Docx resume
import docx2txt
#Wordcloud
import re
import operator
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
set(stopwords.words('english'))
from wordcloud import WordCloud
from nltk.probability import FreqDist
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

Reading the Resume:

Here, I will create two different functions. One to read resumes in pdf file format. Another one to read in .doc format. Both of the functions will return the text in the Resume.

Read PDF Resume:

```
def read_pdf_resume(pdf_doc):
    resource_manager = PDFResourceManager()
    fake_file_handle = io.StringIO()
```

```
    converter = TextConverter(resource_manager, fake_file_handle)
    page_interpreter = PDFPageInterpreter(resource_manager, converter)
    with open(pdf_doc, 'rb') as fh:
        for page in PDFPage.get_pages(fh, caching=True,check_extractable=True):
            page_interpreter.process_page(page)

        text = fake_file_handle.getvalue()
   # close open handles
    converter.close()
    fake_file_handle.close()
    if text:
        return text
```

Read word Resume:

```
def read_word_resume(word_doc):
     resume = docx2txt.process(word_doc)
     resume = str(resume)
     #print(resume)
     text =  ''.join(resume)
     text = text.replace("\n", "")
    if text:
         return text
```

Create a Word Cloud with Keywords

How about a graphical image which will display the keywords in the Job Description? I am always a big fan of Word Clouds. If you are scanning a job description you may miss a few skills that the role demands. Maybe you have some experience in those skills and did not remember to add in your resume. Thus, a word cloud will flash those keywords for a quick review.

Clean the Job Description:

To create a word cloud I usually clean the text first to avoid word repetitions or punctuations or numbers because those don't make much sense in a word cloud.

```
def clean_job_decsription(jd):
    ''' a function to create a word cloud based on the input text parameter'''
    ## Clean the Text
    # Lower
    clean_jd = jd.lower()
    # remove punctuation
    clean_jd = re.sub(r'[^\w\s]', '', clean_jd)
    # remove trailing spaces
    clean_jd = clean_jd.strip()
    # remove numbers
    clean_jd = re.sub('[0-9]+', '', clean_jd)
    # tokenize
    clean_jd = word_tokenize(clean_jd)
    # remove stop words
    stop = stopwords.words('english')
    clean_jd = [w for w in clean_jd if not w in stop]


   return(clean_jd)
```

Create a word cloud:

Now, it’s time to create the image.

```
def create_word_cloud(jd):
    corpus = jd
    fdist = FreqDist(corpus)
    #print(fdist.most_common(100))
    words = ' '.join(corpus)
    words = words.split()
```

```
# create a empty dictionary
data = dict()
#  Get frequency for each words where word is the key and the count is the value
for word in (words):
    word = word.lower()
    data[word] = data.get(word, 0) + 1
# Sort the dictionary in reverse order to print first the most used terms
dict(sorted(data.items(), key=operator.itemgetter(1),reverse=True))
word_cloud = WordCloud(width = 800, height = 800,
background_color ='white',max_words = 500)
word_cloud.generate_from_frequencies(data)

# plot the WordCloud image
plt.figure(figsize = (10, 8), edgecolor = 'k')
plt.imshow(word_cloud,interpolation = 'bilinear')  plt.axis("off")  plt.tight_layout(pad = 0)
plt.show()
```

Get Job Description and Resume Match Score

Now, we are at the final part of our project. To get a score of how the resume matches a specific job description, I am going to use a *Cosine Similarity* metric. Mathematically, it measures the cosine of the angle between two vectors projected in a multi-dimensional space. The smaller the angle, the higher the cosine similarity. In this context, the two vectors are arrays containing the words of two documents.

Now, a commonly used approach to matching similar documents is based on counting the maximum number of common words between the documents. But there is a problem with this approach. As the size of the document increases, the number of common words tend to increase even if the documents talk about different topics.

The cosine similarity is advantageous because even if the two similar documents are far apart by the Euclidean distance because of the size (like, the word ‘python’ appeared 50 times in one document and 2 times in another) they could still have a smaller angle between them. Thus, smaller the angle, higher the similarity.

Okay, so lets create a function to find the match score!

```
def get_resume_score(text):
    cv = CountVectorizer(stop_words='english')
    count_matrix = cv.fit_transform(text)
    #Print the similarity scores
    print("\nSimilarity Scores:")

    #get the match percentage
    matchPercentage = cosine_similarity(count_matrix)[0][1] * 100
    matchPercentage = round(matchPercentage, 2) # round to two decimal

    print("Your resume matches about "+ str(matchPercentage)+ "% of the job
        description.")
```

Test Resume Scanner:

Finally, it is time to get a score! I am using my personal resume and copied it in the same folder so that it can be read by this program. Now, let me get a Job Description from a Job portal. I took a Data Analyst Job Description and let's see how well my profile matches with this specific role.

Let's run all the functions created above and get a score!

```
if name == 'main':
    extn = input("Enter File Extension: ")
    #print(extn)
    if extn == "pdf":
        resume = read_pdf_resume('Resume_OindrilaSen.pdf')
    else:
        resume = read_word_resume('test_resume.docx')

    job_description = input("\nEnter the Job Description: ")
    ## Get a Keywords Cloud
    clean_jd = clean_job_decsription(job_description)
```

```
create_word_cloud(clean_jd) text = [resume, job_description]

## Get a Match score
get_resume_score(text)
```

My Goodness!

Similarity Scores:

Your resume matches about 26.82% of the job description.

4.4 FRONTEND IMPLEMENTATION

1

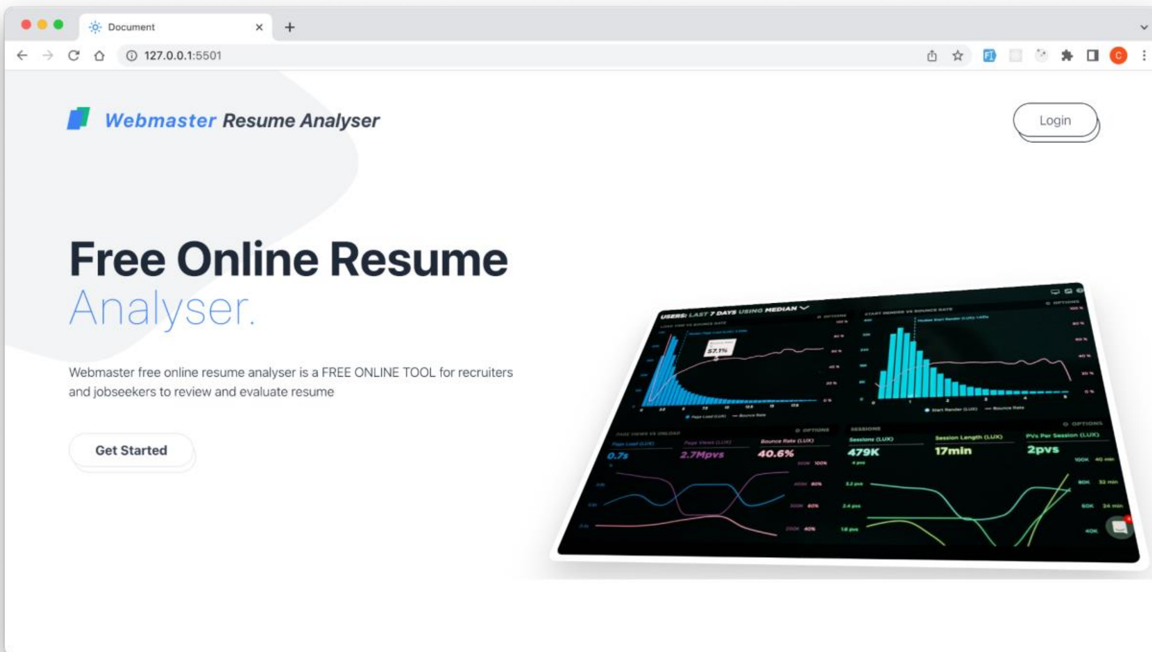

Fig 4.0 Resume analyser landing page.

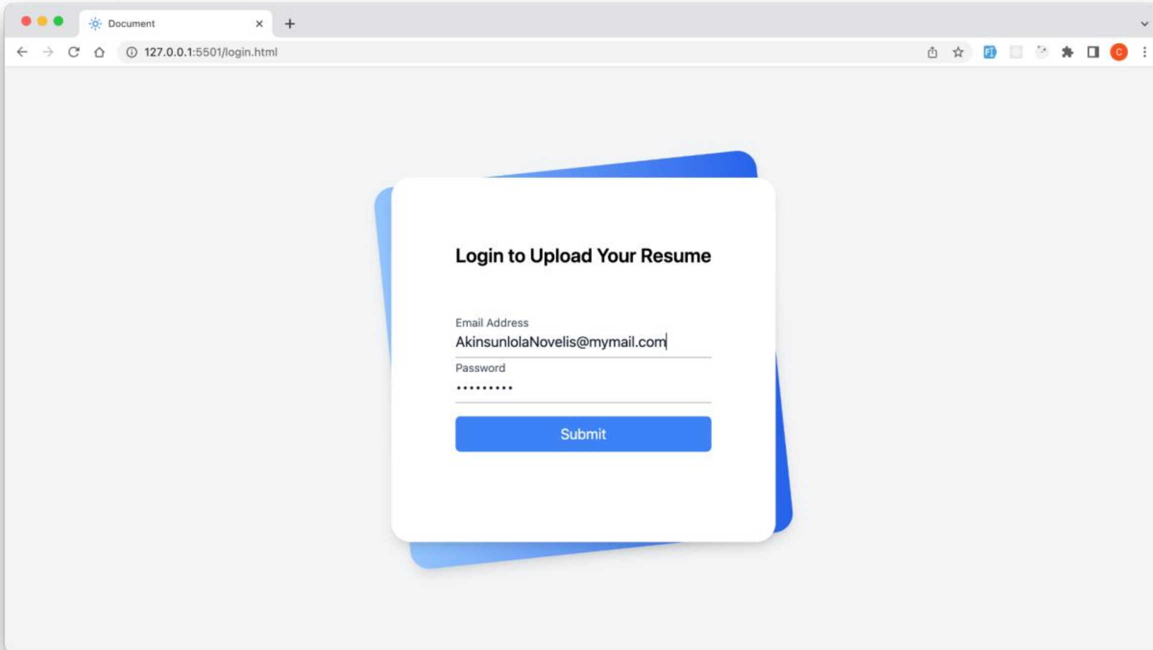

Fig 4.1 Resume analyser login page.

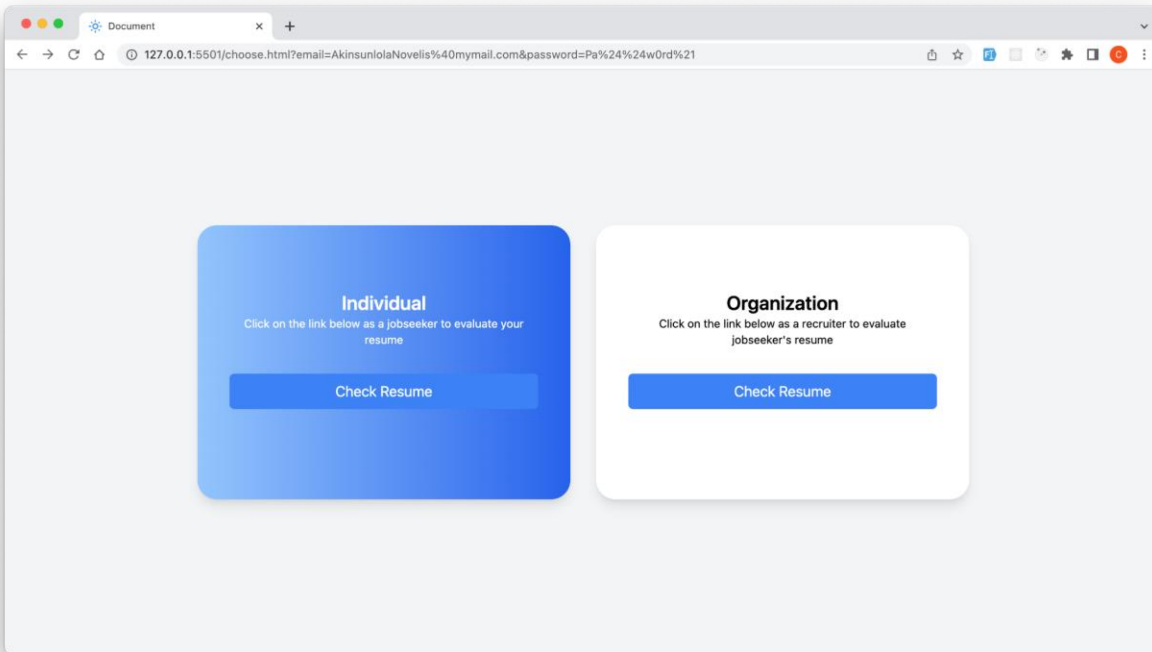

Fig 4.2 Resume analyser Individual and organization's page.

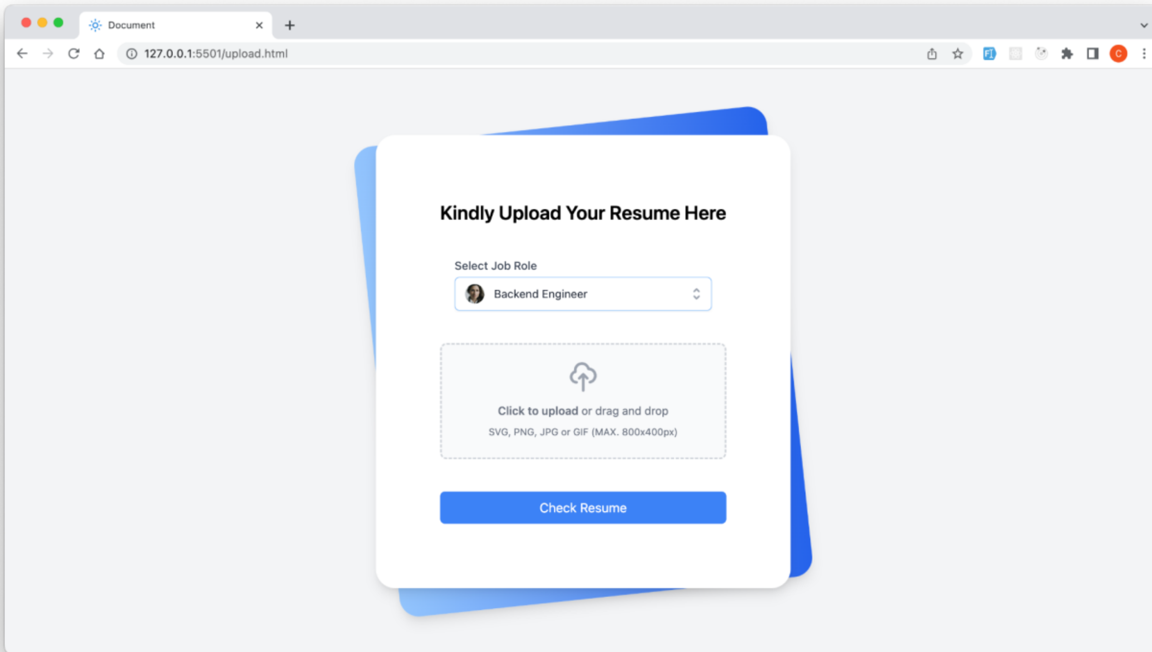

Fig 4.3 Resume analyser upload page.

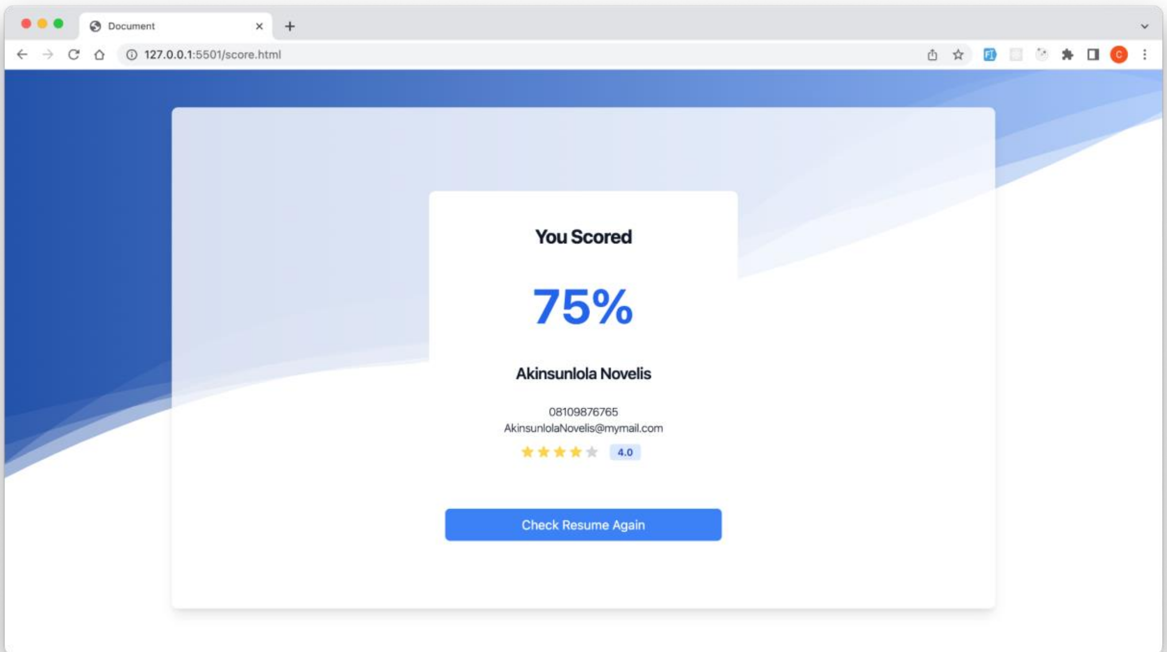

Fig 4.4 Resume analyser final score page.

55

CHAPTER FIVE

5.0 CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

5.2 RECOMMENDATION

REFERENCES

Emily Heaslip. (2022). AI In Resume Screening: Expectations vs. Reality. https://vervoe.com/ai-in-resume-screening, 62(2), 127-142.

Chapter 2

“Resume Definition,” Wikipedia, Wikimedia Foundation, October 11, 2021, en.wikipedia.org/wiki/Résumé.

"Resume | Define Resume at Dictionary.com." Dictionary.reference.com. Retrieved 8 March 2017.

Sehgal, M. K. (2008). Business Communication. Excel Books. p. 392. ISBN 9788174465016.
Garone, Liz (23 June 2014). "To Print Or Not To Print". bbc.com. Retrieved 25 June 2014.

"Career Advice, Find a Job & Salary Trends - Wall Street Journal". Careerjournal.com. 3 October 2002. Retrieved 8 March 2017.

Zwan, Gwen van der (11 June 2019). "Robots reviewed my resume and they were not impressed". TNW | Work2030. Retrieved 12 June 2021.

Indeed Editorial Team (May 2022). What Is a Resume Check? (And How to Use These Tools). Retreived from au.indeed.com/career-advice/resumzxes-cover-letters/what-is-resume-check

TechTarget Network, Search Enterprise AI , Natural Language Processing (NLP)”, March 2021, techtarget.com/searchenterpriseai/definition/natural-language-processing-NLP.

Uldis Bojars, John G. Breslin, "ResumeRDF: Expressing Skill Information on the Semantic Web".

Uldis Bojars, “ Extending FOAF with Resume Information” .

Turney, P.D., Littman, M.: Unsupervised learning of semantic orientation from a hundred billion-word corpus. Technical Report ERC-1094 (NRC 44929), National Research Council of Canada (2002)

Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association, ACM Transactions on Information Systems (TOIS), vol. 21, no. 4, pp. 315--346 (2003)

Ujjal Marjit, Kumar Sharma and Utpal Biswas, " Discovering Resume Information Using Linked Data" , in International Journal of Web & Semantic Technology, Vol.3, No.2, April 2012.

Maryam Fazel-Zarandi1, Mark S. Fox2, " Semantic Matchmaking for Job Recruitment: An Ontology-Based Hybrid Approach" , International Journal of Computer Applications (IJCA), 2013

Kopparapu S.K, " Automatic Extraction of Usable Information from Unstructured Resumes to aid search" , IEEE International Conference on Progress in Informatics and Computing (PIC), Dec 2010.

Zhi Xiang Jiang, Chuang Zhang, Bo Xiao, Zhiqing Lin, " Research and Implementation of Intelligent Chinese Resume Parsing" , WRI International Conference on Communications and Mobile Computing, Jan 2009.

Zhang Chuang, Wu Ming, Li Chun Guang, Xiao Bo, " Resume Parser: Semi-structured Chinese Document Analysis" , WRI World Congress on Computer Science and Information Engineering, April 2009.

Celik Duygu, Karakas Askyn, Bal Gulsen, Gultunca Cem, " Towards an Information Extraction System Based on Ontology to Match Resumes and Jobs" , IEEE 37th Annual Workshops on Computer Software and Applications Conference Workshops, July 2013.

Di Wu, Lanlan Wu, Tieli Sun, Yingjie Jiang, “ Ontology based information extraction technology” , International Conference on Internet Technology and Applications (iTAP), Aug 2011.

Yu, Kun, Gang Guan, and Ming Zhou. "Resume information extraction with cascaded hybrid model." Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005.

Chandola, Divyanshu, et al. "ONLINE RESUME PARSING SYSTEM USING TEXT ANALYTICS."

Kopparapu, Sunil Kumar. "Automatic extraction of usable information from unstructured resumes to aid search." Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on. Vol. 1. IEEE, 2010.

Zhi Xiang Jiang, Chuang Zhang, Bo Xiao, Zhiqing Lin, “ Research and Implementation of Intelligent Chinese Resume Parsing” , WRI International Conference on Communications and Mobile Computing, Jan 2009.

Zhang Chuang, Wu Ming, Li Chun Guang, Xiao Bo, “ Resume Parser: Semi-structured Chinese Document Analysis” , WRI World Congress on Computer Science and Information Engineering, April 2009.

CelikDuygu, Karakas Askyn, Bal Gulsen, GultuncaCem, “ Towards an Information Extraction System Based on Ontology to Match Resumes and Jobs” , IEEE 37th Annual Workshops on Computer Software and Applications Conference Workshops, July 2013.

Mayuri Verma, “ Cluster based Ranking Index for Enhancing Recruitment Process using Text Mining and Machine Learning” , International Journal of Computer Applications (0975 – 8887) Volume 157 – No 9, January 2017.

YixinCai, Ruixi Lin, Yue Kang, “ A Personalized Company Recommender System for Job Seekers” . [17] Douglas Eck, Thierry Bertin-Mahieux, Paul Lamere, and Stephen Green, Automatic generation of social tags for music recommendation. Neural Information Processing Systems Conference (NIPS) 20. Vancouver, British.

4 Douglas Eck, Thierry Bertin-Mahieux, Paul Lamere, and Stephen Green, Automatic generation of social tags for music recommendation. Neural Information Processing Systems Conference (NIPS) 20. Vancouver, British.

Claudio Biancalana, Fabio Gasparetti, Alessandro Micarelli, Alfonso Miola, and Giuseppe Sansonetti, Contextaware movie recommendation based on signal processing and machine learning, Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation, p.5- 10, October 23-27, 2011, Chicago, Illinois.

Yukun Cao, and Yunfeng Li, An intelligent fuzzy-based recommendation system for consumer electronic products, Expert Systems with Applications: An International Journal, v.33 n.1, p.230-240, July, 2017.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In Proceedings of the Nineteenth International Conference on World Wide Web, 2010.

Rafael Sotelo, Yolanda Blanco-Fernández, Martín López-Nores, Alberto Gil-Solla, and José J. PazosArias, “ TV program recommendation for groups based on muldimensional TV-anytime classifications,” IEEE Trans. Consum. Electron., vol. 55, no. 1, pp. 248-256, Feb. 2009.

Feldman, Ronen, and James Sanger. The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge University Press, 2007.

Manning, Christopher D., and HinrichSchütze. Foundations of statistical natural language processing. Vol. 999. Cambridge: MIT press, 1999.

CHAPTER 3

DataWorld. Manipulate and analyze data with this visual interface for Python and other languages. https://data.world/integrations/jupyter. 2022

Sunil Sharanappa. Basics Of Numpy. https://medium.com/@sunils0506/basics-of-numpy-22f8225670bd

Bird, Steven, Edward Loper and Ewan Klein (2019), Natural Language Processing with Python. O'Reilly Media Inc.

63% Overall Similarity

Top sources found in the following databases:

- 57% Internet database
- 17% Publications database
- Crossref database
- Crossref Posted Content database
- 46% Submitted Works database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	jespublication.com Internet	9%
2	searchenterpriseai.techtarget.com Internet	8%
3	vervoe.com Internet	7%
4	ijetmas.com Internet	3%
5	en.wikipedia.org Internet	3%
6	towardsdatascience.com Internet	2%
7	nltk.org Internet	2%
8	ijraset.com Internet	2%

9	Indian School of Business on 2022-08-13 Submitted works	2%
10	Monash University on 2022-09-16 Submitted works	1%
11	techopedia.com Internet	1%
12	cyberlearning.org Internet	1%
13	juniperpublishers.com Internet	1%
14	medium.com Internet	1%
15	accountingtools.com Internet	1%
16	Sim University on 2022-03-24 Submitted works	<1%
17	core.ac.uk Internet	<1%
18	omkarpathak.in Internet	<1%
19	CSU, San Jose State University on 2022-11-09 Submitted works	<1%
20	University of London External System on 2022-01-02 Submitted works	<1%

21	github.com Internet	<1%
22	Jose Rizal University on 2022-10-28 Submitted works	<1%
23	University of Limerick on 2022-08-23 Submitted works	<1%
24	blog.edugrad.com Internet	<1%
25	oindrilasen.com Internet	<1%
26	jmdet.com Internet	<1%
27	Ahsanullah University of Science and Technology on 2022-11-25 Submitted works	<1%
28	University Malaysia of Computer Science & Engineering on 2022-08-18 Submitted works	<1%
29	ijarcce.com Internet	<1%
30	Vardhaman College of Engineering, Hyderabad on 2022-05-10 Submitted works	<1%
31	Manipal University on 2022-12-29 Submitted works	<1%
32	University College London on 2019-09-25 Submitted works	<1%

33	University of Bedfordshire on 2022-12-15 Submitted works	<1%
34	slideshare.net Internet	<1%
35	coursehero.com Internet	<1%
36	University of South Australia on 2022-05-31 Submitted works	<1%
37	thequickadvisor.com Internet	<1%
38	amrita.edu Internet	<1%
39	Takshak Desai, Udit Deshmukh, Mihir Gandhi, Lakshmi Kurup. "A Hybri... Crossref	<1%
40	hindawi.com Internet	<1%
41	projectreserve.com Internet	<1%
42	University of Birmingham on 2022-09-20 Submitted works	<1%
43	publishing.logos-science.com Internet	<1%
44	Jose Rizal University on 2022-08-25 Submitted works	<1%

45	University of Sunderland on 2023-01-13 Submitted works	<1%
46	docplayer.net Internet	<1%
47	Arts, Sciences & Technology University In Lebanon on 2022-03-03 Submitted works	<1%
48	Letterkenny Institute of Technology on 2023-01-06 Submitted works	<1%
49	MCC Training Institute on 2011-05-22 Submitted works	<1%
50	University College London on 2020-01-12 Submitted works	<1%
51	Monash University on 2022-09-16 Submitted works	<1%
52	University of Maryland, Global Campus on 2022-11-16 Submitted works	<1%
53	University of South Florida on 2022-07-13 Submitted works	<1%
54	bticino.be Internet	<1%
55	pt.scribd.com Internet	<1%
56	sepidehdoost.com Internet	<1%

57	Obafemi Awolowo University, Ile-Ife on 2022-03-19 Submitted works	<1%
58	University of Westminster on 2022-07-25 Submitted works	<1%
59	lakhasly.com Internet	<1%
60	87994.com Internet	<1%
61	CITY College, Affiliated Institute of the University of Sheffield on 2015-... Submitted works	<1%
62	Southern New Hampshire University - Continuing Education on 2022-0... Submitted works	<1%
63	Yves Bestgen, Nadja Vincze. "Checking and bootstrapping lexical norm... Crossref	<1%
64	opus-hshn.bsz-bw.de Internet	<1%
65	repository.out.ac.tz Internet	<1%
66	repository.futminna.edu.ng:8080 Internet	<1%
67	Indiana University on 2019-12-19 Submitted works	<1%
68	Vels University on 2017-04-18 Submitted works	<1%

69	gitlab.sliit.lk Internet	<1%
70	samphina.com.ng Internet	<1%
71	Universiti Tunku Abdul Rahman on 2022-04-14 Submitted works	<1%
72	University of Wales Institute, Cardiff on 2016-04-19 Submitted works	<1%
73	Akshay Kulkarni, Adarsha Shivananda, Anoosh Kulkarni. "Natural Lang... Crossref	<1%
74	Akshay Kulkarni, Adarsha Shivananda. "Natural Language Processing ... Crossref	<1%
75	Asia Pacific Instutute of Information Technology on 2019-07-26 Submitted works	<1%
76	Indiana University on 2021-05-07 Submitted works	<1%
77	Liverpool John Moores University on 2021-10-26 Submitted works	<1%
78	The Robert Gordon University on 2020-05-07 Submitted works	<1%
79	ajumosepublictransport.blogspot.com Internet	<1%
80	repository.seeu.edu.mk Internet	<1%

81	swarm.workshop.perforce.com Internet	<1%
82	Federal University of Technology on 2012-10-23 Submitted works	<1%
83	Manipal University on 2022-10-01 Submitted works	<1%
84	Rutgers University, New Brunswick on 2022-12-05 Submitted works	<1%
85	University of Melbourne on 2022-02-02 Submitted works	<1%
86	University of the Pacific on 2021-01-25 Submitted works	<1%