DEVELOPING A FRAMEWORK TO IDENTIFY PROFESSIONAL SKILLS REQUIRED FOR BANKING SECTOR EMPLOYEE IN UK USING NATURAL LANGUAGE PROCESSING (NLP) TECHNIQUES

Submitted in partial fulfilment of the requirements of
the degree of Master of Philosophy
May 2024

Author: Gayanika Shiromala Anthony
Supervisor I: Professor M. Saraee
Supervisor II: Dr Kaveh Kiani

School of Science, Engineering and Environment (SEE)
The University of Salford, Manchester, United Kingdom

Table of Content

List of Tables

List of Figures

List of Abbreviations

ABS	-	Abstractive
AEC	-	Architectural, Engineering, and Construction
AI	-	Artificial Intelligence
BERT	-	Bidirectional Encoder Representations from Transformers
CNN	-	Convolutional Neural Network
CRC	-	Clinical Research Coordinator
DAST	-	De-Identification and Anonymization for Sharing Medical Texts
DT	-	Decision Tree
EXT	-	Extractive
FAHP	-	Fuzzy Analytical Hierarchical Process
GBM	-	Gradient Boosting Machine
GPT	-	Generative Pretrained Transformer
HDLDA	-	Hierarchical Dual Latent Dirichlet Allocation
HMM	-	Hidden Markov Model
HR	-	Human Resource
HRIS	-	Human Resources Information Systems
HRIS	-	Human Resource Information Systems
HRM	-	Human Resources Management
ICT	-	Information and Communication Technology
IDF	-	Inverse Document Frequency
JDs	-	Job Descriptions
KNN	-	K-Nearest Neighbour
LDA	-	Latent Dirichlet Allocation
LDA	-	Latent Dirichlet Allocation
LR	-	Logistic Regression
LSTM	-	Long Short-Term Memory
ME	-	Maximum Entropy
ML	-	Machine Learning
MLM	-	Masked Language Modelling
MNB	-	Multinomial Naive Bayes

MUC-6	-	Sixth Message Understanding Conference
NB	-	Nave Bayes
NBC	-	Naive Bayes classifier
NER	-	Named Entity Recognition
NERC	-	Named Entity Recognition and Classification
NLG	-	Natural Language Generation
NLP	-	Natural Language Processing
NLTK	-	Natural Language Toolkit
NMF	-	Non-negative Matrix Factorization
NN	-	Neural Network
NSP	-	Next Sentence Prediction
OvR	-	One-vs-Rest
PA	-	Performance Appraisal
POS	-	Part of Speech
PSD	-	Phrase Sense Disambiguation
RNNs	-	Recurrent Neural Networks
SLR	-	Systematic Literature Review
SOC	-	Standard Occupational Classification
STEM	-	Science, Technology, Engineering, and Mathematics
SVM	-	Support Vector Machine
TDAP	-	Transformations and Discourse Analysis Project
TF	-	Term Frequency
TF-IDF	-	Term Frequency-Inverse Document Frequency
T-HRIS	-	Tactical Human Resource Information Systems
W2V	-	Word2Vec
WSD	-	Word Sense Disambiguation

Declaration

I, Gayanika Shiromala Anthony, declare that this dissertation titled " Developing a framework to identify professional skills required for banking sector employee in UK using Natural Language Processing (NLP) Techniques" and the work presented within are entirely my own. No part of this dissertation has been submitted in support of any application for any other degree or qualification at the University of Salford or any other educational institution.

I affirm that I have maintained professional integrity throughout all aspects of my research degree, adhering to the Institutional Code of Practice and the Regulations for Postgraduate Research Degrees.

Whenever I have consulted the published work of others, due credit is clearly attributed. Quotations from the work of others are appropriately cited, providing the necessary source information. This research project received no external funding, and I declare no conflict of interest.

Acknowledgement

First and foremost, I would like to express my heartfelt gratitude to Professor Mohamad Saraee, my dedicated supervisor, for his precious guidance, insightful feedback, and continuous support throughout the research process. Special thanks are extended to Dr Kaveh Kiani, whose expertise and encouragement significantly contributed to the success of this project.

I would also want to express my gratitude to the rest of the staff and team at The University of Salford's School of Science, Engineering, and Environment for their insightful remarks and encouragement throughout this research.

I extend my thanks to banking sector professionals who have provided their valuable feedback for my research outcome. I am also grateful to my friends who helped and provided constructive feedback. Without their kind corporation this would not have been completed.

Finally, I extend my deepest gratitude to my husband and two children for their unwavering encouragement and understanding during this academic journey. Their support has been instrumental in the completion of this research.

Abstract

The banking sector is changing dramatically, and new studies reveal that many financial institutions are having challenges keeping up with technology advancements and an acute shortage of skilled workers. The banking industry is changing into a dynamic field where success requires a wide range of talents. For the industry to properly analyses, match, and develop personnel, a strong skill identification process is needed. The objective of this research is to establish a framework for determining the competencies needed by banking industry experts through data extraction from job postings on UK websites.

Data is extracted from job vacancy websites leveraging web-based annotation tools and Natural Language Processing (NLP) techniques. This study starts by conducting a thorough examination of the literature to investigate the theoretical underpinnings of NLP techniques, its applications in talent management and human resources within the banking industry, and its potential for skill identification. Next, textual data from job ads is processed using NLP techniques to extract and categorize talents unique to these categories. Advanced algorithms and approaches are used in the NLP-based development process to automatically extract skills from unstructured textual material, guaranteeing that the skills gathered are accurate and most relevant to the needs of the banking industry. To make sure the NLP techniques-driven skill identification is accurate and up to date, the extracted skills are verified by expert feedback.

In the final phase, machine learning models are employed to predict the skills required for banking sector employees. This study delves into various machine learning techniques, which are implemented within the framework. By preprocessing and training on skills extracted from job advertisements, these models undergo evaluation to assess their effectiveness in skill prediction. The results offer a detailed analysis of each model's performance, with metrics such as recall, precision, and F1-score being used for assessment. This comprehensive examination underscores the potential of machine learning in skill identification and highlights its relevance in the banking sector.

Key Words: Machine Learning, Banking Sector, Employability, Data Mining, NLP, Semantic analysis, Skill assessment, Skill Recognition, Talent management

Chapter 1: Introduction

The Global City Report (2022) highlighted that the financial services industry powers the UK economy, making it one of the top financial hubs in the world. The financial services industry powers the economy of the United Kingdom. 1.1 million jobs in financial services and slightly over 1.2 million in related professional services make up the sector's total employment of over 2.3 million (ONS,2022). According to the IMF, the UK has one of the most developed financial markets in the world, offering efficiency, depth, and straightforward access to capital (IMF, 2022).

The banking industry, one of the key industries under the financial sector, plays a vital component in any country's financial system and significantly impacts the economy of the United Kingdom. Financial services in the UK are creative and globally integrated and offer services to partners worldwide (Hayward, 2022). However, future prosperity for the UK is not assured due to the anticipated challenges globally and locally. The banking industry is undergoing a radical change due to the digital transformation in financial services. Increasing competition, changes in business models, digitalization in every manual process, growing compliances and regulations, and rising customer expectations are critical challenges in today's banking industry. Therefore, a bank career is becoming more challenging than ever before. The COVID-19 pandemic has also worsened the situation by implementing working from home and digitalization in many human-involved banking processes. According to recent research, many financial institutes need help with technological obstacles and the shortage of skilled workplace employees (Mazurchenko et al., 2022).

Furthermore, while this digital transformation challenges the existing workforce in the banking and finance sector, they need a new, innovative and technologically skilled workforce. Due to all the above, talent leaders of the banking and finance sector must rethink their talent acquisition process to build their future workforce (Selimović et al., 2021). According to the Focus on the Future journal article, it is an unfortunate situation that challenges finding, developing, and retaining future bankers (Anderson & Joeveer, 2022). The primary regulatory responsibilities have generated a more complex work environment for young bankers (Anderson & Joeveer, 2022). The article further elaborates that there is a great necessity to identify and suggest education programs for young generations to enhance the skills required to overcome this situation.

On the other hand, many young people are looking for banking careers as the banking sector has many opportunities, benefits, perks, and reputations. Banking sector employees require various professional skills to successfully carry out their daily work when considering the digital transformation, global connectivity of operations, and diversity of roles. Furthermore, it will be challenging and complex for industry professionals and talent management units to map the required skills with the job role and the person with the position. Therefore, it is essential for a simple and agile framework to map those skills against the job easily.

The following report aims to map the critical professional skills required for varied job roles using NLP techniques and verify through the existing banking professionals to develop a helpful framework for job seekers, employees, and HR professionals. This finding will help better manage the existing resources and optimise the banking sector's productivity.

1.1 Problem Statement

A *skill* could be defined as a set of activities surrounded by a specific job. Also, skills could be categorised as soft and hard, generic, and basic skills (Canning, 2015). Identifying the right skills required for a job position is very important for employers and job seekers. More and more employers are turning to skill-based hiring rather than traditional credential-based recruitment. Even though specific soft skills match many professions, such as time management, teamwork, etc., each profession has complex skills. Furthermore, due to the everchanging technological advancement in workplaces, the requirement of job skills will also change more and more. Therefore, researching the skill requirements of jobs is vital to fill the skill gap in the labour market.

Furthermore, education, training and employability should go hand in hand to overcome the above gap. Resourcing is the most critical process in any organisation, as employees play the most valuable role in business growth. As such, identifying the right skills is more important, and it will be more critical when it comes to banking and finance as specialised skills may need to be able to screen through the academic qualifications. On the other hand, in this competitive job market, getting the right job and skills to perform the job are also challenging. Therefore, developing a framework to identify the right skills may help bridge the gap between current training processes and the labour market's needs.

Although standard skills are identified for bankers, there needs to be a system to track the current skill requirement based on the industry's everchanging competitiveness. No methodology can accurately determine industry-relevant skills required to perform a banking and finance sector job, provide growth patterns and insights to enhance training requirements or identify emerging skills required. This research will focus on the above and will do extensive research to identify better use of NLP techniques to identify skills based on the data extracted from employment vacancies on job portals.

1.2 Aims and Objectives

This research will contribute to identifying accurate skills required for banking sector employees. The research questions will be investigated below main areas.

1. How can NLP techniques be used for skill recognition?
2. How can Different Machine Learning models be used for the extracting and predicting the job skills required from a job description?
3. What is the relevant skills for a Job description through some Exploratory data analysis technique?
4. How to develop a model for robust skills extraction from unstructured texts
5. What are the most critical skills for Banking and Finance Sector employees?

This research will aim at a skill identification method that uses NLP techniques to distinguish skills required for employees in the Banking sector. To achieve the above, the below objectives were identified.

1. Investigate existing literature on NLP techniques and their applications in talent management within the banking industry, aiming to inform the development of a robust skill identification process.
2. Develop a comprehensive framework for identifying the competencies required by banking industry professionals through data extraction from UK job postings, leveraging web-based annotation tools and NLP techniques.
3. Utilize advanced algorithms and NLP approaches to automatically extract and categorize skills from unstructured textual data obtained from banking sector job advertisements, ensuring accuracy and relevance to industry needs.

4. Implement and evaluate various machine learning models within the established framework to predict the skills required for banking sector employees, utilizing metrics such as recall, precision, and F1-score to assess the effectiveness of the models.

1.3 Justification of the Research

The primary objective of this study is to develop a framework for identifying skills required for Banking sector employees. The framework aims to leverage the power of NLP techniques to extract, categorise and analyse skills required for professionals in the Banking Industry. By using the ability of NLP techniques to identify natural languages, this research seeks to enhance the accuracy and efficiency of skill identification and simultaneously automate identifying skills specifically for banking sector employees.

The impact of this research is that it can improve and revolutionise talent management and skill assessment practices within the banking industry. Commonly, skill identification has been a time-consuming and manual process that relies on subjective assessments and expert opinions. However, due to analysing a considerable amount of data using NLP techniques, this research offers a data-driven and objective approach to skill identification by overcoming the limitations of manual processes.

The use of NLP techniques for skill extraction and categorisation has numerous benefits. Firstly, NLP techniques enables the automated processing and analysis of large volumes of textual data, such as job postings. This enables the identification of skills in an efficient and accessible way, allowing organisations to handle substantial amounts of data and explore valuable insights from it. Secondly, NLP techniques provide the ability to capture semantic meaning and contextual understanding of skills. By employing algorithms and pre-trained language models, such as BERT or Word2Vec, this research can go beyond mere keyword matching and identify skills based on their inherent meaning and relationships. This leads to more accurate skill extraction and categorisation, resulting in a more precise and comprehensive skill identification framework.

Moreover, using NLP techniques reduces human bias and subjectivity in skill identification. By adopting automated algorithms and computational methods, this research ensures consistency and objectivity in skill assessment, avoiding potential biases associated with manual evaluations. Standardised NLP techniques also allow for easier comparisons and benchmarking of skills across different job roles, domains, and industries.

Developing a framework for skill identification in the banking sector using NLP techniques holds immense promise for improving talent management practices, enhancing recruitment processes, and enabling targeted skill development initiatives. By harnessing the power of NLP techniques, this research aims to provide organisations in the banking sector with a reliable and data-driven tool for identifying and understanding the skills crucial for industry success.

1.4 Research Methodology

This section explains the research methodology exercised in this study, defining the methods, procedures and techniques used to address the research objectives and answer the research question. The primary object of this section is to deliver a clear understanding of the approach taken to collect and analyse the data, as well as the rationale behind the chosen methodology. The research objectives of this study were established to explore the application of NLP techniques in skill recognition within the banking sector. Specifically, the aim is to conduct extensive literature on NLP techniques for skill recognition, prepare a comprehensive dataset containing banking job postings, develop a framework to identify the skills required for banking sector employees, validate the developed framework and evaluate results to identify skill sets.

The chosen methodology for this research aligns with the study's exploratory nature and the need to leverage data-driven approaches in the field of data science. By employing quantitative and qualitative techniques, the aim is to drive meaningful insights from the data and provide recommendations for skill recognition in banking. This study adopts an exploratory research design to explore the complexities and distinctions of skill recognition using NLP techniques. The utilization of a mixed methods approach, combining quantitative and qualitative methods to understand the research subject comprehensively. Therefore, this design allows us to capture objective measures and subjective interpretations of skill identification.

1.4.1 Theoretical Perspective

A research philosophy states the collection of fundamental principles that shape a research investigation's overall approach and conduct. Various research philosophies provide distinct perspectives on how scientific research should be followed. Research philosophy selection is of utmost importance as it should align with the research objectives, the nature of the research question and the available resources to conduct the research.

Before identifying the philosophy relevant to the research, it is essential to consider the differences in research assumptions typically made by scholars working within each philosophy (Saunders et al., 2015). These assumptions serve as fundamental principles for the research process and help identify the study's overall approach. Ontology and epistemology are more often discussed assumptions in research methodology. Ontology refers to the nature of reality and presence. It investigates queries about what objects or phenomena exist, their fundamental properties, and how they relate. Ontological concerns in the research context include understanding the nature of the examined subject matter and the assumptions made regarding its existence and qualities. It concerns the researcher's views about the nature of reality and what constitutes knowledge. Epistemology studies the nature of knowledge and how it is gained, justified, and understood. It investigates topics such as "What is knowledge?" How did we come to know what we know? What are the sources of information? In research, epistemological issues pertain to the researcher's views and assumptions about how information is gained, verified, or justified (Saunders et al., 2015).

Ontology is well suited to this research due to its relevance in understanding the nature of the data and underlying structure of the domain being studied. Here are the reasons for selecting ontology: Ontology provides a formal framework for defining a domain's entities, relationships, and attributes. Hence, this promotes data interoperability and facilitates data integration, enabling researchers to analyse and interpret diverse data sets more effectively. Further, Ontology allows for the modelling of domain knowledge and the explicit representation of concepts and their properties (Saunders et al., 2015). This facilitates the development of sophisticated data models that capture data semantics, enabling more precise and meaningful analysis. Furthermore, Ontology aids in data exploration and discovery by providing a conceptual framework for organizing and navigating data. Ontologies can also support advanced data querying and exploration techniques, enhancing the exploratory data analysis process (Smith, 2008).

The research philosophy accepted for this study will be a pragmatic approach since pragmatism underlines the feasibility and applicability of the study, focusing on solving real-world problems and generating actionable insights (Creswell & Creswell, 2018). By applying pragmatic philosophy, this study aims to close the gap among theory and practice in data science for skill recognition in Banking. In line with the Pragmatic philosophy, this research applies a mixed-methods research design. Using qualitative and quantitative methods allows for a complete understanding of the research objective and enables a more robust examination of the skills required for Banking sector employees. Quantitative methods, such as NLP techniques and statistical analysis, provide quantitative insights from the large data set obtained from job postings. On the other hand, qualitative methods, such as interviews and feedback from industry experts, offer subjective interpretations and contextual insights that validate the research findings.

Further, the Pragmatic philosophy is suitable for this research since it aligns to provide practical recommendations for identifying skill requirements of the Banking sector. By applying both qualitative and quantitative methods, this research philosophy aids in a more comprehensive examination of the research objectives by taking both subjective explanations and objective measures required to identify the skill requirements of the banking sector.

1.4.2 Research Approach

The research approach refers to the overall strategy and plan of how the researcher conducts the study and how to achieve the research objectives. The selected research approaches or approaches should guide the selection and application of specific research methods and techniques (Clark et al., 2021). The research approach for the objectives outlined as literature review, data collection and web annotation, framework development, and validation of results involves interpretivism, pragmatism, and a combination of positivism and interpretivism.

The objective of conducting an extensive literature review on NLP techniques, which could be used to identify skills required for Banking sector employees, will be to adopt an interpretive research approach. Since the focus will be on understanding the literature's subjective interpretation, meanings, and contextual consideration, this approach will be well suited. The pragmatic research approach will be employed to prepare a comprehensive data set containing job postings in the Banking sector using web annotation tools. This data set will be a foundation for further analysis and framework development. A combination of positivist and interpretivist research approaches will be used to develop a framework to identify the skills required for Banking sector employees. Firstly, a positivist approach will be employed to develop a structured framework based on observed facts and objective criteria. This will include identifying and categorizing skills using quantitative methods and implementing frameworks, such as statistical analysis and data mining techniques. After that, an interpretive approach will be utilized to obtain insights into the subjective explanations of the identified skills. This will include considering industry experts' practical implications and inputs to finetune and enhance the framework.

1.5 Structure of the Dissertation

This dissertation begins by discussing the relevance of the study issue in the banking industry, as well as establishing the research problem, questions, goals, and objectives. The technique and data sources are briefly reviewed, giving a preview of the upcoming conclusions. The Literature Review investigates the relationship between banking sector skills and NLP techniques, with an emphasis on NLP's effects on HR job skills, AI applications, growth of NLP, fundamental concepts of NLP, and talent identification. Chapter 3 discusses the practical elements, including data gathering methods, text processing, and model design. The Data Collection and Analysis chapter uses advanced data mining techniques to get insights into skill recognition and concluding with expert feedback. In Chapter 5, look at how to create machine learning models, include TF IDF, and experiment with classification algorithms. Chapter 6 evaluates results, emphasizing metrics like recall, precision, and F1 score. In Chapter 7, the dissertation concludes by summarizing findings, reflecting on contributions, and suggesting future research directions.

Chapter 2: Literature Review

2.1 Introduction

Acting as a foundation for financial transactions, investments, and economic growth, the banking industry is critical to the global economy. Due to the industry's dynamic nature and the ongoing growth of financial services, the need for a qualified workforce in the banking sector has become more critical. Banking organizations demand workers who can negotiate the complexity of modern finance, adapt to technology changes, and serve clients with effective and innovative solutions. Banking institutions' success and competitiveness are dependent on their workforce's expertise and adaptability. Hence, recognizing and comprehending the different abilities required for the various roles in the banking industry is a strategic need for organizational success. Therefore, the integration of cutting-edge technologies becomes critical in the aim of finding and leveraging the necessary capabilities within the banking staff. NLP, an artificial intelligence area, is emerging as a revolutionary technology in the Human Resources (HR) domain. This literature explores the relationship between the banking industry, staff skills, and various elements of NLP's application in skill identification and revealing approaches.

2.2 Banking Sector Job Skills

Job skills are the different talents utilized to do the profession, ranging from workplace skills like teamwork to technical skills like computer programming. While businesses still appreciate credentials such as degrees, more and more are shifting to skills-based recruiting to fill vacancies. So, job skills are becoming increasingly important.

Skills can be categorized into three types: workplace skills, technical skills, and transferable skills. Workplace skills: personal abilities that guarantee accomplishing the job, such as teamwork, time management, and problem-solving. It is also known as "human talents," "employability skills," or "soft skills". Technical skills often referred as Technical or certain job-related proficiency, such as the ability to develop web applications using HTML, do data analysis, or utilize specific accounting software (National Research Council, 2012). They are frequently described to as "hard skills." Transferable skills are workplace and technical talents that can be transferred from one profession to another, such as when someone utilizes their affinity for teamwork and their ability to write in Python to shift occupations from programmer to STEM instructor. Transferable skills are many different skills that may be transferred across occupations (Robles, 2012).

Reacting to current skill situations or forecasting future skill situations is only possible when trustworthy and accurate skill information is accessible when it is needed (Breugel, 2017). There are several quantitative and qualitative approaches accessible, and each of them can fulfil specific research objectives, such as the type of skills and the number of people who require these talents now or in the future. These strategies also need the participation of several organizations like ministries, social collaborators, educational institutions, regional authorities, public employment agencies, and councils dedicated to sector-specific skills. Finally, these strategies need a particular mix of resources, including data, human, and financial resources. Input-output models, computable general equilibrium models, workforce requirement approach, informed opinion and specialized knowledge, employer surveys, labour force surveys, and scenario development are some of the approaches and sources mentioned (Breugel, 2017). Requirement of skills could be determined by combining these findings with comprehensive explanation of job roles, including skill information and level of educational achievement (education type, level, and field of study. Furthermore, this could also be used to determine the Vocational Education and Training program that is required. Details about skill requirements can be applied for various policy areas, including employment, education and training, migration, and social and other development concerns (Breugel, 2017).

Many industries are undergoing radical changes in the business process due to digitalization. Hence, identifying the right skills required for its labour force is crucial. Banking is such an industry which has been going through a structural change for many years. Financial services have been pivoting as more individuals rely on digital to complete activities and expect enterprises to fulfil their new digital demands. Therefore, focus on digital revolution has risen in recent years. Discovering strategies to surpass client prospects and remain aggressive are the significant challenges that banking sector employees face today.

Employability skills are not just job-specific; they cut horizontally across all industrial sectors and vertically across all positions from entry-level to chief executive officer. Employability skills have become a big concern for students, educational institutions, and the government in the current context since employers in all sectors of the economy require such skills. With the passage of time, changing workforce mix, personal values, rapid improvements in Information and Communication Technology (ICT), increased global competitiveness, and the advent of the knowledge economy have produced several obstacles for every firm (Renu, 2021). Numerous advancements in wireless technology and the growing adoption of smart mobile devices have given rise to worldwide mobile banking. As a result of these changes, bank officials now require current employability skills such as sense-making, novelty, cross-cultural competency, IT and new-media literacy. Therefore, to transmit skill training to the youth and existing workforce on the banks, it is necessary to raise awareness so that students and staff can better understand such talents and have better future chances.

A study was employed to inspect the level of information of reading and writing skills of banking sector employees, their requirement of need and usage of these skills (Taoffik et al., 2016). However, the study only applies to banks in Ogun State, Nigeria. Five banks were selected at random, and 125 bank employees participated in the survey. The participants were given 125 copies of the survey created for the study, 75 (60%) of which were completed and returned. The research findings revealed that bankers possess a grasp of information literacy and there exists a notable correlation between their information literacy and their utilization of information. Among bankers there is a need for job related information health related information and financial related information. Additionally, they frequently engage in activities such, as staying updated with affairs conducting research and providing quality service to cater to their informational needs (Taoffik et al., 2016).

Data Science, Machine learning, and Artificial Intelligence are technologies that can fundamentally change industries. Banking is no exception; AI and machine learning have been used in almost all systems, such as core banking systems, online banking, Human Resources Information Systems (HRIS), fraud detection, etc. To identify the role of big data as a service and application in the Indian Banking sector has been evaluated by (Jaspreet Singha, 2023). This article examined how big data as a service and application has proven to be a powerful and creative tool that supports the Indian banking sector in identifying security risks and fraudulent conduct and early prevention. The results of an analysis conducted on Scopus and Web of Science databases using Vos Viewer software clearly demonstrate the importance of integrating big data technology into the Indian banking sector. The findings presented in this study will greatly benefit researchers in this field by enhancing their knowledge and understanding of big data technologies and methods employed for managing financial risks in banking as well as their role, in operational banking workflow management. According to the research, deep learning, representation learning, user profiling, knowledge graphs, NLP, and graph and network analysis have been identified as critical data science tools for banking applications.

2.3 Relationship between NLP Techniques and HR Job Skills

NLP techniques has significant implications for HR job skills, as it can automate many repetitive, time-consuming tasks that HR professionals perform daily, freeing them up to focus on higher-level strategic work. Here are some specific ways NLP techniques impacts HR job skills: NLP techniques can help automate resume screening and identify the most qualified candidates. NLP algorithms can analyse candidates' resumes and job descriptions, identifying relevant keywords and assessing the candidates' fit for the position.

Many HR departments are adopting chatbots and virtual assistants powered by NLP techniques to handle routine queries from employees and candidates. These virtual assistants can answer common questions about benefits, policies, and procedures, freeing HR professionals to focus on more complex tasks. NLP techniques can analyse employee feedback and performance reviews to identify patterns and trends. This can assist HR experts in pinpointing areas where employees might require training or assistance enabling them to offer more precise feedback.

NLP techniques can analyse employee feedback and sentiment to identify potential retention issues. This can help HR professionals proactively address employee concerns and improve engagement. Overall, NLP techniques has the potential to revolutionize HR job skills by automating many routine tasks, freeing up HR professionals to focus on higher-level strategic work, and providing more accurate, data-driven insights into employee performance, engagement, and retention. As such, HR professionals who are familiar with NLP techniques and its applications will be better positioned to succeed in the modern workplace.

2.4 HR applications with NLP Techniques

Due to the automation and computerization of many human processes, there is abundant data in many fields, which could be used for meaningful insights in their specific fields.

Accordingly, data is the foundation of any organization of the 21st century. Human Resources (HR) and companies have been fascinated by data analytics during the past few years for the betterment of their functions. HR has benefited from data analytics by being more and more strategic and objective than ever. HR needs to adopt the potential of analytics despite being a corporate department that has historically been a major consumer and generator of enormous amounts of data (Editor et al., 2021). However, artificial intelligence, machine learning, and NLP have changed this. With this new technology, HR teams could streamline their analytical efforts to a greater extent.

While AI and ML have been extensively used in the HR world over the past few years, NLP techniques are still a new concept for the domain of HR. HR is a business function which is heavily dependent on human communication. Therefore, NLP techniques can assist all HR functions from the onboarding to the exit interview. Furthermore, as HR chatbots and text analytics grow more common in HR, NLP techniques may aid HR professionals in transforming enormous quantities of text into measurable insights. This information may be used to make decisions about employee retention, engagement, succession planning, talent management, and employee performance.

The field of Human Resources Management (HRM) has found ways to leverage previous breakthroughs, such as computers and the internet to enhance productivity, cost efficiency and competitiveness in the market (Hmoud & Várallyai 2020). The integration of technologies like a trebuchet has accelerated the development of Human Resource Information Systems (HRIS) incorporating emerging capabilities like Artificial Intelligence, into tactical HR procedures known as tactical HRIS (T-HRIS). The volume of organizational, people and task-oriented data for which HR is naturally accountable has resulted in adopting AI in many tactical HR procedures since it improves long-term business models (Di Vaio et al., 2020). AI is being utilized more and more in areas of tactical operations such as resourcing, employee performance evaluation and satisfaction analysing, salaries and benefits studying, best practices, managing disciplinary actions, and implementing employee training and development systems. To further comprehend this trend, Alexis et al (Alexis et al., 2021) have investigated published sources and literature discussing AI's use in HRM. This study has uncovered the components of tactical HRIS (T-HRIS) that are discussed in existing literature.

By using a literature review approach the researchers have examined how each T-HRIS component is described and portrayed.

Moreover, this paper offers an overview of HRIS and HRM delving into the evolution of AI in the field of HR and the methods employed. It also outlines the Systematic Literature Review (SLR) methodology utilized and provides insights into the constituents of T-HRIS, their representation and future research considerations (Alexis et al., 2021). Tactical HRIS (T-HRIS) pertains to the amalgamation of Human Resource professionals and technology for executing specific activities inherent in Human Resource Management with the aim of achieving corporate objectives. To depict the structure of T-HRIS Figure 1 presents a framework that defines its components.

This study investigates the significance of AI in conjunction with HRM practices by exploring existing literature on HRIS components (Alexis et al., 2021). The researchers identified gaps in literature between managerial and technical HRIS practices highlighting a need for more intuitive and emotionally intelligent practices. Furthermore, by analysing 33 publications from four databases encompassing six T-HRIS components, four different AI approaches and four types of articles; it becomes evident that there exists a research gap between technical and managerial aspects, within T-HRIS (Alexis et al., 2021).

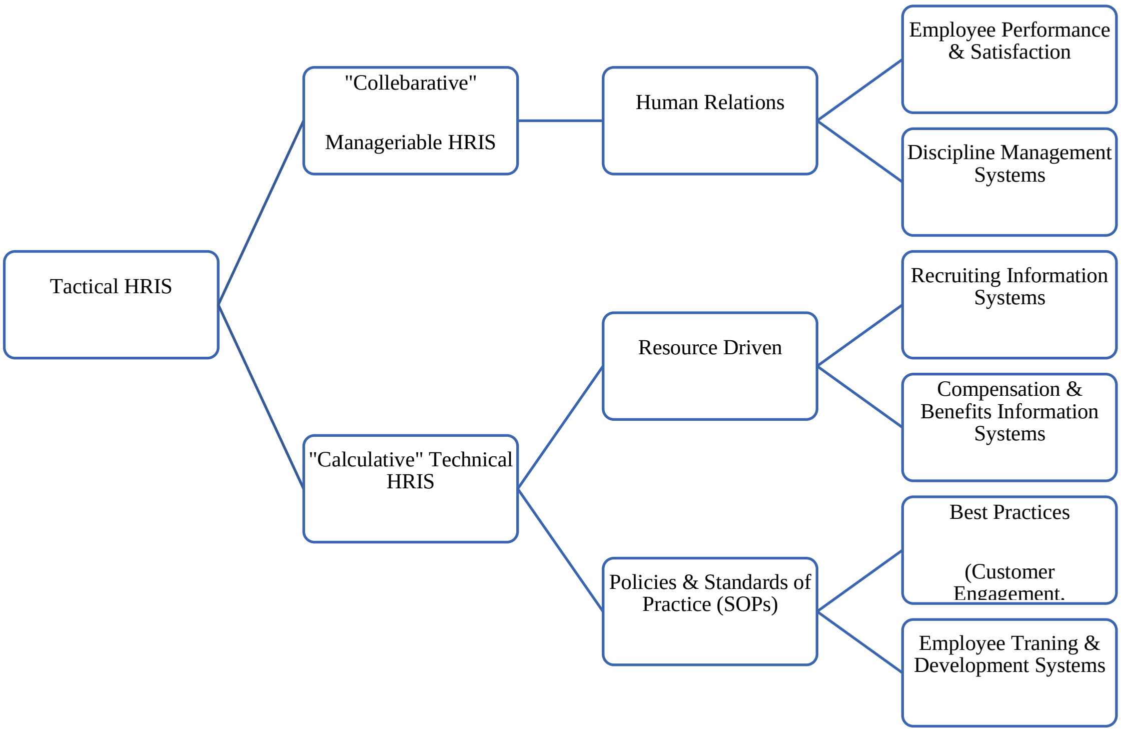

Tactical HRIS
"Collebarative"
Manageriable HRIS
"Calculative" Technical HRIS
Human Relations
Resource Driven
Policies & Standards of Practice (SOPs)
Employee Performance & Satisfaction
Discipline Management Systems
Recruiting Information Systems
Compensation & Benefits Information Systems
Best Practices
(Customer Engagement.
Employee Traning & Development Systems

Figure 1: Tactical HRIS Framework (Alexis Megan Votto, 2021)

More data-driven and task-oriented T-HRIS applications are overrepresented in the literature, whereas management techniques are underrepresented. The assessment of existing research lays the groundwork for future research to bridge the gap between technical and management T-HRIS AI applications (Alexis et al., 2021).

Advancements in NLP have allowed for efficient evaluation of narrative text particularly in the context of performance appraisal (PA) and organizational sciences. A recent study conducted by Speer and colleagues (Speer, 2020) demonstrated that NLP can effectively capture variations in performance related narratives, which can then predict behavioural outcomes. In the past qualitative comments on employee performance were often overlooked in PA studies due to the challenges associated with coding and analysing narrative text. To overcome this limitation researchers utilized a predefined taxonomy of performance criteria to develop and evaluate improved NLP generated sentiment scores for performance comments (Speer, 2020). This research explored NLP scoring methods, including traditional approaches as well as more contemporary techniques. It also investigated the generalizability of these NLP scores across rating sources and survey designs. Overall, this study offers insights, into how researchers can develop theory driven NLP algorithms and validate their effectiveness (Speer, 2020).

Aqel (Aqel, 2014) conducted a study on employee appraisals and published his findings in ProQuest Dissertations & Theses. The author aimed to develop a framework for employee evaluations based on inductive reasoning in programming and data mining methods. The author first discussed the importance of employee appraisals in organizations and the various challenges associated with this process. He then introduced the concepts of inductive logic programming and data mining methods and discussed their potential for improving the accuracy and efficiency of employee appraisals. The author developed a framework for employee appraisals based on the proposed inductive logic programming and data mining methods and evaluated its performance through experiments. The results showed that the framework effectively improved the accuracy and efficiency of employee appraisals (Aqel, 2014).

Due to its ability to read human language, NLP techniques have recently been widely used in resume screening applications. In the paper "An End-to-End Framework for Information Extraction from Italian Resumes" by Barducci (Barducci et al., 2022), the authors propose an end-to-end framework for information extraction from Italian resumes. The framework aims to automate extracting meaningful information from resumes, such as education, work experience, skills, and personal information. The framework utilizes various NLP techniques to extract this information efficiently and accurately. The authors evaluate the framework's performance on a dataset of Italian resumes and show that it outperforms existing methods in accuracy and efficiency. This research has the potential to help organizations save time and resources in the recruitment process by automating the information extraction process from resumes.

Bondielli and Marcelloni (Bondielli & Marcelloni, 2021) have explored summarization and transformer architectures in resume profiling. Their study, published in Expert Systems with Applications, aims to provide insight into the effectiveness of these techniques for resume profiling. The authors begin by highlighting the importance of accurately profiling résumés for various applications such as human resources and recruitment. They then introduce the concept of summarization and its potential for improving the efficiency and accuracy of resume profiling. Next, the authors examine the use of transformer architectures for this task. They discuss the advantages of using transformers, such as their ability to handle sequential data and their ability to handle large amounts of information. The authors also mention the limitations of using transformers, such as their high computational cost and the need for extensive training data. Bondielli and Marcelloni (Bondielli & Marcelloni, 2021) present the results of their experimental study, which compares the performance of summarization and transformer-based approaches for resume profiling. They found that the transformer-based approach outperformed the summarization approach regarding accuracy and efficiency. Overall, the authors conclude that using summarization and transformer architectures holds promise for improving the accuracy and efficiency of resume profiling.

A literature review of the study "Classifying Online Job Advertisements through Machine Learning" by Boselli et al. (Boselli et al., 2018) would provide an overview of the current state of research on machine learning for classifying online job advertisements. The study presents a machine learning model to categorize job advertisements into categories, such as job titles and industries. The authors used a dataset of online job advertisements and evaluated the machine learning model's performance in classifying the advertisements. The results showed that the model was effective in accurately categorizing the job advertisements. This study contributes to the field by demonstrating the effectiveness of machine learning for classifying online job advertisements. The study's findings suggest that this approach could be helpful for organizations and recruitment agencies to sort and categorize job advertisements efficiently.

Kuodytė and Petkevičius (Kuodytė & Petkevičius, 2021) studied education-to-skill mapping using hierarchical classification and transformer neural networks. The study aimed to improve the process of matching education to the necessary skills for a specific job. The authors proposed a framework combining hierarchical classification and a transformer neural network to map education to skills. The framework uses a hierarchical classifier that divides education into different categories based on their levels. The second stage involves using a transformer neural network to map the education to the skills. The authors evaluated the accomplishment of the proposed framework using a job description database and found that it outperforms other existing models in mapping education to skills accurately.

"Intelligent Recruitment System Using NLP" by Sharma et al. (Sharma et al., 2021) involves the development of a recruitment system that uses NLP techniques. The authors aim to automate recruitment and reduce human involvement by using NLP techniques to extract meaningful information from resumes and job descriptions. The system is designed to match resumes with job descriptions and highlight job applicants' relevant skills and qualifications.

The authors use various NLP techniques to develop the system, such as text classification, information extraction, and text clustering. The system's performance is evaluated based on accurately matching resumes with job descriptions.

Enterprise Tech web article by Linly Ku explained how NLP techniques could change the future of HR (Ku, 2021). According to the author, NLP-enhanced automated interviews (NLP) are used in chatbots to identify any blind spots or human behaviour quickly and accurately in the interviewing process. These tools can help managers improve candidates' fit in the company by analysing their words, speech patterns, and facial expressions. Censia and Mya are examples of such recruiting chatbots empowered by NLP techniques. Onboarding is an essential part of effective onboarding, and NLP techniques is used in chatbots to answer any employee question. NLP techniques can be used to analyse customer feedback and offer positive and practical advice to improve future involvements. *Ambit* is a tool that could be used for effective onboarding. Furthermore, many built-in NLP applications exist for other HR functions, such as employee engagement, training, and development. In the end the author asserts that numerous AI applications would be unimaginable or precise, without NLP. Yet, NLP alone is not enough to solve the entire problem.

2.5 Artificial Intelligence and its Synergy with NLP

Artificial intelligence (AI) is a machine's ability to reproduce or improve human intelligence, such as analysing and learning from experience. Artificial intelligence has long been utilized in computer programs but has recently been used in various products and services.

The field of intelligence has a rich and intricate history, marked by the unwavering dedication of diligent researchers who have navigated through funding fluctuations periods of promise and uncertainty all with the goal of advancing more realistic AI (Smith, 2018). One significant milestone in AIs days was the development of the Turing test in 1950 by Alan Turing. This test served as a benchmark to assess how well a machine could imitate interaction. According to this test for a machine to pass it must convincingly deceive a judge into believing it is human. Concurrently with the inception of this test Turing’s publication "Computing Machinery and Intelligence" enabled machines to comprehend and store information using language while also applying knowledge and responding appropriately to various scenarios (Smith, 2018). Turing is widely regarded as the pioneer of AI and computer science as a discipline. Although his initial research required computers costing up to $200,000 per month (Patrick, 2018) his dedication propelled him far beyond those early developments. In the 1980s deep learning emerged as a point for ongoing AI research efforts. Neural network research was revitalized thanks to the contributions of John Hopfield and David Rumelhart along with financial investments, from Japan, the United States and the United Kingdom.

This period is also known as the resurgence of AI. The modern era of AI is commonly referred to as the time between 1997 and 2005. It was in 1997 when IBMs Deep Blue system defeated Garry Kasparov, the world chess champion capturing the attention of the public towards AI (Smith, 2018). Yann LeCun, a student of Geoffrey Hinton made significant strides in computer vision by creating the Convolutional Neural Network (CNN) at Bell Laboratories within a year using the backpropagation algorithm and years of research on computer vision applications (Smith, 2018). The long short-term memory unit (LSTM) developed by Hochreiter and Schmidhuber remains widely used for sequence modelling today. In terms of research breakthroughs, Hinton and two of his students demonstrated in 2012 that deep neural networks surpassed all methods in large scale visual recognition challenges for image recognition. This marked the birth of artificial intelligence (AI). Since its inception in the 1940s and 1950s AI has made progress (Smith, 2018). Many present-day technologies and concepts have roots in these early discoveries. Visionaries like Geoffrey Hinton played a role, in driving AI forward through various ups and downs during the latter half of the twentieth century.

NLP is a branch of AI that assists computers in recognizing how people write and communicate. This is a challenging job because of the large amount of unstructured data. Moreover, the way individuals communicate through speech and writing (commonly known as "tone of voice") is distinct for each person and continually evolves with its widespread usage. According to the journal article "The Power of Natural Language Processing" by Gruetzemarcher (Gruetzemacher, 2022), NLP is the most powerful tool of Artificial Intelligence (AI) when it comes to decision-making like humans or even better than humans. Even though AI was better than humans in data-driven decision-making, until recently, it was still lower than humans for cognitive and creative thinking. However, NLP has changed that exceptionally with its ability to process language as humans do.

Furthermore, according to the author, NLP has been used in very advanced language tool models such as GPT-3 and Elicit. This AI-based research assistant tool can even do literature on specific research topics. However, the article further elaborates that NLP needs to be more utilized as businesses have not fully utilized advances of NLP to make better values for their businesses, make better decisions, recognize the potential of the labour force, and understand their capabilities.

2.6 Evolution of NLP

In the 1950s, NLP emerged as a specialized field that blended artificial intelligence and linguistics (Nadkarni et al., 2011). However, NLP became famous and used practically when Machine Learning (ML) and deep learning (Arumugam & Shanmugamani, 2018) significantly advanced. There are several applications that this technology can be used in various fields such as Sentiment analysis, Named Entity Recognition (NER), machine comprehension, speech recognition, and more (Arumugam & Shanmugamani, 2018).

Because of advances in technology, scholars now have access to a vast amount of written knowledge. This helps them study and understand things happening today more thoroughly. NLP is a computer-assisted technique that helps to understand and gain insights from human language (Reese & Bhatia, 2018). It allows researchers to extract valuable information from text have been significant advancements in NLP. These incredible advancements in machine translation (Och, 2003), pattern matching (Califf & Mooney, 2003), sentiment analysis (Liu, 2012), and speech recognition (Weber, 2002) have truly made a lasting impact on our everyday lives. They have also completely transformed how businesses operate. For instance, the Norwegian News Agency (NTB) recently reported that robotic journalists can deliver news articles within 30 seconds after an event has ended. Yeah, NLP has sparked some incredible innovations. datasets efficiently, reducing the computational work required. In the past few years, there the existence of smart devices like voice-activated assistants and simultaneous translation tools like Baidu's STACL2. Moreover, remember speech-to-text recognition, sentiment-based market projections, and those handy intelligent shopping guides. Oh, and Alibaba has even developed robotic customer service attendants. It is incredible how these things have changed how businesses operate.

More and more scholars are using NLP to understand better textual datasets found in annual reports, press releases, online reviews, and other sources. NLP has the advantage of capturing concepts objectively and accurately, which helps reduce the laborious task of manually coding text. NLP has made a significant impact on the development of management theories. At the end of 2019, 72 articles in the UT Dallas List of 24 Top Business Journals (UTD List) used NLP as the primary analytical approach. However, despite much research on NLP, there still needs to be a big gap.

The requirement of complete overview of how NLP techniques has been used in management research, and there is also a need for a detailed guide on how to use it practically as an analytical tool. So, it is essential that we thoroughly review all the existing literature on how NLP techniques is being used in management research. A review like this can help anyone to understand how this topic can be applied across different fields and offer new ideas for improving management theories. In addition, providing a detailed guide on using NLP techniques for data analysis can help researchers who work with text data easily understand and use it. We focused our review on articles included in the UTD List because they tend to have strong theoretical foundations, rigorous empirical designs, comprehensive analytical methodologies, and extensive datasets. According to Ricks, Toyne, and Martinez (1990), it is essential to review recent literature to have a comprehensive understanding. This choice aligns with that principle (p. 220).

NLP is the capacity of a machine to successfully converse with people in their native language. This skill has helped robots to better grasp both voice and text and to develop meaningful responses to human incentives. When chatbots and language analysis algorithms began presenting themselves and expediting different T-HRIS processes such as employee on boarding, recruiting, training, and leave requests, such AI methods have grown increasingly widespread when communicating with consumers and aiding employees (Majumder and Mondal, 2021). A chatbot may engage with the user and answer their inquiries on its own.

NLP is a modern computer technology that allows for the analysis of naturally occurring texts in a human-like manner. NLP allows a machine to understand and analyse text. By examining the structure of phrases, grammar and sentence elements like nouns, adjectives and verbs the analysis extracts an interpretation, from natural language texts (Zaho, 2017). Earlier NLP applications relied on a complicated set of hand-written rules, but more modern NLP applications rely on machine learning techniques, broadening its reach to include the artificial intelligence sector. Logistic regression (LR), Nave Bayes (NB), and support vector machine (SVM) are three popular NLP modelling methods (Witten, 2016). Because of its scalability to numerous class values and the substantial benefit of model sparsity, LR maximizes probability estimations. Even when predictors are rather weak, NB can swiftly and reliably document categorization. Using high-dimensional input spaces, SVM can detect linear and polynomial separators and tackle text categorization (Frank, 2016). While being utilized in other areas for years, NLP is emerging as a potent tool for improving efficiency in the architectural, engineering, and construction (AEC) sector.

Salama and El Gohary conducted a study on categorizing construction documents using NLP. Their categorization approach involved labelling each document as either "good" or "bad" (Salama and El Gohary, 2013). They employed techniques such as Naive Bayes (NB) Support Vector Machine (SVM) and Maximum Entropy (ME) models measuring their performance based on Precision and Recall metrics.

In another study by Zhang et al. NLP approaches were used to evaluate construction accident records, which were classified into eleven categories (Zhang et al., 2019). Several models were employed in this research including Logistic Regression (LR) Naive Bayes (NB) Support Vector Machine (SVM) K neighbour (KNN) and Decision Tree (DT). The effectiveness of these models was assessed using the F1 score metric (Coal et al., 2016).

NLP has also been applied in other areas within the ACE field. These include extracting safety information from text reports classifying stakeholders concerns and client opinions regarding AEC project processes predicting cost overrun levels during project estimation stages as well as creating knowledge maps and enhancing semantic clarity for facility management purposes. However, there is research on NLP applications related to personnel assignment in the AEC domain specifically focusing on building maintenance tasks. This research aims to address that gap, in the literature (Zhang et al., 2019).

The purpose of the model is to arrive at an answer for staff assignment predicated on the service request. Crew and Priority are the two output labels to forecast to allocate personnel (see figure 2). The dataset lays out 19 distinct classes for Crew and 2 for Priority, hence the former is predicted using a multiclass classification and the latter using a binary classification. Notable occupations in building maintenance include electrician (EL), labourer (LB), and plumber (PL), which together make up 51% of Personnel, while routine tasks (RG) comprise 81% of Priorities.

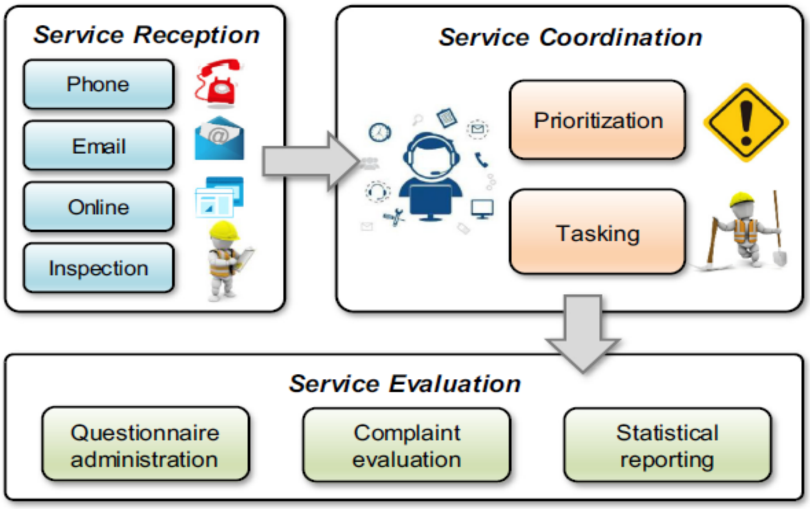


Figure 2: Workflow Service Request Processing in NLP (Liddy et al., 2012)

Dataset documents with the label "Brief explanation for explaining above theory" are used as inputs to make predictions about other texts. Extended description texts are likewise good inputs; however, they are optional and are not provided by the majority of requests. In addition to the short description, additional labels (such as Location) can be included as metadata. Metadata is data about data, and it is used to describe and organize data resources.

This one of theoretical development of using NLP techniques in the service sector to design the model and get maximum benefit from it. NLP is of utmost prominence in AI and computer science. Research into NLP entails the development of ideas and techniques that facilitate efficient verbal interaction between people and machines. NLP is an interdisciplinary branch of research that aims to convert human speech into machine instructions. It draws upon computer science, linguistics, and mathematics. Both Natural Language Generation (NLG) and Natural Language Understanding (NLU) are subfields of NLP (NLG). Ultimately, Natural Language Understanding (NLU) aims to decode texts and extract useful information for subsequent tasks by understanding natural language (human language) (Schank, 1972). In contrast, natural language generation (NLG) refers to the generation of human-understandable content in a variety of natural languages from pre-existing structured data, text, pictures, audio, and video (McDonald, 2010). There are three subcategories of NLG: text-to-text (Genest & Lapalme, 2011) tasks like translation and abstracting; text-to-other (Xu et al., 2018) tasks like creating graphics from text; and other-to-text (other-to-Text) tasks like creating text from video (Rohrbach et al., 2013).

NLP has gone through four stages of development; the early years before 1956 a period of intense activity from 1957 to 1970 a slowdown in progress, between 1971 and 1993 and the current era of revival starting in 1994 (McDonald, 2010). The early stages of NLP study fall under the umbrella of the germination phase. The term "Turing Machine" was coined by Alan Turing in 1936. Modern computing can be traced back to the theoretical foundation of the "Turing Machine," which in 1946 led to the development of the electronic computer and therefore provided a concrete foundation upon which machine translation and NLP could be built. It was around this time that the groundwork for NLP was laid, owing to the need of machine translation. By using a probabilistic model based on discrete Markov processes, Shannon automated the description language in 1948. He then attempted to quantify the quantity of information available in human language by applying the thermodynamic idea of entropy to the probabilistic algorithm of language processing. Kleene studied finite automata and regular expressions in the early 1950s, while Chomsky introduced context-free grammar to NLP in 1956. Through this research, two rule- and probability-based NLP methods emerged. The debate between rule-based versus probabilistic approaches to NLP has raged on for decades. Since artificial intelligence (AI) was first developed in 1956, a new era of NLP has begun, with AI steadily combining with other NLP technologies over the following few decades to improve the technical means of NLU and NLG, consequently expanding the social application of NLP.

The years between 1957 and 1970 were pivotal for the growth of NLP, as this is when NLP was initially incorporated into the study of artificial intelligence. Research into both rule-based and probabilistic approaches advanced greatly during this time period. Formal linguistics and generative syntax were first studied by symbolist researchers like Noam Chomsky from the 1950s through the 1960s. At the same time, probabilistic method experts embraced the

Bayesian approach based on statistical research techniques, and they too made significant strides forward at this time. Yet, in artificial intelligence, most researchers have focused solely on logical and rational thought.

Yet only a small percentage of statistics and electronics majors have taken the time to learn about probability-based statistical approaches and neural networks. Around this time, the University of Pennsylvania's groundbreaking Transformations and Discourse Analysis Project (TDAP) was created in 1959, and the Brown University American English Corpus was established. Neisser, an American psychologist, proposed cognitive psychology in 1967, connecting NLP squarely to the study of the mind. The second period of significant growth in NLP occurred from 1971 and 1993. At that time, it was clear that NLP-based applications would not be able to be quickly solved, and that new issues related to the statistical methodology and the construction of a corpus were continually cropping up. As a result, a lot of individuals stopped looking at NLP with any seriousness. Hence, the study of NLP hit rock bottom in the '70s.

Despite this, scientists in wealthy nations persisted in their investigation, and in the 1970s, they made significant advancements in the field of speech recognition by developing statistical approaches based on the Hidden Markov Model (HMM). Additionally, important advancements in human-machine dialogue were achieved with the advent of discourse analysis in the early 1980s. In the years that followed, as NLP scholars re-examined their work, finite state models and statistical techniques also made a comeback.

Two major developments occurred after the mid-1990s that sparked renewed interest in NLP study. The first is the fast improvement in computer speed and storage, which has strengthened the material basis for NLP and allowed for the commercialization of speech and language processing. The Internet's eventual commercialization in 1994 was the second major development. Natural language-based information retrieval and information extraction have become increasingly important as a result of the rise of network technology throughout this time period. Yoshua Bengio presented the feed-forward neural network as the first neural language model in 2001.

Multitasking was initially used to NLP's neural network by Ronan Collobert in 2008. Tomas Mikolov created Word2Vec, a neural network–based statistical approach for learning independent word embedding from a text corpus, at Google in 2013. The sequence-to-sequence learning model, published by Ilya Sutskever in 2014, is a generic framework for converting one sequence into another using a neural network. Machines' comprehension and reproduction of human speech is improved with the help of these statistical models.

Scholars in domains outside of computer science have recently begun to recognize the benefits of NLP and apply it to study, joining others who have been working to improve current algorithms or propose new approaches for NLU and NLG. For instance, numerous management theorists have put forward alternative approaches and enhanced algorithms to deal with various management situations. For instance, Li and Qin (2017) in the discipline of classification introduced the de-identification and Anonymization for Sharing Medical Texts (DAST) framework to help with the clustering of medical text data. By presenting an innovative active learning approach for large-margin classifiers, Xu, Liao, Lau, and Zhao (2014) attempted to get around the problem of correct annotation amid a sea of manually labelled data. A novel method for assessing market structure was proposed by Gabel, Guhl, and Klapper (2019), which combines a neural network language model with dimensionality reduction. Similarly, T. Y. Lee and Bradlow (2011), proposed a method for determining product qualities and a brand's relative position from online customer evaluations, with a concentration on market structure research.

Small-scale investor sentiment may be extracted from stock message boards using a technique described by Das and Chen (2007). Fang, Dutta, and Datta (2014) developed a hybrid method to sentiment analysis that use sentiment analysis to address the prohibitive expense of gathering labelled data. Toubia, Iyengar, Bunnell, and Lemaire (2018) suggested guided latent Dirichlet allocation (LDA) to extract aspects of entertainment items to better topic modelling. Sentence-level data analysis (sent-LDA) developed by Bao and Datta (2014) is another way that enhanced topic modelling in 2014. It is used to quantify different forms of risk based on 10-K filings. As a means of concisely characterizing items in terms of latent topics and specifying customer preferences via topics, Ansari, Li, and Zhang (2018) constructed a new covariate-guided, heterogeneous, supervised topic model. Because J. Liu and Toubia (2018) considered the semantic connection between two distinct document categories, they developed a hierarchical dual latent Dirichlet allocation (HDLDA). Based on the CTM, Trusov, Ma, and Jamal (2016) offer a method for deducing user profiles from web navigation records by factoring in visiting intensity, heterogeneity, and dynamics. To provide an effective and economical summary of legal decisions, Bansal et al. (2019) suggested a unique Fuzzy Analytical Hierarchical process (FAHP) based on sentence structure, term-frequency, theme word, and sentence proximity to analyse phrases. To improve the quality and efficiency of EMR annotation, Xu et al. (2016) proposed an indirect annotation approach. This process decomposes medical terminology into individual words and then uses phrase sense disambiguation (PSD) on 'compound terms' to narrow down potential annotation terms. All these papers suggest that NLP is being used more often in management research. Thus, our work examines the usage of NLP across fields to figure out how it has been employed and offer research possibilities based on interdisciplinary comparison.

NLP involves computer-aided analysis of substantial amounts of natural language data to extract meaning and value for use in practical applications. To achieve this there are many concepts and terminologies related with NLP. Next section will elaborate these basic concepts and terminologies of NLP.

2.7 Basic Concepts and Terminologies of NLP

NLP involves computer-aided analysis of substantial amounts of natural language data to extract meaning and value for use in practical applications. To achieve this there are many concepts and terminologies relate with NLP. This section will elaborate these basic concepts and terminologies of NLP.

2.7.1 Text Corpus or Corpora

A corpus is a sizable collection of text data in one or more languages, such as English, French, and so on. A single document or a collection of documents might make up the corpus. The corpus is divided into sections for further analysis in different NLP tasks. These units might be in the form of sentences, paragraphs, or words (Arumugam & Shanmugamani, 2018).

2.7.2 Paragraph, Sentences, Phrases and Word

The greatest text unit that an NLP job can handle is a paragraph. Without being broken down into sentences, paragraph level limits may not be very useful on their own. Even yet, there are situations when paragraph limits may be appropriate (Arumugam & Shanmugamani, 2018).

The next lexical level of linguistic data of NLP is a sentence. A whole meaning, thought, and context are included in a sentence. It is often taken out of a paragraph depending on the bounds set by punctuation, such as a period. Moreover, the statement may represent an opinion or attitude (Arumugam & Shanmugamani, 2018).

A phrase is a collection of related words that appear together in a sentence and have a single meaning. For search and retrieval purposes, certain NLP tasks extract important phrases from sentences. The word is the next smallest textual component. Based on punctuation, such as spaces and commas, the standard tokenizers divide sentences into words (Arumugam & Shanmugamani, 2018).

2.7.3 N-grams

An N-gram is a collection of letters or words. For instance, a character unigram is made up of only one character, a bigram is made up of two characters in succession, and so on. Similar to word N-grams, a series of n words makes up a word N-gram. N-grams are characteristics in NLP that are used for things like text categorization (Arumugam & Shanmugamani, 2018).

2.7.4 Bag-of-Words

Unlike the N-grams, bag-of-words does not consider word order or sequencing. It records the word frequency distribution in the corpus of texts. Moreover, NLP applications like sentiment analysis and subject identification employ the bag-of-words as characteristics (Arumugam & Shanmugamani, 2018).

2.7.5 Text Processing

When processing text several NLP techniques such as encoding, cleaning, tokenization, misspelling correction, POS tagging (part of speech), removing stop words, and stemming/lemmatization will be used. This process will help to convert unstructured data to structured data (Arumugam & Shanmugamani, 2018).

2.7.6 Test representation

Even though, text processing converting the unstructured data to structured data it is still need some efficient methods such as sentiment Analysis, text classification, and further methods to represent the text data. The main necessity, for representation is that the meaning of a word is determined by the words that commonly occur in its neighbourhood. Discreate representation and distributed representation are the most common representations of text (Arumugam & Shanmugamani, 2018).

2.7.7 Model Training

Modelling involves reviewing common algorithms and models that must apply to generate expected out comes from the research/real world problem. NLP techniques such as text classification, sentiment analysis, topic modelling, and deep learning will be used in this purpose (Arumugam & Shanmugamani, 2018).

2.7.8 Model Evaluation

Before using the trained model in management analytics, it must first be reviewed to make sure it is sufficiently generalizable to the corpus. These assessment reviews help researchers select the best model for their study situations (Arumugam & Shanmugamani, 2018).

2.7.9 Tokenization

Tokenization is a text preprocessing method used in NLP and which is the fundamental technique for most NLP tasks. Tokenization refers to the act of dividing a phrase, sentence, paragraph or even a whole text document into components, like individual words or phrases (Brownlee, 2019). These smaller units are referred to as tokens (Singh, 2023). Tokenization could be perform with Python’s Split() function, Python Libraries: NLTK, Regular Expression (RegEx). An open source Library such as SpaCy is also used to tokenization, which is much faster than other libraries. Keras, one of the hottest deep learning libraries in the industry today, which could also be used to tokenization (Brownlee, 2019).

2.7.10 Lemmatization

Lemmatization is a method used in NLP aims to simplify words to their root form, which is referred to as the lemma (Brownlee, 2019). Lemmatization's purpose is to regularise words so that different transformed forms or variations of a word are considered as a single, common form. For example, "running" has the lemma "run," while "better" has the lemma "good"(Brownlee, 2019). Lemmatization is essential in NLP applications including text mining, information retrieval, and machine learning, where understanding the underlying meaning of words is critical (Manning et al., 2008).

2.7.11 Stemming

Stemming is commonly used in NLP applications such as information retrieval, text mining, and search engines. It is especially valuable when the purpose is to capture the real meaning of a term while ignoring its grammatical or semantic complexities. While stemming can result in imprecise stems, it is a computationally efficient means of dealing with word form variations (Jurafsky & Martin, 2020).

2.8 Applications of NLP

Machine translation, language modelling, text creation, sentiment/emotion analysis, natural language comprehension, and question answering are the significant tasks where current NLP techniques have demonstrated great gains. Due to the advancement of NLP techniques and methodologies several NLP jobs now perform at a human-level or even beyond human level. Reporting on some of these accomplishments, NLP establishes a helpful resource for industry and researchers on innovative human language technology (Montejo-Ráez et al., 2022).

A study conducted to research NLP in all areas of management by (Yue Kang, 2020) indicates that each study field regularly uses analytical techniques, categorization, topic modelling, and sentiment analysis. They suggest scholars can investigate themes, similarities, and contrasts across documents, as well as the sentimental orientation of texts, using these conventional techniques. Deep learning, a sophisticated NLP technique for completely mining text, has only been used in marketing and information systems research, nevertheless. Nearly all studies in the disciplines of information systems, operations management, and marketing use existing management theories as their theoretical framework to explain their research model and support their theoretical influences.

Text preprocessing, text representation, model training, and model evaluation are the four main components of NLP (See Figure 3). The main objective, behind text preprocessing is to generate a text by removing symbols, checking for spelling mistakes, breaking it down into tokens, assigning POS tags, eliminating common words, and reducing words to their root forms. This process aims to enhance the effectiveness and accuracy of analysis. When researchers preprocess text, they need to determine how to represent words in a way that computers can process (Yue Kang, 2020). To achieve these words are converted into representations, in the form of vectors or matrices. Thereafter these word vectors and algorithms may be used, including topic extraction, sentiment analysis, and categorization to train a model to find solution for a real-world problem (Yue Kang, 2020).

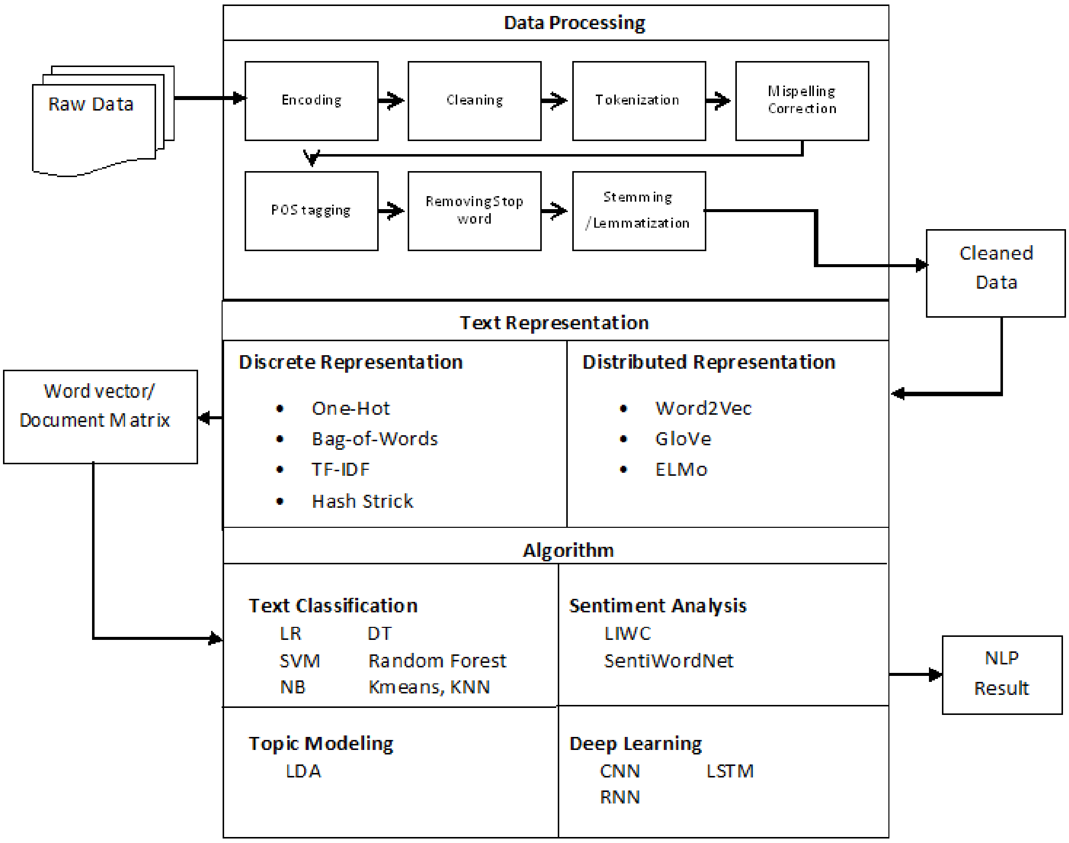


Figure 3: Floor Chart of NLP

2.8.1 Name Entity Recognition

In activities related to NLP it is essential to process and retrieve information from both structured and unstructured data. Extracting insights from vast amounts of data can be challenging, which is why there is a need, for innovative technologies to handle such large-scale data. Various fields of information extraction and NLP often require the implementation of processing techniques to analyse the textual structure in terms of vocabulary, morphology, phonetics, syntax and semantics. Name Entity Recognition (NER), is one such NLP technique that used to extract information from unstructured text data and process of locating nouns that are referenced throughout a passage of text, a phrase, or a paragraph, such as persons, places, organizations, etc. NER is essential for a variety of natural language applications, including automatic text summarization, machine translation, information retrieval, and question answering (Goyal et al., 2018).

The idea of named entity extraction was initially put up in 1996 at the Sixth Message Understanding Conference (MUC-6). Since then, several academics have created a variety of strategies for extracting diversity of entities from various textual genres and languages. However, the research community is still increasingly interested in creating fresh methods for extracting distinct named entities that are useful in a range of natural language applications (Goyal et al., 2018).

A named entity is a word form that recognizes components from a collection that have comparable qualities. It goes by names depending on the context such as a rigid designator, an elemental component, or a part of a specific category (Goyal et al., 2018). For instance, in the field of Biomedicine entities that're important include genes and gene products. In general scenarios entities, like individuals, locations organizations, numbers, dates, and times are relevant. Likewise in the realm of homeopathy we recognize medication and illness names as entities.

Named Entity Recognition and Classification (NERC) is an essential task of information extraction is to detect and categorize members of rigid designators from data suitable to various sorts of named entities such as organizations, individuals, locations, and so on. With the advent of MUC-6, the idea of named entity was born. Named entities played an integral role in achieving the conference's main goal by extracting ENAMEX (person, location, organization) and NUMEX (time, currency, and percentage expressions) objects from structured information about company activities as well as unstructured text of military messages (Goyal et al., 2018). Following that, various scientific events and other conferences contributed significantly to the appearance of NER. From that point onwards, Named Entity Recognition has grown into a interesting field of study.

Up to date, distinct entities have been recognized in various languages and domains using various methodologies. Older systems relied on handmade rule-based algorithms that produced better results for confined domains only, while newer systems commonly depend on machine learning-based algorithms that addresses the limitations of rule-based systems. There are several aspects that might affect NERC performance, including textual genres, entity types, language, and more (Goyal et al., 2018). A NERC system designed for one domain is difficult to transfer to another. Certain languages or domains have limited resources, making the NERC process difficult. The enhancement in tag set enhances the system's complication. Additional rules or traits must be identified to recognize a greater number of entities.

The work of Named Entity Recognition and Classification contains numerous challenges that make it difficult, such as embedded entities, lack of clarity in the text, readiness of resources, and so on. These problems must be addressed carefully for Named Entity Recognition Systems to be resilient. Nested entities are entities inside other named entities, which is challenging to identify. According to the researchers, Segment labelling is a solution to this issue. Text is ambiguous when it occurs as a named entity in one place and a common noun in another, or when it refers to various entity kinds. To address this issue, the named entity disambiguation task gives a system the ability to infer whether a chunk is a named entity or not. A big, annotated dataset (corpus) as well as gazetteers are excellent resources for developing and assessing the performance of NERC systems. However, certain languages, such as Arabic, Mongolian, Indonesian, and Indian languages such as Hindi, Punjabi, Bengali, and Urdu, are resource-poor, making the NERC process more difficult (Goyal et al., 2018).

2.8.2 Sentiment Analysis

Sentiment analysis is a task that aims to identify differences and understand the emotions/sentiments associated with an individual, topic or situation (Kastrati et al., 2021). Typically, the goal of sentiment analysis is to locate users’ opinions, identify the sentiments they convey, and then classify their polarity into positive, negative, and neutral categories. Sentiment analysis systems combine NLP and ML approaches to extract, store, and filter facts and opinions from massive volumes of textual data (Kastrati et al., 2021). When considering the related work, sentiment analysis has been vastly used in handful fields, including corporate, social networks and education sector.

Sentiment analysis may generally be done at three separate levels: the document level, the sentence level, and the aspect level. When analysing the content sentiment analysis, at the document level aims to comprehend the sentiments of users. When evaluating the overall document, sentence level analysis is more accurate as it focuses on determining the polarity of individual phrases. Aspect level sentiment analysis on the hand strives to categorize user opinions regarding specific elements or qualities mentioned in reviews.

Sentiment analysis has been greatly used in the domain of business and social media platforms. Goods and service assessments, commercial markets, customer engagement management, and marketing approaches and research are the commonly used business applications for sentiment analysis. The most typical use of sentiment analysis in social network applications is to track a brand's rating on Twitter or Facebook and investigate how people respond to crises, such as COVID-19.

2.8.3 Keyword Extraction

Keyword extraction is a method used in text analysis to identify and extract the most commonly used and significant words and phrases from a document. It assists in summarizing text content and identifying the major problems discussed. Keyword extraction is a technique that merges AI and NLP to identify words from different types of content, like research papers, business reports, social media comments, online forums, reviews and news articles. Keyword extraction is the most crucial part of text analysing. Since keywords are simple to construct, edit, remember, and distribute, they are commonly utilized to facilitate inquiries inside Information Retrieval (IR) systems. When comparing scientific signatures, corpus independent alternatives provide more versatility and can be applied to various corpora and information retrieval (IR) systems. Additionally, keywords have proven useful in improving the performance of IR systems. To put it simply relevant keywords that are extracted can be used to generate an index for a collection of documents or serve as representations, for categorization or classification tasks (Biswas et al., 2018).

Beliga has conducted research to give an overview of methodologies and approaches for keyword extraction tasks and to elaborate method systematization (Beliga, 2014). Furthermore, the report compiles a complete assessment of existing research, related work on keyword extraction for supervised and unsupervised approaches, with a specific emphasis on graph-based methods and Croatian keyword extraction (Beliga, 2014). Keyword assignment and keyword extraction are two methods of selecting the best keyword. In keyword assignment the selection of keywords is based on a predetermined list of terms or a predefined taxonomy, while in keyword extraction, words are analysed to identify the most representative ones. Keyword extraction does not rely on an existing thesaurus to identify the relevant keywords (Beliga, 2014).

Existing methods of key word extraction can be divided into four methods according to the categorization anticipated by Zahang et al (2008): these include simple statistics, linguistics, machine learning and other methods (refer to figure 4). Simple statistics approaches are techniques that don't rely on specific training data and can be applied across different languages and domains. They involve analysing aspects of a document, like word frequency, n-gram statistics, TFIDF (Term Frequency Inverse Document Frequency) word co-occurrences and PAT Tree, among others. However, in professional texts there might be instances where the key keyword is mentioned only once in the article. This could lead to filtering of these crucial words by statistically empowered models. Linguistic approaches primarily make use of the linguistic features of words, phrases, and documents. Analysing language involves types of analysis such, as examining the words used the structure of sentences, their meaning and how they contribute to the overall conversation.

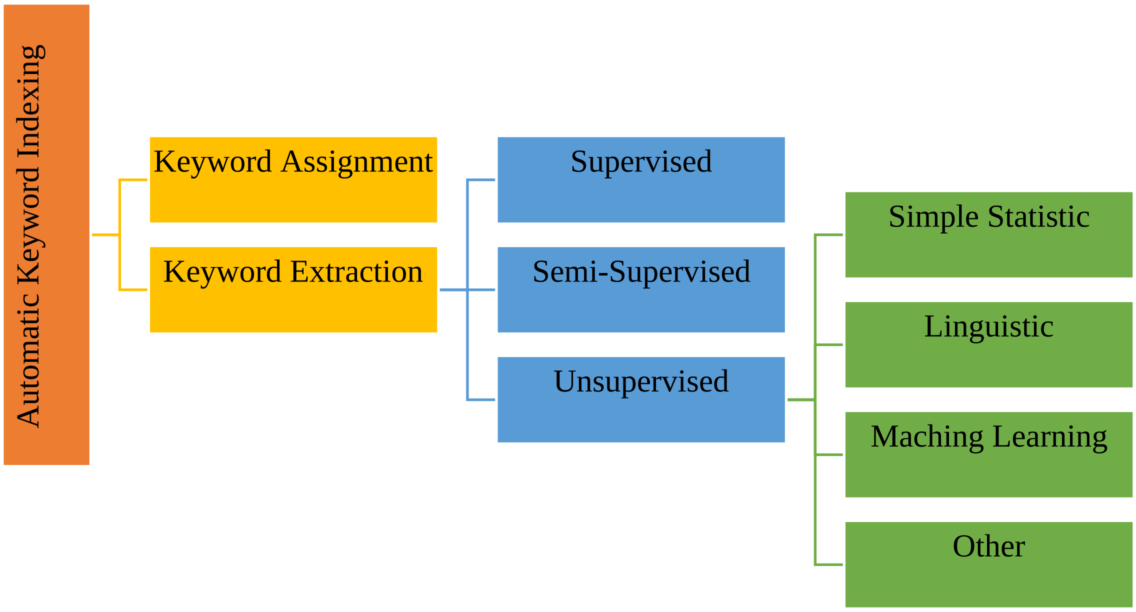


Figure 4: Classification of Keyword Extraction Methods

Machine Learning methods explore both supervised and unsupervised learning methods for keyword extraction. However, previous research highlight preference for supervised approach (Biswas et al., 2018). Supervised machine learning methods involve training a model using a set of keywords that are manually labelled in the training dataset. Examples of these methods include Naïve Bayes, SVM, C4.5 and Bagging among others. Inducing the model can be quite challenging and time consuming when dealing with datasets as the system needs to re learn and establish the model whenever there is a change, in the domain. Other methods for keyword extraction in overall integrate all the strategies discussed above. Moreover, for combination, they sometimes include investigational knowledge, such as the location, length, layout properties of the phrases, html and related elements, text formatting, and so on.

2.8.4 Text Classification

Machine learning algorithms utilize sample data, known as "training data “, to construct a model that enables them to make predictions or decisions without explicit programming (Razno, 2019). Machine learning algorithms are classified into five types: supervised, semi-supervised, active learning, reinforcement, and unsupervised learning. NLP tasks usually include voice recognition, natural language comprehension, and natural language production (Razno, 2019).

Text classification is a common and significant task in supervised machine learning. Assigning categories to documents, which can be a web page, library book, media item, gallery, or anything else, has several uses such as spam screening, email routing, sentiment analysis, and so on (Razno, 2019). To accomplish text classification popular machine learning and NLP tools, such as Pandas, Scikit-learn, NumPy, and a little bit of NLTK could be used (Razno, 2019).

Over the past few decades, text classification concerns have been deeply explored and handled in an array of variety practical applications. With the recent innovations in NLP and text mining, several researchers are increasingly interested in building more and more applications that use text categorization algorithms. Most of the text classification and document categorization systems may be broken down into four stages: feature extraction, dimension reductions, classifier selection, and assessments as indicated in Figure 5 (Kowsari et al., 2019).

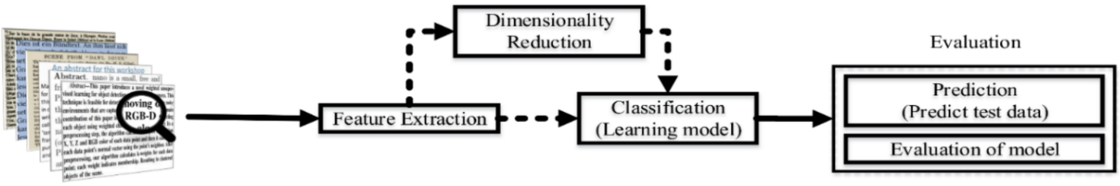


Figure 5: Overview of Text Classification Pipeline (Kowsari et al., 2019)

The initial pipeline input is raw text data collection. In general, text data sets comprise text sequences in documents like D=X_1, X_2,..., X_n, where X_i refers to a data point (i.e., document, text segment) with s sentences, each of which has w_s words with l_w letters (Kowsari et al., 2019). Every point is labelled with a class value from a set of $_k$ distinct discrete value indices. Ultimately, to train effectively it is necessary to create a curated dataset known as Feature Extraction. The process of reducing dimensionality is a step that can be incorporated alongside the classification system. Deciding most accurate and relevant classification method is the crucial step in document classification. The pipeline also includes an evaluation process, which is separated into two portions (prediction the test set and evaluating the model). In general, the text classification method may be implemented at four distinct levels of scope: Document Level, paragraph level, sentence level and sub-sentence level.

In the research community it is crucial to have comparable performance measures to assess algorithms. However, these measures are often available for a limited number of methods. One major challenge in evaluating text classification methods is the lack of protocols for data collection. Moreover, comparing performance measures, across experiments can be tricky since these measures typically focus on specific aspects of classification task performance.

2.8.5 Text Summarization

Text summarization in NLP is the process of breaking down lengthy text into a meaning full paragraph or sentence. This process extracts the important piece of information out of the large text, while conserving the meaning of the original text. Text summarization has been a subject of research and academia. To provide succinct summaries, many models have been developed and tested on various datasets. They were compared by using various comparison ratings. Text summarization can be query-based or general, extractive (EXT) or abstractive (ABS), single document or multi documents (Rahul et al., 2020).

EXT text summarization creates summaries using the same identical phrases as the original text. ABS is more broad-based and concentrates on the document's main ideas. While single document summaries text of a single document, multi documents summaries text of several documents. Additionally, text summarization based on queries is nowadays widely use. Whereas generic summaries are often ones that focus on the overall region of the text input, question-based summarization models produce summaries of the text depending on a specific area as indicated by the query given by the user.

In the research conducted by Rahul et al. (Rahul et al., 2020), the accuracy of the different NLP based machine learning approaches for text summarizations has been evaluated. The common methods they were found from previous five years were Machine learning (ML), neural networks (NNs), reinforcement learning, sequence to sequence modelling, and fuzzy logic. Similar to this, a variety of optimization techniques have been employed to improve the suggested goal function for text summarizing. They observer that when employ several approaches against the same dataset and that their accuracy results are varied. Also, some researchers have merged the various approaches and discovered that the summaries are more accurate when using combined methods than using a single method. Python libraries like scikit learn, nltk, spacy, and fastai have been employed when NLP processing has been used as a method to summarize text documents.

2.8.6 Topic Modelling

Topic Modelling is one of the powerful text mining techniques and it is a method for automatically determining the topics included in a text object and for determining any hidden patterns displayed by a corpus of texts.

The rule-based text mining methods that employ regular expressions or dictionary-based keyword searching methods are different from topic modelling. It is an unsupervised method for locating and observing groups of words (referred to as "to[pics") in big groups of texts. Topic can be repeating pastern of co-occurring items of a corpus. Figure 6 illustrates the process of topic modelling simply.

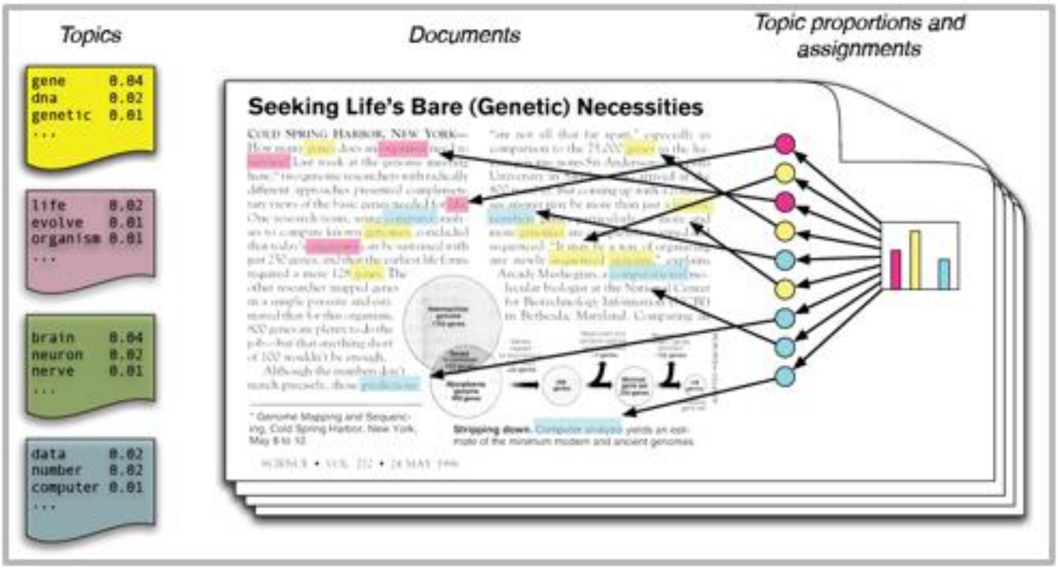


Figure 6: Process of topic modeling (Vidya, 2020)

Topic models are particularly helpful to organize enormous blocks of textual data, cluster documents, retrieve information from unstructured text, and feature selections. To extract hidden information from job descriptions and match them to the appropriate individuals, many experts in the recruiting industry use topic models. Large datasets of emails, customer reviews, and user social media profiles are all organized using them.

To find the presence of "strategic topics" and further connect them to the enterprises under consideration, the management discussion and analysis sections of the annual reports of the top publicly traded Indian construction contracting firms are examined by (Jagannathan et al., 2022). According to the study, the construction sector is facing unprecedented times due to the Coronavirus pandemic, and top management decision-making is essential to ensure its value. Even though, data is essential for informed decision making, but the sector is often criticized for its lack of a research-ready database. Researchers need to use basic data collection techniques such as questionnaire surveys and interviews to prepare themselves with data to analyse and convey research discoveries easily. The construction sector is available with abundant unstructured text data such as annual reports and financial statements, which are compulsory disclosures in the public domain. NLP-based topic modelling algorithms are being used to extract keywords and topics from publicly accessible annual reports of construction contracting firms, allowing firms to analyse their strategies to deal with emerging sectoral challenges. In this work, a qualitative content analysis is carried out to find the latent themes while using NLP based topic modelling algorithms such as Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF). LDA is a statistical model based on Bayesian inference. It is assumed that each document is a mixture of a specific number of themes, and that each word in the corpus is probabilistically connected with each of the supplied topics. In contrast, NMF is a linear algebraic model. While NMF and topic modelling share the same goal, NMF is a matrix factorization and multivariate analytic approach that creates coefficients (rather than probability) for each word while mapping it to a particular subject. This research focuses on the 49 'lean signs identified by Roy D and Jagannathan M (Roy and Jagannathan 2021). Compares them to previous non-NLP studies based on keywords. The findings reveal that both LDA and NMF based models accurately identify four, out of the five companies with the highest count of these 'lean signs. In the context and data considered for this research NMF outperforms LDA. This study also demonstrates how NLP based topic modelling algorithms can be used to analyse and group keywords aiding researchers in visualizing strategies employed by construction contracting firms. Additionally, the NMF algorithm holds promise in enhancing text document accuracy while simplifying the reading, comprehension, and summarization of text data.

2.9 Deep Learning in NLP

Deep learning is a class of machine learning algorithm in artificial intelligence that trains computers to process data in the way of human's brains (Deng & Liu, 2018). Deep learning models can recognize complex patterns in images, text, sound and other data to provide exact analysis and forecasts. Figure 7 outlining the NLP process from raw data to results, including data cleaning, representation techniques, and algorithmic applications.

In the book of "Deep Learning in Natural Language processing" by (Deng & Liu, 2018), author describes evolution of NLP as three waves: rationalism, empiricism, and deep learning. The first wave elaborated in the basis of the presumption that linguistic knowledge in the human mind is predetermined by generic inheritance, rationalist methods promoted the building of handmade rules to incorporate knowledge into NLP systems. The second wave of empirical techniques makes the assumption that the mind can acquire the complicated structure of natural language with or without the need for rich sensory input and visible linguistic facts in surface form (Deng & Liu, 2018). To extract the regularities of languages from vast corpora, probabilistic models were created. In the third wave, deep learning uses hierarchical nonlinear processing models that are modelled after biological neural networks to learn essential illustrations from linguistic input in ways that intention to replicate human cognitive capacities.

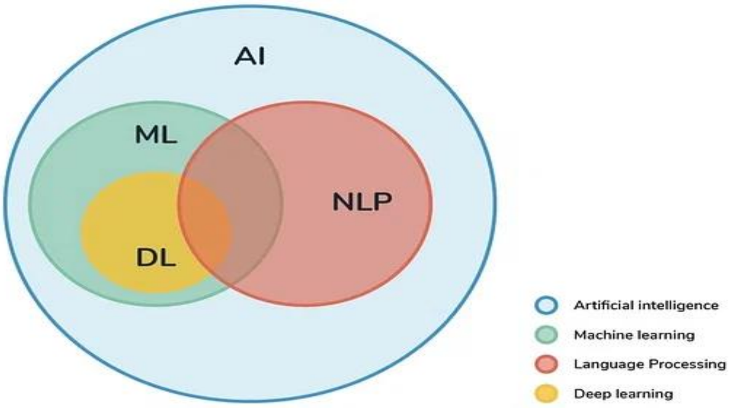


Figure 7: Representation of Deep Learning in the Domain of AI

Deep learning and NLP have combined to produce impressive accomplishments in real-world applications. The first industrial NLP application that deep learning has significantly influenced is speech recognition. Deep neural networks obtained noticeably reduced recognition errors than the conventional empirical techniques because to the availability of large-scale training data. Machine translation is another well-known and effective deep learning NLP application. Translation quality has been shown to be significantly improved by end-to-end neural machine translation, which uses neural networks to represent the mapping between human languages. Hence, significant commercial online translation services provided by major technological firms, such as Google, Microsoft, Facebook, Baidu, and others, have soon adopted neural machine translation as the new de facto technique. Deep learning has made progress in various domains of NLP including but not limited to language comprehension, dialogue systems, lexical analysis and parsing knowledge representation, information retrieval, text-based question answering, social computing, language generation and sentiment analysis. At present deep learning is widely considered as the approach, for nearly all NLP tasks.

2.10 NLP with Transformers

Transformers have established a genuine perspective for solving wide range of NLP problems in both scholarly realm and business since their inception in 2017 (Tunstall et al., 2022). Many people interact with a transformer today without even realizing it: BERT is currently used by Google to improve its search engine by improved comprehending users' search queries. Similarly, OpenAI's GPT family of models has gained mainstream media headlines for its capacity to create human-like writing and imagery (Tunstall et al., 2022). These transformers now fuel applications like GitHub's Copilot, which can translate a remark into source code that generates a neural network automatically (Tunstall et al., 2022). This design has proven to be more effective, than recurrent neural networks (RNNs) when it comes to machine translation tasks both in terms of the cost of training and the quality of translation (Tunstall et al., 2022).

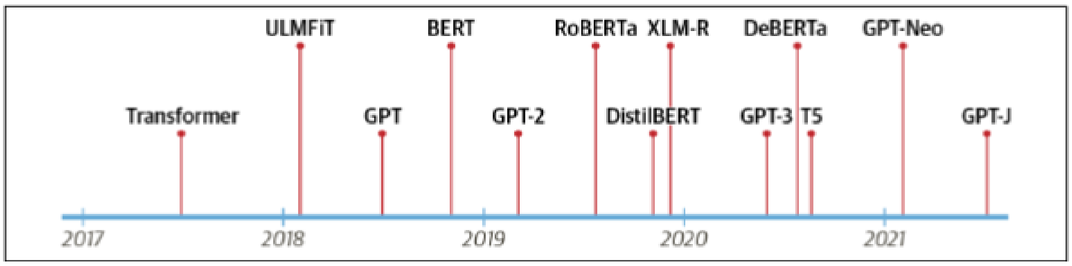


Figure 8: The transformers timeline (Tunstall et al,2022.)

Simultaneously, an efficient transfer learning approach known as ULMFiT demonstrated that training long short-term memory (LSTM) networks on an extremely extensive and diverse datasets may provide cutting-edge text classifiers with minimal labelled input (Tunstall et al., 2022). These advancements set the stage for two of the popular transformers we know today; the Generative Pretrained Transformer (GPT) and Bidirectional Encoder Representations from Transformers (BERT). By combining the Transformer design with learning these models eliminated the need to build task specific architectures, from scratch and surpassed almost all NLP benchmarks by a significant margin (Tunstall et al., 2022). Figure 8 represents a zoo of transformer types that have arisen since the release of GPT and BERT.

Transformer architectures have made it possible to create powerful models and through pretraining these models can be effectively utilized for various tasks. Transformers is an accessible library that aims to make these advancements accessible to a wider community of machine learning practitioners. The collection is made up of meticulously developed advanced Transformer architectures united by a common API. This library is supported by a curated set of pretrained models created by and available to the community (Wolf et al., 2020).

Research on resume categorization was conducted by (Changmao Li, 2020) to considerably decrease the time and labour required to filter an irresistible quantity of applications while enhancing the collection of acceptable candidates. A total of 6,492 resumes were retrieved from 24,933 job submissions for 252 Clinical Research Coordinator vacancies classified into four levels of experience (CRC) (Changmao Li, 2020). Each resume is manually annotated to its most relevant CRC position by experts using numerous rounds of triple annotation to set criteria. As a result, the agreement between annotators shows a Kappa score of 61%. Using this dataset, they built classification models based on transformers for two tasks; the first task involves analysing a resume and categorizing it based on CRC level (T1) while the second task involves analysing both a resume and a job description to determine if the application is suitable for the position (T2) (Changmao Li, 2020). The performing models, which utilize section encoding and multi head attention decoding techniques achieve an accuracy of 73.3% for T1 and 79.2%, for T2.Their investigation demonstrates that the majority of prediction mistakes occur between nearby CRC levels, which are difficult for even specialists to discern, emphasizing the practical relevance of models in real-world HR systems (Changmao Li, 2020). According to the author, previous research in this area has concentrated on categorizing resumes or job descriptions into work-related groups (e.g., data scientist, healthcare provider). Nevertheless, no work has been discovered to differentiate resumes based on degrees of skill.

2.11 Machine Learning Approaches

This section explores cutting-edge methods for skill extraction through Machine Learning (ML) techniques, including NLP and deep learning networks.

In 2020, Akshay Gugnani and Hemant Misra introduced a comprehensive framework (Gugnani & Misra, 2020) for skills extraction that leveraged various NLP techniques. The framework consisted of four primary submodules:

Named Entity Recognition (NER): NER, typically used for recognizing keywords and concepts, played a pivotal role in extracting skills from Job Descriptions (JDs). The researchers harnessed NER, employing Watson NLU 3 services, to extract skills as entities. Subsequently, the extracted skills were categorized as "Probable Skills." These processed skills were then assigned relevance scores using Word2Vec.

Part of Speech (PoS) Tagger: Part of Speech Tagging involves labelling each word in the text with its corresponding part of speech. Group of five industry experts manually analysed a subset of JDs, labelling words or phrases as skills in this context. It became evident that the definition of skills was influenced not only by the job industry but also by the subjective perspective of the labeller. To address this, a broader dataset of JDs underwent processing through the Stanford Core NLP Parser and PoS Tagging. The observations from this process led to the formulation of rules and patterns for identifying potential skill terms that may not be found in skill dictionaries or taxonomies. For instance, if a sentence contained a comma-separated list of nouns, and some of those nouns were skills, then it was likely that the other nouns were skills as well. The system was programmed with similar rules for skill-term identification.

Word2Vec (W2V): Word2Vec, a technique for representing words as vectors, was employed to create vector representations of words. Typically, this involved using a large text corpus to generate a multi-dimensional vector space. Each unique word was mapped to a corresponding vector within this space, with words in similar contexts positioned near each other. While single-word skills were relatively straightforward to extract, challenges arose when dealing with skill phrases, such as "Hard Working" or "Web Development." To address this challenge, the researchers proposed representing a skill phrase as a vector, the average of individual vectors composing the skill phrase. Additionally, a skill dictionary was utilized to compare each potential skill term in the embedded W2V space. The model was further enhanced through user feedback mechanisms. The model's training data comprised a text corpus of 1.1 million JDs from over 50 diverse categories.

Skill Dictionary: A Skill Dictionary was essential for identifying words or phrases as skills. To create this dictionary, the researchers followed an approach previously proposed by Gugnani and colleagues. Skills were extracted from publicly available resources such as Onet, Hope, and Wikipedia. These terms were then rigorously validated by a team of three experts. The resulting Skill Dictionary encompassed 53,293 soft and hard skills across different categories.

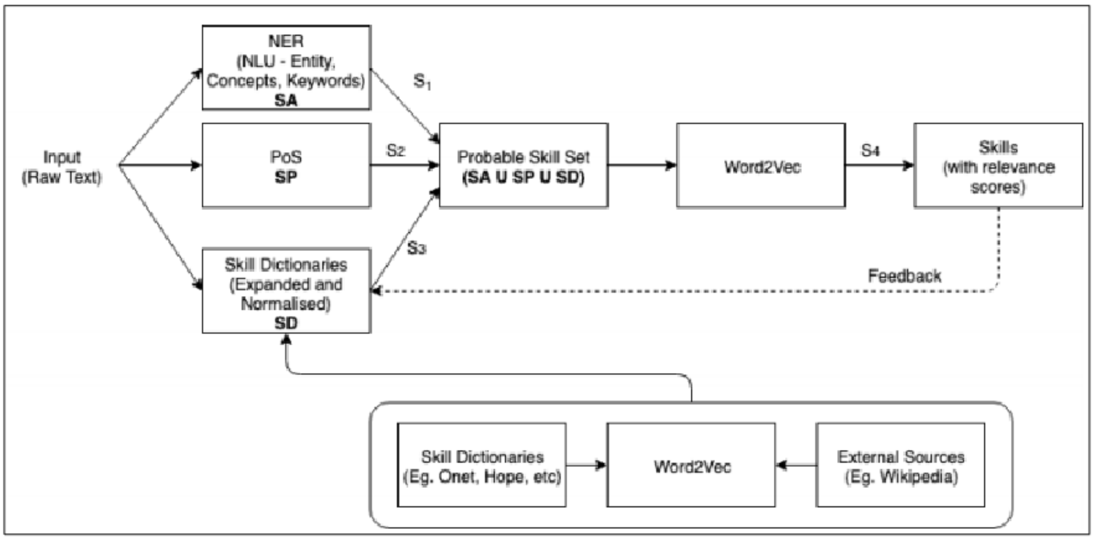


Figure 9: Skill Identification Flow Diagram (Gugnani & Misra, 2020)

This framework provided a robust and multifaceted approach to skill extraction, combining the power of NER, Part of Speech Tagging, Word2Vec, and a meticulously curated Skill Dictionary to deliver accurate and comprehensive results (refer figure 9).

In 2019, Nikita Sharma conducted a comparative analysis (Sharma, 2018) of diverse approaches, utilizing both unsupervised and self-supervised learning techniques to extract pertinent skills from unstructured text. The models were initially trained on a modest dataset comprising job advertisements within the Data Science category and were subsequently adapted for use across different domains. Two primary models were examined on that approach:

Topic Modelling: This unsupervised technique aims to identify abstract topics within the text. While it displayed a robust grasp of context, the extracted keywords occasionally failed to align precisely with the relevant set of skills defined in the problem statement.

Word2Vec: As a self-supervised neural network, Word2Vec excels in recognizing words used in similar contexts. In this study, Word2Vec was employed as an extension of the Topic Modelling method. The keywords extracted by Topic Modelling were employed to train the Word2Vec model, which proved proficient at skill recognition. Nevertheless, it is worth noting that Word2Vec also introduced some extraneous noise, necessitating careful separation of valuable skills from the noise.

The subsequent two models in this study employed supervised learning. The training dataset was meticulously annotated, primarily encompassing noun phrases as skills. Below, we provide a detailed comparison of these two approaches:

- Word Embedding-Based Classifier with Convolutional Layer: The first model utilized a straightforward approach, incorporating a convolutional layer, and was trained on the labelled dataset. This model successfully extracted numerous relevant skills from the job descriptions, achieving a test accuracy of 0.6803.
- Combination of Word Embedding and Long Short-Term Memory (LSTM): The second model combined word embedding with Long Short-Term Memory (LSTM), resulting in an enhanced skills classifier with broader keyword extraction capabilities. Notably, this approach exhibited the most promising outcomes, boasting a test accuracy of 0.7658. Moreover, it managed to reduce the presence of noise when compared to other models.

Despite being trained on a notably small dataset, the LSTM and Word Embedding model demonstrated commendable results. However, it is imperative to recognize that the training data primarily consisted of noun phrases, whereas many job postings articulate required skills using verb phrases and other grammatical structures. Consequently, an imperative step forward involves expanding the initial dataset with a more diverse set of labelled examples to facilitate the development of a new model.

2.11.1 Skills Folksonomy/Taxonomy and Graph-Based Approaches

A common approach to skills extraction involves the utilization of methods that centre around the construction of either a "folksonomy" or a "taxonomy" of entities (du Preez, 2015). While both terms pertain to controlled dictionaries, they exhibit distinct characteristics, as highlighted in Table 1:

Table 1: The comparison of taxonomy and folksonomy

Characteristic	Taxonomy	Folksonomy
Hierarchy	Hierarchical	Flat
Relationships	Parent/child & sibling	No levels, no order, no explicit relationship
Exclusivity	Exclusive	Not Exclusive
Establishment	Top-down (Established by experts)	Bottom-up (Created by users)

ESCO, which stands for the " classification of European Skills, Competences, Qualifications and Occupations " is a notable skills extraction system based on skills taxonomy principles (du Preez, 2015). Its main purpose is to serve as a dictionary that promotes interoperability, among labour markets, educational systems, and training programs. Regrettably, detailed information regarding the techniques and methodologies employed to construct the ESCO taxonomies is not readily available. Nonetheless, it is worth noting that this publicly accessible resource can be a valuable asset for constructing taxonomies in the context of this research (du Preez, 2015).

LinkedIn stands as the global leader among job search systems in today's landscape. In 2014, the LinkedIn team created a formidable skills extraction framework. This framework was crafted atop an extensive skills and expertise folksonomy, which is a pivotal component for a recommender system (Bastian et al., 2014). Building the folksonomy included key phases of discovery, disambiguation, and deduplication.

The initial discovery stage was grounded in the observation that many users curated lists of skills in the specialty section of their profiles, typically separated by commas (Bastian et al., 2014). This user-generated data became a valuable resource for identifying potential skills. The disambiguation phase aimed to alleviate uncertainties surrounding skill phrases with multiple meanings contingent on their context. To address this, the system implemented clustering techniques that relied on the co-occurrence of phrases. This clustering approach effectively resolved issues associated with the disambiguation of skills (Bastian et al., 2014). Certain skills showed associations, with senses each connected to groups often labelled with an industry tag. When it came to eliminating duplicates that had meanings, such as "Python development" and "Python programming," their primary goal was deduplication. To tackle this challenge, they opted for a crowdsourcing method involving LinkedIn users. They requested users to link a skill phrase, with the Wikipedia page from a list of suggestions. This collaborative effort successfully resolved the problem of duplicates.

Moreover, the skills extraction system included a component for inferring skills. By using profile attributes such, as company, title and industry as features a Naïve Bayes classifier was trained on the collected set of features (Bastian et al., 2014). However further research efforts revealed that a graph-based model outperformed the Naïve Bayes classifier highlighting the evolving nature of skills extraction methods (Bastian et al., 2014).

The approach proposed by Kivimäki et al. (Kivimäki et al., 2013) for automated skills extraction from unstructured text documents was novel and hinged on leveraging the hyperlink graph of Wikipedia in conjunction with a skills folksonomy derived from LinkedIn. This system's core concept revolved around computing relationships between an input document and the textual content within Wikipedia pages. Subsequently, the system employed the Spreading Activation algorithm on the Wikipedia graph to establish associations between the input document and skills (Kivimäki et al., 2013).

This approach demonstrated the capability to extract skills from text, encompassing both explicitly stated skills and those that were inferred from the content. It underscored the synergy between external knowledge sources, such as Wikipedia and LinkedIn, and the potential for automated skills extraction from textual data.

Wang et al. (Wang et al., 2014) introduced an approach akin to that of Kivimäki et al., capitalizing on a skills folksonomy sourced from LinkedIn and a graph-based model. This model drew upon textual data available within the skills and expertise sections, personal profile connections (including shared majors, titles, companies, and universities), and skills connections (i.e., skills that co-occurred together).

Notably, the approach that was founded on skills connections showcased superior performance when compared to the one relying solely on profile connections. However, the most compelling outcomes were achieved by the hybrid system that effectively harnessed both connection types. This mutual approach demonstrated the most favourable results, underscoring the potency of integrating multiple connection types for skills extraction (Wang et al., 2014).

In the research endeavour by Hoang et al. (Hoang et al., 2018), a novel approach to skill taxonomy generation was unveiled. This work, much like the methodology employed by LinkedIn (Bastian et al., 2014) encompassed the crucial stages of discovery, disambiguation, and deduplication. However, the implementation of these stages was approached differently.

To construct the taxonomy system, the researchers collected skill-related content from candidate resumes and job descriptions, which were accessible on the Career Builder website. The collected textual data was methodically segmented at punctuation marks and underwent a cleansing process to eliminate noise, including stop words, superfluous adverbs, and other predefined terms based on domain expertise. Essentially, this cleaning phase sought to remove words that contributed minimal or no semantic value to the eventual skill taxonomy.

For normalization and deduplication, researchers leveraged the Wikipedia API. An integral step in building the skill taxonomy involved validation, wherein the Standard Occupational Classification (SOC) system was employed to corroborate the Wikipedia category tags assigned.

In addressing the challenge of word sense disambiguation (WSD), the Google Search API was enlisted. For instance, when a skill term had multiple senses, the system selected the one with the highest Google Search relevancy ranking. However, it's worth noting that this approach displayed a limitation by not considering semantic context. As a response to this, the researchers introduced the Skill Tagging system, which is elaborated in Figure 10. This skill taxonomy encompassed a substantial collection of 39,000 raw terms mapped to 26,000 normalized skill entities, enhancing the understanding and classification of skills in a structured manner.

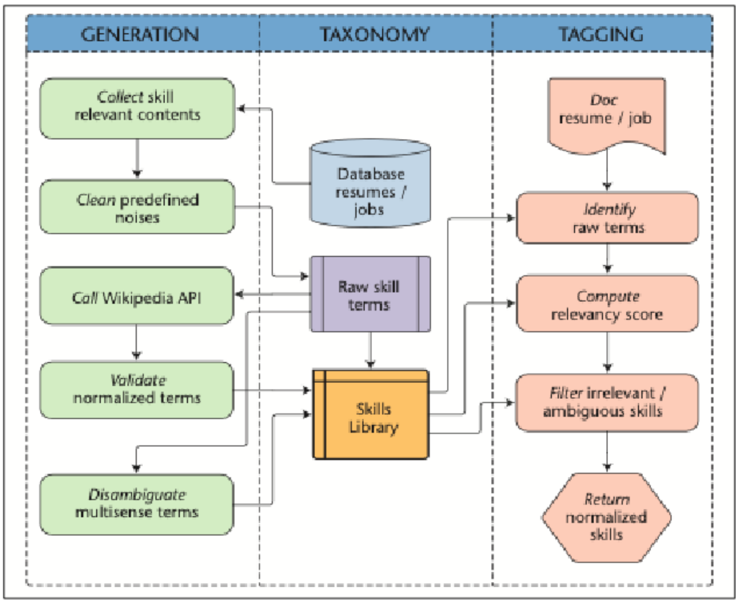


Figure 10: Architecture of the SKILL System (Hoang et al., 2018)

2.11.2 Limitations for context and Summary of existing approaches

Numerous methods exist for extracting skills from unstructured text, but this thesis focuses exclusively on the most advanced and successful approaches discussed earlier. This thesis aims to advance this field while considering the specific requirements and limitations within the Banking and Finance sector Job Descriptions (JDs).

Creating a skills folksonomy or taxonomy is a collaborative effort involving experts and users, addressing deduplication and disambiguation challenges. Regrettably, this thesis could not involve additional individuals in this endeavour due to resource constraints.

Furthermore, the chosen Machine Learning (ML) approaches are based on supervised learning, necessitating labelled datasets for model training. This labelling process is both time-consuming and resource intensive. Consequently, the developed model must be trained on a relatively modest dataset while achieving satisfactory results.

In addition to these challenges, the availability of computational resources is of utmost importance. Analysing extensive textual data and employing advanced ML methods like BERT often demands robust GPUs or advanced multicore CPU machines. Consequently, the performance of a single laptop may be inadequate for training and effectively testing the skills extraction model.

Based on the described limitations and the specific context of this study, various approaches for skills extraction are analysed. The conclusions and suitability of each approach are summarized in Table 2 below:

Table 2:Summary of Existing Skill Extraction Approaches

Approach Name	Conclusion
Skills Folksonomy/Taxonomy and Graph-Based Approaches	
Bastian et al. LinkedIn, LinkedIn Skills: Large-Scale Topic Extraction and Inference (Bastian et al., 2014)	Not suitable for this study due to several factors: ● The dataset consists of free-form CVs and JDs, making it challenging to construct a template for entity extraction. ● The model relies on a crowd-assisted approach to eliminate semantic duplicates, which is not feasible in this study. ● It uses a Bayesian classifier, considered an outdated approach.
Kivimäki et al. A Graph-Based Approach to Skill Extraction from Text (Kivimäki et al., 2013)	The Wikipedia graph-based model used to associate input documents with skills might be applicable.
Wang et al. Skill Inference with Personal and Skill Connections (Wang et al., 2014)	This system, similar to Kivimäki et al. (Kivimäki et al., 2013), applies user profile data to enhance performance. However, it's not usable in this research as access to Banking and Finance sector data is unavailable.

Hoang et al. Large-Scale Occupational Skills Normalization for Online Recruitment (Hoang et al., 2018)	This approach has limitations, primarily due to its failure to consider semantic context. Additionally, its implementation is complex and resource-intensive, making it impractical for this study.
ML Approaches	
Sharma. Job Skills Extraction with LSTM and Word Embeddings (Sharma, 2018)	This approach, a combination of word embedding and LSTM techniques, offers promising results in skills extraction from unstructured text. It's relatively straightforward to implement and may be suitable for this study. However, it is advisable to consider newer NLP methods like BERT to replace the LSTM component.
Gugnani et al. Implicit Skills Extraction Using Document Embedding and Its Use in Job Recommendation (Gugnani & Misra, 2020)	One of the modules employs NER to identify keywords and extract entities as skills. This approach can be valuable for this thesis work.

In summary, while specific traditional approaches are unsuitable for this study due to dataset constraints or outdated methodologies, ML approaches, such as the combination of word embedding and LSTM or NER-based techniques, present more viable options for skills extraction within the context of this research. When selecting the most appropriate method, it is essential to consider the specific needs and limitations of the Banking and Finance sector JDs.

The construction of a skill folksonomy or taxonomy is an intricate and labour-intensive endeavour, and it lies outside the primary focus of this research. Additionally, pre-existing dictionaries primarily contain skills relevant to common job descriptions. In contrast, the central objective of this study is to develop a skills extraction system explicitly tailored for the Banking and Finance sector job descriptions.

In this context, the most suitable approach for this research aligns with the model developed by Nikita Sharma (Sharma, 2018), where Word embedding, and LSTM techniques are combined. This model enables training a neural network on a small dataset while delivering outstanding results. This master's thesis presents a similar approach but with a more contemporary NLP technique, BERT, to enhance the NLP component.

Moreover, it is worth noting that the challenge of skills extraction from unstructured text can be viewed through POS tagging data. This perspective has been explored in the work of Gugnani et al. (Gugnani & Misra, 2020) and Chernova, M (Chernova, 2020), further underscoring the relevance of NER techniques and POS tagging in this research.

Chapter 3: Data Collection and Design

3.1 Introduction

This study takes a mixed methods approach, integrating qualitative and quantitative techniques. A complete literature review, data collecting, framework creation, and validation are all part of the research strategy. The approach involves a sequential investigation, with the quantitative phase (which includes data collection and framework creation) coming before the qualitative phase (which focuses on framework validation).

The first step involves conducting an extensive literature review to combine current information and insights related to NLP methods, artificial intelligence, transformers, and skill identification in the banking sector. This requires a thorough investigation of academic publications, conference proceedings, and industrial reports.

Web-based annotation tools will be used to prepare a comprehensive data set of job postings in the banking sector. These tools will be implemented in Python and will follow for efficient and accurate extraction of skills mentioned in the job postings. Random sampling technique will be exercised to select job postings from various Banks. This will represent job roles such as banking operations, financial analysis, risk management, and customer service. Text classification techniques of NLP will be used for data cleansing and summarization. Finally, qualitative interviews will be conducted with Banking sector professionals to obtain deeper insights into the critical skills required of employees in this sector.

Descriptive statistical analysis will be conducted on the data collected from the web-based annotation tools to identify the frequency and distribution of skills mentioned in the job postings. This analysis will provide quantitative insights into the most critical skills required in the banking sector. This will provide high-accuracy qualitative insights into the skills required and their relative measures.

3.2 Data Collection

The data collection of this research primarily involved collecting online job postings from Indeed UK, one of the largest job portals in the United Kingdom. According to online sources, Indeed is the #1 job site globally, with over 300 million unique visitors every month. In the data collection, Indeed UK provided a rich and diverse dataset of job advertisements from the banking sector, encompassing various job roles and skill requirements. The job postings were retrieved using a web scraping tool developed using Python. The relevant textual data included job title, company, job description and qualifications. The data collection from Indeed UK was extended by gathering additional job postings from the career pages of the top Banks in the United Kingdom. Those included HSBC UK, Barclays Bank PLC, NatWest Group, etc. This additional step ensured a comprehensive dataset encompassing general banking industry postings and specific job opportunities advertised directly by banks. Combining the above datasets formed the foundation for subsequent analysis and development of the skill identification framework.

A robust web scraping tool was designed using Python and Selenium to extract valuable job information from Indeed, a popular job search platform (See Appendix A). This tool dynamically navigates through job listings, overcoming obstacles such as pop-ups and dynamic elements, ensuring a comprehensive collection of data. Leveraging ChromeDriver and the Selenium library (Lubanovic, 2019), the scraper maximizes efficiency and accuracy in retrieving job titles, descriptions, company names, and locations. Through meticulous design, the tool adapts to the dynamic nature of the website, handling interruptions gracefully to provide a seamless and uninterrupted scraping process. The resulting dataset, stored in a CSV file, serves as a valuable resource for analysing and understanding the landscape of banking jobs, offering insights into job titles, requirements, and geographical distributions within the industry. Table 4 indicates the summary of data collection.

Table 3: Summary of Data Collection

Time Frame	
Start Date	July 2023
End Date	August 2023
Frequency	
1. Data was collected 4 times using the web scrapping tool during the above period from Indeed UK website.	

2. During month of July 223 job advertisements were extracted from career pages of UK Banks directly.	
Scope	
Web Scraping from Indeed: Job Listings: Information related to job titles, descriptions, locations, and other relevant details available on the job search section of the Indeed website. Utilized web scraping tool developed using Python to systematically retrieve and organize job-related data. **Direct Copying from Bank Career Pages:** Job details were manually extracted by copying information directly from the career pages, including job titles, descriptions, requirements, and company/bank name.	
Number of Records/Size of Data	
Number of job postings collected prior to remove Duplications	4857
Number of Job postings after removing duplicates and non-banking advertisements	2962

3.3 Text Processing

Text processing plays a pivotal role in this research, preliminary extracting and analysing text data obtained from job advertisements. NLP approaches will be used to negotiate the complexities of unstructured data, allowing to identify vital information and to categorize. The objective of this excursion was not only to peruse job descriptions but also to construct a description that clarified the difficult terminology employed in this particular domain of employment. The approach transformed unprocessed textual data into a rich source of knowledge by effectively combining data science skills with the linguistic capabilities of NLP.

Initially, a data-cleaning framework was created to structure and streamline the data environment. In addition to the job criteria, extraneous columns were elegantly removed, allowing the primary characteristics - "Job Title," "Company Name," "Job Description," and "Location" - to become the focal point. Data minimization facilitated the readability of the employment tableau by eliminating unnecessary details and maintaining a focused narrative (See Appendix B).

After eliminating duplicates, a dataset featuring unique job descriptions emerged. Subsequently, attention shifted towards text preprocessing, a harmonious amalgamation of techniques to enhance the content's quality. In information technology, lemmatization evolved as a powerful language tool, simplifying words to their essential forms. This technique effectively grouped terms with common linguistic origins, establishing the foundation for advanced and comprehensive research.

The process of expanding contractions introduced a conversational rhythm to the dataset, converting "you'll" to the more formal "you will" and unravelling the condensed complexities of "we're" into the more extensive "we are." The linguistic unfolding increased the level of formality and improved the interpretability of the dataset. At the same time, the entire body underwent a transition where all letters were changed to lowercase, promoting uniformity, and eliminating unforeseen variations in letter cases.

A coding procedure was executed to eliminate special characters, non-alphanumeric parts, and punctuation marks. These undesired parts were methodically eradicated by employing regex, a potent pattern-matching tool, leading to a more efficient and concentrated dataset. Furthermore, every text was transformed to lowercase, guaranteeing consistency and facilitating later analysis.

This procedure laid the foundation for the crucial step of tokenization. The process of tokenization involved breaking down the text into its basic units, which are individual words. Now existing as separate entities, each word established the basis for subsequent examination. This exposed the inherent linguistic intricacies incorporated in job descriptions, providing a more distinct and detailed perspective of the data. As fundamental components, the tokenized descriptions were ready for further processing to delve into insights and enhance comprehension.

In text analysis, common English words such as "the," "and," and "is" are often removed. These are known as "stopwords" and typically do not contribute much to the text's overall meaning. By eliminating them, focus can be placed on the more significant words. This is particularly beneficial in specialized fields like banking, where specific vocabulary is frequently used. The removal of stopwords allows for a closer examination of the text, revealing the unique language used in the banking sector. This method enhances the analysis by emphasizing the core content of the text.

From a technical perspective, removing stopwords is a standard step in text preprocessing for NLP. *Stopwords* are words that are filtered out before or after text processing. When constructing an NLP model's vocabulary, it is often beneficial to exclude these stopwords. This reduces the complexity of the model and allows it to focus on the words that carry more meaning. This highlights the industry-specific language used in banking or any other professional field.

The text-processing steps have resulted in a refined dataset, demonstrating the effective combination of linguistic processing and data science techniques. This enhanced dataset goes beyond simple job descriptions, serving as a detailed representation of the language complexities within the banking sector. It signifies the transformation from raw text to a processed dataset, capturing the subtleties of language and the specific vocabulary of the banking sector.

Upon completing the text processing phase, the refined dataset is more than just a collection of words. It is a well-structured representation of linguistic patterns. This processed text is evidence of the careful effort put into understanding the complexities of language within the banking sector. It serves as a platform where applying NLP techniques has revealed unique patterns and subtleties embedded in the text.

The text processing phase represents a conscious effort to unlock the hidden potential within textual data. It acts as a bridge connecting the unstructured world of language with the structured domain of data science, setting the stage for further analysis. The refined dataset is a launchpad for deriving actionable insights illustrating the skills, requirements, and complexities of banking jobs.

In conclusion, the text processing phase is not just a technical task but a transformative process that adds value to textual data. The dataset has been shaped, refined, and customized to reveal language subtleties through NLP techniques, laying the groundwork for further analysis. This phase is a critical part of the broader process of unravelling the complexities of the banking job landscape. This narrative's fusion of linguistic subtlety and data science precision unfolds.

3.4 Job Description and Job Requirement Extraction

In the process of skill extraction, various NLP techniques are utilized to analyse the preprocessed text. Essential techniques such as Named Entity Recognition (NER) and Part-of-Speech (POS) tagging are employed to identify skill-related entities or phrases within the text. The goal is to pinpoint and extract the skills crucial to the banking sector.

In addition to these techniques, pre-trained language models like BERT, Word2Vec, and GloVe are used to understand skills' semantic nuances and contextual significance. With a vast amount of linguistic knowledge, these models allow for a thorough interpretation of the complex relationships between words and their context within professional skills. By incorporating these advanced language models, skill extraction achieves greater accuracy and depth, improving the ability to identify and categorize skills found in job descriptions. The combination of NLP techniques and pre-trained language models forms a powerful strategy for uncovering the skill landscape embedded in the requirements of banking sector jobs.

3.5 Job Requirements Categorization and Taxonomy Development

In the process of creating a Job Requirements taxonomy for the banking sector, skills extracted from job descriptions are systematically grouped into relevant categories and subcategories. This is achieved through advanced techniques, transforming the Job Requirements into a comprehensive skill taxonomy.

This taxonomy's basis is formed by applying clustering and topic modelling techniques. These methods, driven by semantic similarities and co-occurrence patterns within the skills, enable an automatic categorization process. The clustering technique identifies connections and patterns among the Job Requirements, grouping related skills into distinct categories. This approach creates a structured taxonomy and reveals the relationships and dependencies among various Job Requirements in the banking sector.

Topic modelling techniques are also used, adding depth to the categorization process. These techniques uncover latent topics and themes within the skills, contributing to creating meaningful subcategories and further refining the taxonomy. The insights gained from topic modelling provide a detailed understanding of skill interdependencies, ensuring that the categorization reflects the complex landscape of professional Job Requirements within the banking sector.

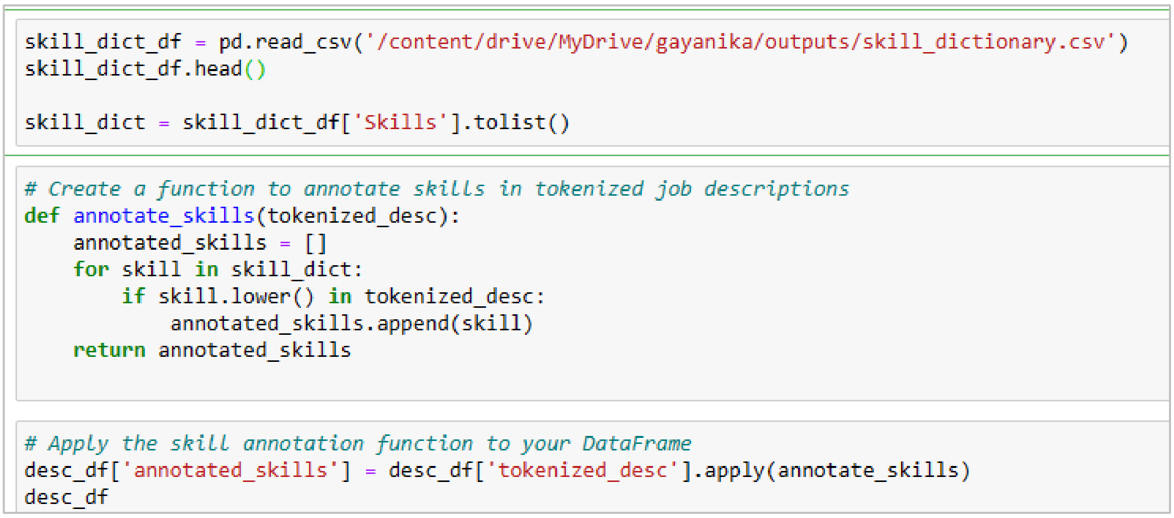
Combining clustering and topic modelling techniques is crucial in creating a robust skill taxonomy for the banking sector. This taxonomy, enriched by semantic relationships and organizational structures, is a valuable tool for understanding and navigating the diverse skill requirements in the banking industry.

3.6 Application of Skill Dictionary

The development of a comprehensive skill dictionary is essential for mapping skills inside job descriptions. This process involved wide range of sources, including online skill dictionaries and research literature. The Occupational Information Network (O*NET) database for skills have been used as a significant reference (Popov et al., 2022). O*NET is the primary source of occupational competence information in the United States.

It is a skills taxonomy that includes assessments of skills, abilities, work activities, training, and job characteristics for almost 1,000 distinct jobs. O*NET is extensively utilized in the United States and overseas, and its comprehensiveness (it covers the whole labour market) made it a logical starting point for the Skill and Productivity Board. Furthermore, O*NET is continuously updated and open source, making it freely accessible (Popov et al., 2022). The skill dictionary generation process concludes in the actual application of the mapping dictionary to annotate talents inside tokenized job descriptions. The skill dictionary (See Appendix C), which was precisely compiled as described above, serves as a complete reference for finding relevant skills within the tokenized job descriptions. The python code snippet in figure 11 demonstrate the skill dictionary was used.

The skill dictionary is used accurately in this implementation to indicate talents within tokenized job descriptions. The annotate_skills method repeats over tokenized job descriptions, finding skills based on skill dictionary entries. The annotated skills are then appended to a new column in the DataFrame (annotated_skills), demonstrating the skill dictionary's direct applicability to real-world data.



```
skill_dict_df = pd.read_csv('/content/drive/MyDrive/gayanika/outputs/skill_dictionary.csv')
skill_dict_df.head()

skill_dict = skill_dict_df['Skills'].tolist()
```

```
# Create a function to annotate skills in tokenized job descriptions
def annotate_skills(tokenized_desc):
    annotated_skills = []
    for skill in skill_dict:
        if skill.lower() in tokenized_desc:
            annotated_skills.append(skill)
    return annotated_skills
```

```
# Apply the skill annotation function to your DataFrame
desc_df['annotated_skills'] = desc_df['tokenized_desc'].apply(annotate_skills)
desc_df
```

Figure 11: Code for mapping skill dictionary.

This section highlights the practical relationship between the skill dictionary's creation and its actual use in the annotation process, which is an important step in preparing the data for further analysis and modelling.

3.7 Model Development and Evaluation Process

The data collection process for this study involved several key stages contributing to the overall objective of predicting skills based on job postings. Initially, job posting data was gathered, encompassing details such as job titles, company names, job descriptions, locations, and required skills. This data was organized in a tabular format, typically loaded into a Pandas DataFrame, a widely used data structure for manipulation and analysis in Python.

The next stage was the analysis of the collected data. Descriptive statistics were computed to provide insights into the data. These statistics included the count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum for a column labelled "No." This step helped in understanding the distribution and variability of the data.

The third stage involved implementing a Support Vector Machine (SVC) model. SVC is a type of machine learning model that is commonly used for classification tasks. In this case, it was used to predict the skills required for different job roles based on the job posting data. The model was implemented using a linear kernel, one of the most common kernels used in SVC.

After the model was trained, it was evaluated using various metrics. These metrics included precision, recall, and the F1-score computed for each skill predicted by the model. The model's overall accuracy was also reported. These metrics provided a comprehensive view of the model's performance.

The final stage involved the interpretation of the model's output. The output was an array of predicted skills for the given data, with each element corresponding to a set of skills predicted by the model for a particular job posting. The model's performance was further analysed by examining the precision, recall, and F1 scores for different skills. Skills with high precision and recall indicated that the model performed well for these skills. However, these metrics were undefined for some skills, likely because there were no predicted samples for those skills.

Throughout the research process, any warning messages encountered during the model evaluation were addressed and considered in interpreting the model's performance. This comprehensive methodology ensured a thorough analysis of the job posting data and the practical implementation and evaluation of the machine learning model. The resulting skill predictions provide valuable insights into the skills required for various job roles in the dataset.

3.8 Skill Importance and Frequency Analysis

In preparing a machine learning model for job classification, a comprehensive analysis of skill importance and frequency was conducted. The analysis utilized a dataset containing job-related information, including job titles, company names, job descriptions, locations, and associated skills. The dataset was organized into three parts for ease of interpretation.

The initial part of the analysis involved examining the dataset's structure and content. The dataset comprised various columns, such as 'No,' 'Job Title,' 'Company Name,' 'Job

Description,' 'Location,' and 'Skills.' In particular, the 'Skills' column contained comma-separated lists of skills associated with each job listing.

Subsequently, data preprocessing steps were undertaken to facilitate the training of a machine learning model. The 'Skills' column was transformed to represent individual skills, enabling a more granular analysis. The skills were then consolidated into a single list, and a frequency analysis was performed to identify the most commonly occurring skills across the entire dataset.

Simultaneously, a Support Vector Classification (SVC) model was trained on the 'Job Description' and 'Skills' columns using the scikit-learn library. The trained model was evaluated for accuracy, and classification reports were generated to assess precision, recall, and F1-score metrics for each skill category.

An analysis of feature importance was conducted to gain insights into the importance of individual skills in the model's predictions. The 'CountVectorizer' from sci-kit-learn was employed to convert textual data into a numerical format suitable for model training. The coefficients of the linear SVM model were then examined to identify the most influential features, i.e., words representing skills.

The final step involved visualizing the results of both the frequency analysis and feature importance analysis. The top N most frequent skills were plotted to provide a clear overview of prevalent skills in the dataset. The importance of the top N features (words) was also visualized, shedding light on the key factors driving the model's predictions.

This comprehensive skill analysis approach offers valuable insights for talent acquisition and workforce planning stakeholders. It provides a nuanced understanding of the skills landscape within the dataset, aiding in identifying critical skills and informing strategic decision-making processes.

Chapter 4: Data Analysis

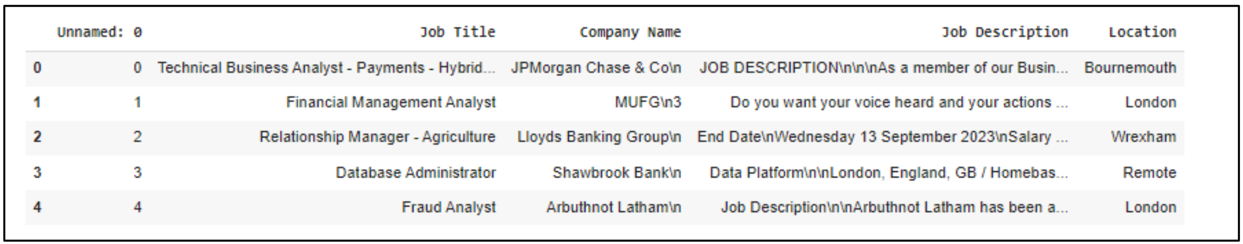
4.1 Introduction

Data analysis is a step-by-step procedure where the gathering, tidy up, modify, and create models using data. The main objective is to find valuable insights, draw conclusions, and help make informed decisions. The process entails utilizing approaches, methodologies, algorithms and systems to acquire understanding and valuable information from both organized and unorganized data.

As the crucial point of the study, this chapter embarks on a complete exploration of the identified skills, offering a detailed examination of their frequency, patterns, and contextual degrees. Employing advanced data mining techniques, the analysis aims to resolve valuable insights into the dynamic landscape of skill requirements. The chapter displays with an accurate examination of the length and structure of job descriptions, recognizing their pivotal role in shaping the outcomes of skill extraction. Subsequently, the discussion navigates through the most frequent job titles, shedding light on their prevalence and significance. The overall goal of is to provide a proper understanding of the skill landscape of the banking sector, laying the foundation for practical suggestions.

4.2 Structure of the Data Set

The primary dataset of this research is the job vacancy details scrapped from job posting websites. The dataset presented in Figure 12 is a structured table extracted from a job posting websites which includes information about various job positions.



	Unnamed: 0	Job Title	Company Name	Job Description	Location
0	0	Technical Business Analyst - Payments - Hybrid...	JPMorgan Chase & Co\n	JOB DESCRIPTION\n\n\nAs a member of our Busin...	Bournemouth
1	1	Financial Management Analyst	MUFG\n3	Do you want your voice heard and your actions ...	London
2	2	Relationship Manager - Agriculture	Lloyds Banking Group\n	End Date\nWednesday 13 September 2023\nSalary ...	Wrexham
3	3	Database Administrator	Shawbrook Bank\n	Data Platform\n\nLondon, England, GB / Homebas...	Remote
4	4	Fraud Analyst	Arbuthnot Latham\n	Job Description\n\nArbuthnot Latham has been a...	London

Figure 12: Description Data Representation in Python Data frame

Given below the explanation of the dataset, column by column:

Job Title: The job title column lists the available job positions. The titles are tailored to each role and sometimes provide extra details about the job. For example, a title like "Technical Business Analyst - Payments - Hybrid..." suggests that the role might involve working with payment systems and could offer the flexibility of hybrid work arrangements.

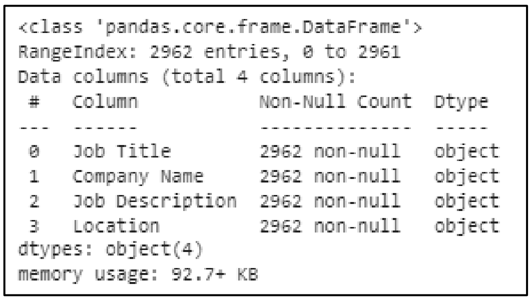
Company Name: The company name refers to the names of the companies offering the positions. For instance, while observing "Lloyds Banking Group" and "Deutsche Bank" it suggests that these positions belong to the financial sector.

Job Description: The job description contains information in detail about the roles and responsibilities of the person. Further, what type of skills are needed and how the relevant skills are being implemented while performing a job role.

Location: The location refers to the location of the job on the map. The locations can range from specific cities like "London" to more general descriptions like "Remote." This means the job does not require the employee to be physically present in a specific office.

Every row in this table represents a different job listing. The first column gives each entry a unique identifier, like 0, 1, 2, etc. There are different job titles available, ranging from technical positions such as "Technical Business Analyst" and "Database Administrator" to financial positions like "Financial Management Analyst" and "Fraud Analyst".

The Figure 13 provides statistic information about a DataFrame object in pandas, a widely used Python library for data manipulation (Brownlee, 2019). A data frame is a type of data structure that is two-dimensional and can be changed in size. It stores tabular data that may have different types of values and has labelled rows and columns. When analyse the provided statistics, the non-null count for each column is 2962, indicating 2962 entries without missing values.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2962 entries, 0 to 2961
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Job Title        2962 non-null   object
 1   Company Name     2962 non-null   object
 2   Job Description  2962 non-null   object
 3   Location         2962 non-null   object
dtypes: object(4)
memory usage: 92.7+ KB
```

Figure 13: Representation of Statistics of Dataframe in Jupyter note book

The term "Dtype" is commonly used in pandas to represent text data, specifically strings. Given that all the columns have the Dtype 'object', it suggests that the data in these columns are all in string format (Brownlee, 2019). This aligns with the nature of the described data, which includes job titles, company names, job descriptions, and locations. The next section illustrates an analysis of job posting data on different attributes.

4.2.1 Location Wise Distribution of Job Advertisements

The bar graph shown in Figure 14, generated using the Python library matplotlib. Typically, this kind of graph is used to display the distribution of categorical data (Kuhlman, 2011). The x-axis represents various categories, precisely the "Locations" where jobs are posted or available. The y-axis displays the "Counts," which most likely represents the number of job listings or occurrences for each location in the dataset.

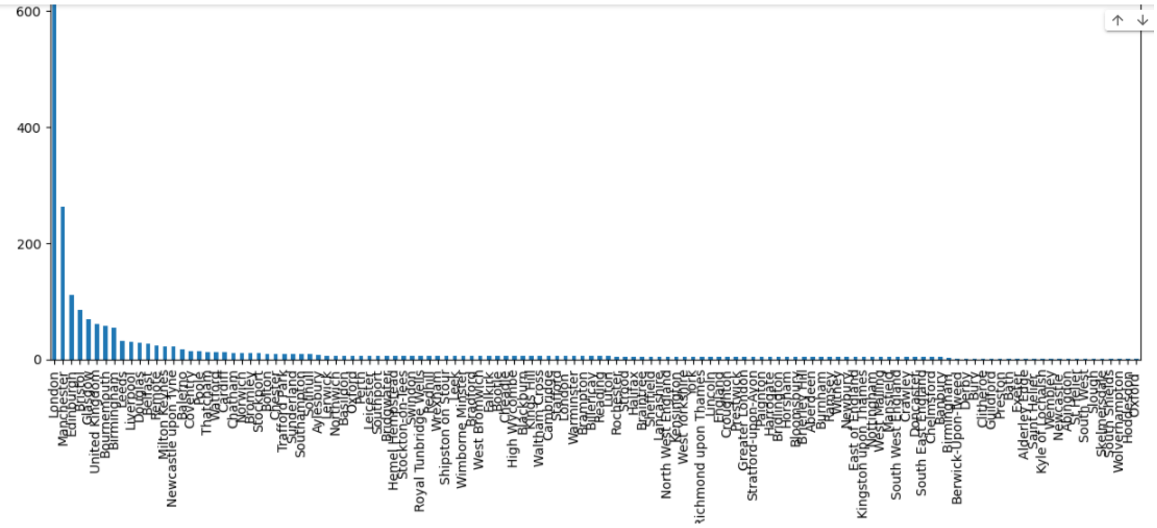
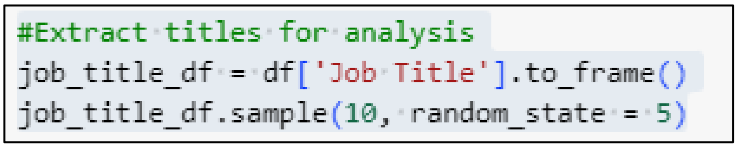


Figure 14: Location wise Job Posting Distribution

The labels on the x-axis, which represent the names of the locations, are closely positioned, and tend to overlap, making it challenging to read them. It is common for bar charts that have many categories to exhibit these behaviours. The graph provides a detailed breakdown of job postings across 128 different locations. London emerges as the top location, boasting the highest number of job advertisements, approximately 1550 out of 2962 total job listings in the dataset. Following London, other notable locations include Manchester, with nearly 300 job advertisements, and Edinburgh, with around 100. Bristol, Glasgow, Birmingham, Liverpool, Leeds, Bournemouth, Newcastle, and Milton Keynes fall within the 100-50 range of advertisements. Remarkably, all other locations feature less than 30 advertisements each. This distribution highlights the concentration of job possibilities in specific areas, with London standing out as a notable hub for job postings.

4.2.2 Job Title Analysis

As the next step the code snippet in figure 15 accomplishes randomly selecting a sample of job titles from the dataset for analysis.

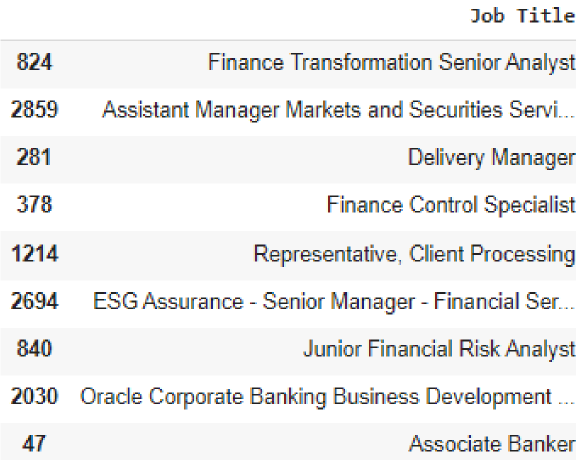


```python
#Extract titles for analysis
job_title_df = df['Job Title'].to_frame()
job_title_df.sample(10, random_state = 5)
```

Figure 15: Python Code for extract random sample of job titles

After creating a new DataFrame called "job_title_df", which contains only the "Job Title" column from the original DataFrame "df". To create a new DataFrame called "job_title_df", extract the 'Job Title' column from an existing DataFrame called "df". Then the use of .to_frame() method to convert the Series object returned by df['Job Title'] into a DataFrame. This DataFrame will provide a tabular representation of the data. People often do this for convenience when they need to manipulate or analyse data and prefer to use a DataFrame instead of a Series.

The line of code `job_title_df.sample (10, random_state=5)` randomly selects 10 rows from the `job_title_df` DataFrame using the `.sample()` method. The `random_state=5` parameter ensures that the same random sample is selected each time the code is run. To ensure reproducibility, the random_state parameter is set to a fixed number of 5. When a random state



	Job Title
824	Finance Transformation Senior Analyst
2859	Assistant Manager Markets and Securities Servi...
281	Delivery Manager
378	Finance Control Specialist
1214	Representative, Client Processing
2694	ESG Assurance - Senior Manager - Financial Ser...
840	Junior Financial Risk Analyst
2030	Oracle Corporate Banking Business Development ...
47	Associate Banker

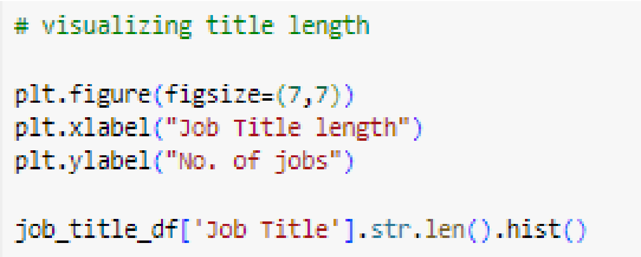
Figure 16: Random Sample of Job Titles

is specified, the code will consistently sample the same set of rows every time it is run. This is important for maintaining consistency in analysis and sharing the code with others.

The above figure 16 includes the sampled rows. Each row is accompanied by its corresponding job title and an index. The index on the left side shows where each row was initially located in the complete dataset. The 'Job Title' column displays the titles of the jobs that were sampled. This sample can be utilized for conducting exploratory analysis, which involves examining the variety of job titles, identifying patterns or trends, or simply inspecting the data before proceeding with a more comprehensive analysis.

4.2.3 Interpretation of Job Title Length Distribution from a Histogram

Histograms are extremely useful in data analysis because they provide valuable insights into the distributions of variables, help identify outliers, and allow for a comprehensive exploration of data characteristics. Regarding to this study, examining the lengths of job titles could offer valuable information about the complexity of the titles in the dataset. Additionally, this analysis could assist in improving the user interface design for displaying these titles. Figure 17 illustrates the python code used to represent the said histogram.



```
# visualizing title length

plt.figure(figsize=(7,7))
plt.xlabel("Job Title length")
plt.ylabel("No. of jobs")

job_title_df['Job Title'].str.len().hist()
```

Figure 17: Code to retrieve job title length histogram

Based on the above code the histogram generated given in figure 18 and it seems that job titles typically fall within the range of around 20 to 30 characters in length. This is evident from the tallest bar in the graph. This implies that most job titles are brief and to the point.

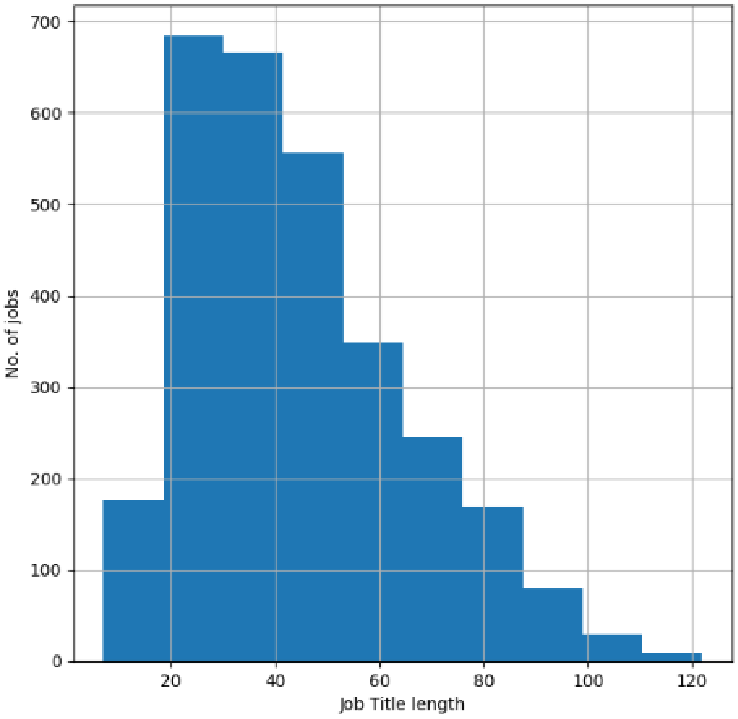


Figure 18: Job title string length distribution

As the length of job titles exceeds 30 characters, their frequency tends to decrease. The decrease in job titles is not a straight line. There are some fluctuations, which suggest that job title lengths vary, but overall, there is a trend of having fewer job titles as their length increases.

A distinct tail on the right side is visible. This indicates that the majority of job titles are on the shorter side, but there is a smaller group of job titles that are quite long, with some even reaching up to 120 characters.

The histogram gives us a visual representation of the lengths of job titles, indicating that they generally vary from short to moderately long. Businesses can gain valuable insights by examining the concentration of job titles that fall within the 20-to-30-character range. This information can help them determine the ideal length for job titles, ensuring they are clear and impactful. Additionally, it could emphasize the industry's preference for concise job titles. Longer job titles may suggest that the positions are more specialized or complex. However, it is also possible that these longer titles are exceptions and require additional investigation.

4.2.4 Interpretation of Job description Length Distribution from a Histogram

Length of job description is crucial factor when analysing and extracting skills. The main objective of this project is to extracting skills and the skills are to be derived directly from the job descriptions. Figure 19 illustrates the histogram of length of job descriptions.

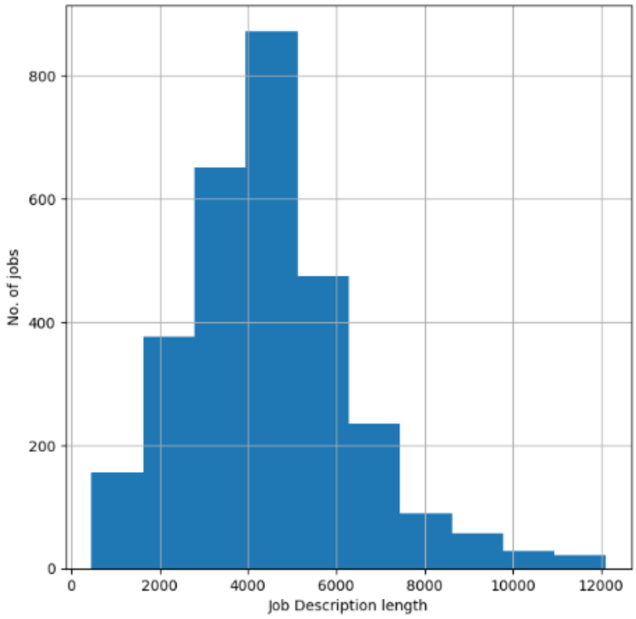


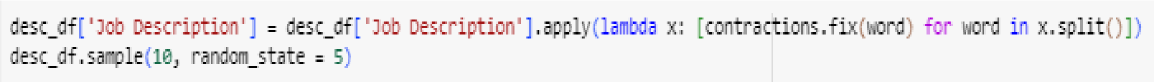
Figure 19: Job description string length distribution

The data shows that the tallest bin is around the 2,000-character mark. This suggests that the most frequently observed job description length is approximately 1,500 to 2,500 characters.

This histogram offers valuable information about the level of detail or briefness in the job descriptions found in the dataset. Employers prefer to keep their job descriptions concise and clear, as indicated by the standard length of around 2,000 characters. Longer descriptions may suggest positions that demand in-depth explanations or specific qualifications. This type of distribution can provide valuable insights to job listing platforms and employers. It helps them understand the average length of job descriptions and decide how much information to include. This knowledge allows them to align with industry standards or choose to stand out by deviating from the norm.

4.3 Text Pre-processing

Text Preprocessing is an important stage in preparing raw textual data for more complex NLP tasks. A crucial element requires attending to language subtleties, including contractions, to improve the accuracy of the analyses that follow. To do this, the code snippet uses the contractions library to extend contractions in the dataset's 'Job Description' column in a methodical manner (see figure 20). In addition to guaranteeing a consistent depiction of words, this careful procedure lays the groundwork for a deeper and more complex examination of the text's underlying ideas. The subsequent sections deconstruct the several processes that were used to select and enhance the text data, providing the groundwork for significant discoveries and efficient machine learning models.

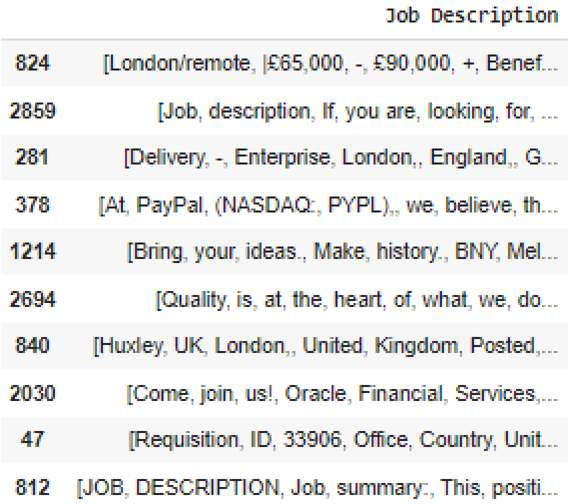


```
desc_df['Job Description'] = desc_df['Job Description'].apply(lambda x: [contractions.fix(word) for word in x.split()])
desc_df.sample(10, random_state = 5)
```

Figure 20: Code for Text Processing

The explanation of the code snippets in above figure 20 is as follows. The 'Job Description' column in the desc_df dataframe is being assigned to itself. The `apply` function is used to apply a lambda function to each element in a list. In this case, the lambda function splits a string `x` into individual words. Then, it uses the `contractions. Fix` function to fix any contractions in each word. Finally, the lambda function returns a list. To access the 'Job Description' column in the desc_df DataFrame, user can use the syntax desc_df['Job Description']. The .apply() method is used to apply a function to the DataFrame along a specified axis. In this scenario, the purpose is to utilize a function that will be applied to every element within the 'Job Description' column.

In Python, the lambda keyword is used to create an anonymous function. In this case, the variable x represents each element or job description in the 'Job Description' column. The function `x.split()` splits the job description text into separate words. The code snippet [contractions. Fix (word) for a word in x.split()] is a list comprehension that applies the contractions .fix function to each word in the job description. The function contractions .fix is commonly employed to expand contracted words by converting them into their complete forms. For example, it can transform "isn't" into "is not".



	Job Description
824	[London/remote, |£65,000, -, £90,000, +, Benef...
2859	[Job, description, If, you are, looking, for, ...
281	[Delivery, -, Enterprise, London,, England,, G...
378	[At, PayPal, (NASDAQ:, PYPL),, we, believe, th...
1214	[Bring, your, ideas., Make, history., BNY, Mel...
2694	[Quality, is, at, the, heart, of, what, we, do...
840	[Huxley, UK, London,, United, Kingdom, Posted,...
2030	[Come, join, us!, Oracle, Financial, Services,...
47	[Requisition, ID, 33906, Office, Country, Unit...
812	[JOB, DESCRIPTION, Job, summary:, This, positi...

Figure 21:Tokenized Job Description

As a result, every job description has been transformed into a list of words where contractions have been expanded. The utilization of the 'sample' function to randomly select ten samples from the dataframe `desc_df`, with a fixed random state of 5. This line of code randomly selects ten rows from the desc_df DataFrame to display a subset of the data. The random_state parameter sets a specific number so that the same sample can be generated again in future runs. This is crucial for ensuring that scientific research and analysis can be reproduced accurately.

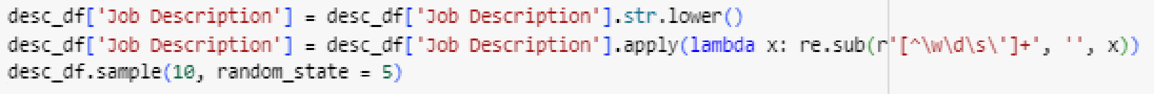
The output displays an image that appears to be a DataFrame. It consists of two columns: an index column and a column labelled 'Job Description' (see figure 21). The job descriptions in the 'Job Description' column have been expanded to include the total words instead of contractions. However, this expansion may take time to be apparent from the displayed snippet. The index numbers (824, 2859, 281, etc.) represent the original row numbers in the complete dataset.

Text preprocessing is vital in preparing data for analyses like NLP, sentiment analysis, or topic modelling. Expanding contractions is a necessary step in the preprocessing process to ensure that text data is consistent and analyzable. After the data has been cleaned, it can be utilized for various text analysis tasks like analysing word frequency, extracting keywords, or training machine learning models.

4.3.1 Normalizing and Cleaning Text Data in Python

In the preprocessing phase of text analysis, using the provided Python code to normalize and clean text data collection. Preparing data for NLP tasks often involves following this common practice.

Below is the explanation of how the code works, breaking it down into individual steps.



```
desc_df['Job Description'] = desc_df['Job Description'].str.lower()
desc_df['Job Description'] = desc_df['Job Description'].apply(lambda x: re.sub(r'[^\w\d\s\']+', '', x))
desc_df.sample(10, random_state = 5)
```

Figure 22: Normalizing and Cleaning Text Data

Using this line of code to import the regex module and give it the alias "re". The regex module enhances the functionality of Python's built-in re-module, enabling more sophisticated and nuanced pattern matching. The job description column in the desc_df dataframe is assigned to the 'Job Description' column. The function `str.lower()` converts a string to a lowercase.

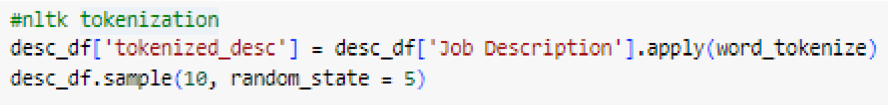
The following line of code is used to convert all the text in the 'Job Description' column of the DataFrame desc_df to lowercase. *Lowercasing* is a common practice used to normalize text and ensure that words like "Project," "Project," and "PROJECT" are treated as identical by the algorithm. The job description column in the desc_df dataframe is assigned the value of desc_df.Job Description one can use the `apply` function along with `lambda` to apply the regular expression `re.sub(r'[^\w\d\s\']+', '', x)` to each element in a dataset. This regular expression removes any characters that are not alphanumeric, whitespace.

The apply method is being utilized to handle each entry in the 'Job Description' column. After using a lambda function that applies the re. sub function from the regex module to replace patterns in the text. The regular expression pattern r'[^\w\d\s\']+' is used to match any character that is not a word character (\w), a digit (\d), a whitespace character (\s), or an apostrophe (\'). The replacement string '' means that the characters that match will be removed from the text. To clean the text, the removal of punctuation and special characters. These elements are usually unnecessary for text analysis and can create unnecessary noise in the data. To randomly select 10 samples from the dataframe `desc_df`, with a random state of 5, you can use the `sample()` function.

Ultimately, we utilize the sampling method to choose 10 rows from the DataFrame randomly. The setting of the random_state to ensure reproducibility. Here are some job descriptions that have been cleaned and are ready for review or further analysis. Regarding data science, it is crucial to perform specific preprocessing steps to create a clean dataset that can be effectively utilized for various NLP tasks. These tasks include tokenization, sentiment analysis, topic modelling, and data feeding into machine learning models. Normalizing and cleaning the text helps reduce the variability in the dataset and enhances the quality of insights that can be obtained from the data.

4.3.2 Tokenization and Stopword Frequency Analysis

Text tokenization is an essential step in decomposing complicated textual material into individual words or tokens, which paves the way for more in-depth linguistic analysis (Monsters, 2022). In this section it explores the nuances of tokenizing job descriptions with the Natural Language Toolkit (nltk). Furthermore, investigated is the frequency analysis of stopwords, which are often occurring yet frequently non-informative words. This code sample (see figure 23) shows how to apply the nltk tokenization technique to the dataset's 'Job Description' column, resulting in the creation of a new column called 'tokenized_desc.' The tokenized version of every job description is shown in this column, which serves as the foundation for further analysis.

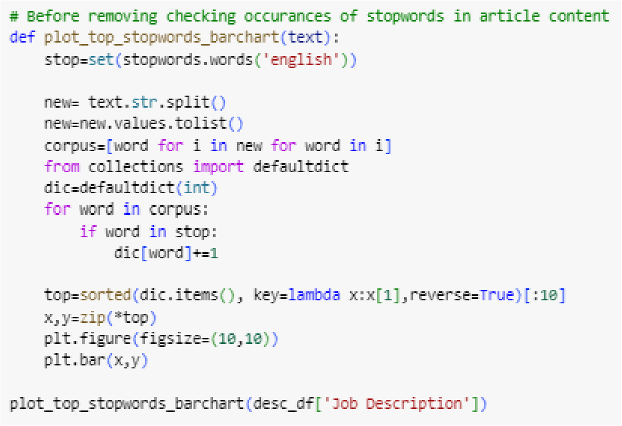


```
#nltk tokenization
desc_df['tokenized_desc'] = desc_df['Job Description'].apply(word_tokenize)
desc_df.sample(10, random_state = 5)
```

Figure 23: Code for tokenization

After using the' sample' function to randomly select ten samples from the dataframe `desc_df`, with a fixed random state of 5. Setting the random_state ensures that the same sample can be generated consistently.

A key component of effective text analysis is the recognition and elimination of stopwords, common but uninformative words. This section describes the integration of a Natural Language Toolkit (nltk) stopwords library to expedite the removal of these common phrases. There is a function called `plot_top_stopwords_barchart(text)`, which creates a bar chart showing the most frequently used stopwords in job descriptions (see figure 24). Examples of stopwords include words like "and," "the," and "a." The text is divided into separate words, and then a new list called "corpus" is generated, which includes all the words from all the job descriptions. To keep track of how many times each stopword appears in the corpus, a defaultdict is utilized. After sorting the dictionary based on the number of occurrences, extract the top 10 most frequent stopwords and create a bar chart using Matplotlib to display the most common stopwords. The x-axis represents the stopwords, while the y-axis represents their frequency.



```python
# Before removing checking occurances of stopwords in article content
def plot_top_stopwords_barchart(text):
    stop=set(stopwords.words('english'))

    new= text.str.split()
    new=new.values.tolist()
    corpus=[word for i in new for word in i]
    from collections import defaultdict
    dic=defaultdict(int)
    for word in corpus:
        if word in stop:
            dic[word]+=1

    top=sorted(dic.items(), key=lambda x:x[1],reverse=True)[:10]
    x,y=zip(*top)
    plt.figure(figsize=(10,10))
    plt.bar(x,y)

plot_top_stopwords_barchart(desc_df['Job Description'])
```

Figure 24: Analysis of Stopword Occurrence

Figure 25 displays the bar chart output generated by the plot_top_stopwords_barchart function. The tool visually represents the top 10 most used stopwords found in job descriptions. The bars that are highest in the graph indicate the words that are most frequently used, such as "and," "to," and "the.".

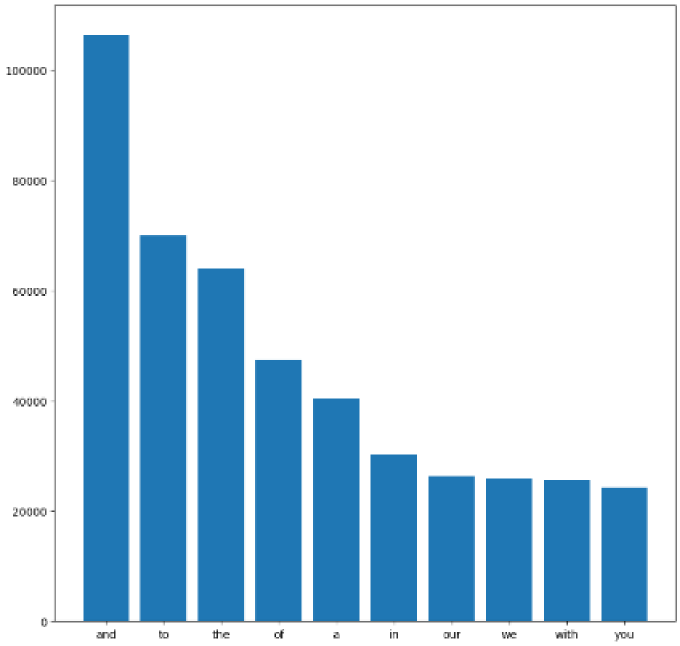
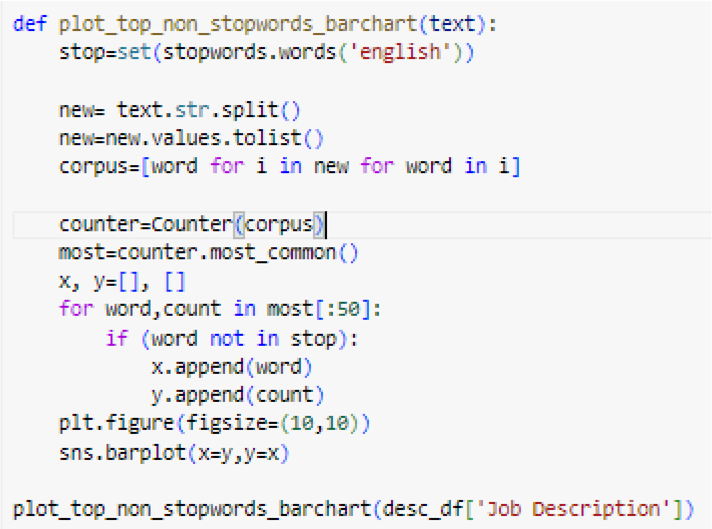


Figure 25: Count of stopwords

The graph plays a crucial role in text analysis as it helps us grasp the distribution of frequently used words in the text data. This understanding is valuable when making decisions about additional preprocessing tasks, such as removing stopwords, to prioritize more significant words. These analyses are commonly employed in data science to preprocess text data for more advanced NLP tasks. Some examples include sentiment analysis, thematic analysis, and enhancing the accuracy of machine learning models by removing irrelevant information from the data.

4.3.3 Visualization of Frequent Non-Stopwords in Job Description

The code below aims to plot the most commonly occurring words in a corpus of text data, which, in this case, are job descriptions (See Figure 26). The code focuses on excluding stopwords from the analysis. This program uses Python's Seaborn library to create visualizations, and it also uses the Counter class from the collections module to count the occurrences of words. In this process, the elimination of common English stopwords so that the concentrate on words that have a higher chance of conveying specific meanings within the job descriptions.



```
def plot_top_non_stopwords_barchart(text):
    stop=set(stopwords.words('english'))

    new= text.str.split()
    new=new.values.tolist()
    corpus=[word for i in new for word in i]

    counter=Counter(corpus)
    most=counter.most_common()
    x, y=[], []
    for word,count in most[:50]:
        if (word not in stop):
            x.append(word)
            y.append(count)
    plt.figure(figsize=(10,10))
    sns.barplot(x=y,y=x)

plot_top_non_stopwords_barchart(desc_df['Job Description'])
```

Figure 26: Code for visualization of frequent non-Stopwords in job description

The horizontal bar chart shows the most frequently used non-stop words. The chart displays horizontal bars, each representing a distinct non-stop word. Each bar's length corresponds to the word's frequency in the corpus. The words are arranged in order, starting with the most commonly used non-stopword at the top and continuing with less frequently used words. The job descriptions analysed show that terms like "team," "business," and "financial" are frequently used. This indicates that these terms are commonly found in job descriptions (refer figure 27). Visualizations like this are critical in the field of data science. They help us gain valuable insights from text data, uncover patterns and themes, and identify keywords that could be crucial for further analysis. These visualizations are particularly useful for tasks like identifying trends or creating features for machine learning models.

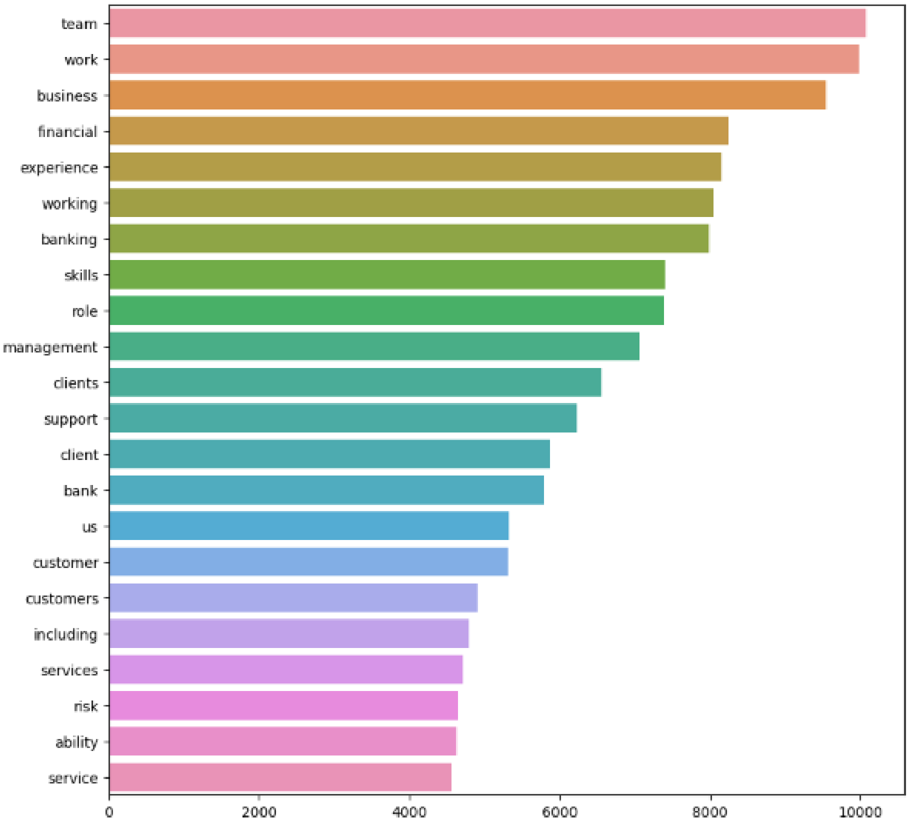
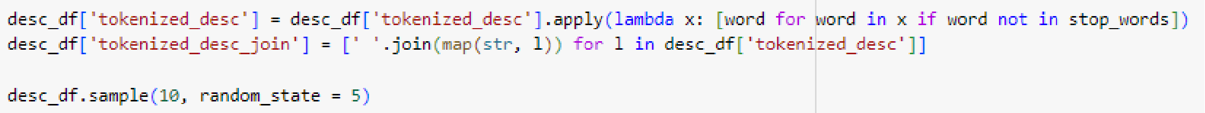


Figure 27: Top Count of Non – stopwords Frequency

4.3.4 Tokenization and Stopword Removal in Job Descriptions

The following code (see figure 28) snippet is a crucial component of a text preprocessing pipeline used in this study, particularly for tasks related to NLP. The first step in processing the DataFrame is to tokenize the job descriptions. This involves breaking down the text into individual words or tokens. After that, the next step is to remove stopwords, commonly used words that do not contribute much to the text's overall meaning.

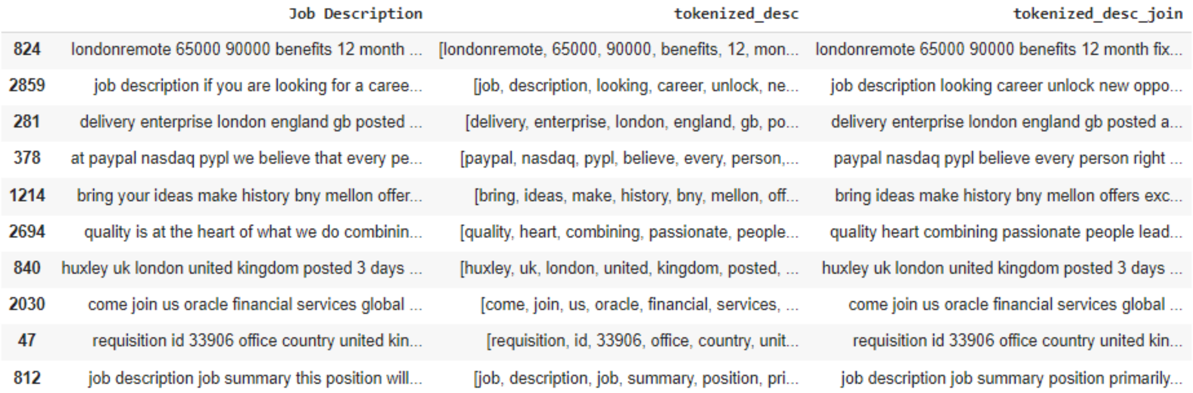


```
desc_df['tokenized_desc'] = desc_df['tokenized_desc'].apply(lambda x: [word for word in x if word not in stop_words])
desc_df['tokenized_desc_join'] = [' '.join(map(str, l)) for l in desc_df['tokenized_desc']]

desc_df.sample(10, random_state = 5)
```

Figure 28: text preprocessing pipeline

These are the important steps of the code: In this code the iteration over the 'tokenized_desc'is performed. This column contains lists of word tokens extracted from the job descriptions. The code utilizes a lambda function to remove words from the list stop_words selectively. As a result, every job description has been transformed into a list of words, excluding any stopwords. In the 'tokenized_desc' column of the 'desc_df' data frame into a single string. The resulting strings are stored in a new column called 'tokenized_desc_join'. Once the stopwords have been eliminated, this line combines the tokens into a single string for each job description. The join() function combines a list of words into a single string, with spaces separating each word.

Using `map(str, l)` ensures that all tokens in the list `l` are converted to strings before joining. This is done to account for the possibility of having non-string tokens in the list. To randomly select ten samples from the data frame desc_df, with a random state of 5, used the code desc_df.sample(10, random_state=5). This line of code utilizes a specific random state for reproducibility to display a random sample of 10 rows from the DataFrame.



	Job Description	tokenized_desc	tokenized_desc_join
824	londonremote 65000 90000 benefits 12 month ...	[londonremote, 65000, 90000, benefits, 12, mon...	londonremote 65000 90000 benefits 12 month fix...
2859	job description if you are looking for a caree...	[job, description, looking, career, unlock, ne...	job description looking career unlock new oppo...
281	delivery enterprise london england gb posted ...	[delivery, enterprise, london, england, gb, po...	delivery enterprise london england gb posted a...
378	at paypal nasdaq pypl we believe that every pe...	[paypal, nasdaq, pypl, believe, every, person,...	paypal nasdaq pypl believe every person right ...
1214	bring your ideas make history bny mellon offer...	[bring, ideas, make, history, bny, mellon, off...	bring ideas make history bny mellon offers exc...
2694	quality is at the heart of what we do combinin...	[quality, heart, combining, passionate, people...	quality heart combining passionate people lead...
840	huxley uk london united kingdom posted 3 days ...	[huxley, uk, london, united, kingdom, posted, ...	huxley uk london united kingdom posted 3 days ...
2030	come join us oracle financial services global ...	[come, join, us, oracle, financial, services, ...	come join us oracle financial services global ...
47	requisition id 33906 office country united kin...	[requisition, id, 33906, office, country, unit...	requisition id 33906 office country united kin...
812	job description job summary this position will...	[job, description, job, summary, position, pri...	job description job summary position primarily...

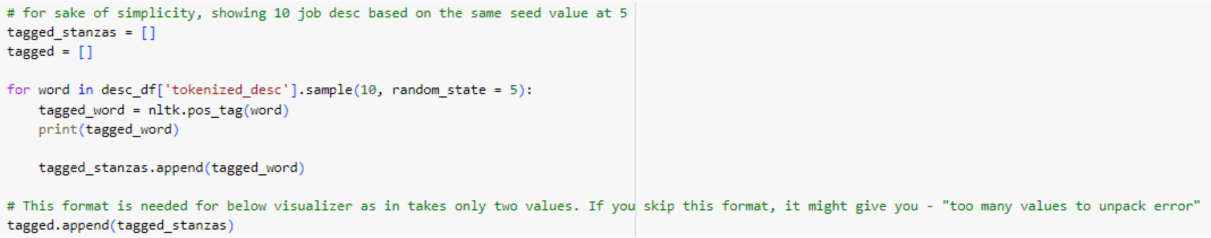
Figure 29: specific random state for reproducibility to display

Figure 29 is displayed as the output after performing these operations. In the dataset, there are three columns. The first column, called 'Job Description', contains the original job descriptions. The second column, ' tokenized_desc', contains lists of words from the job descriptions, excluding common stopwords. The third column is 'tokenized_desc_join' and shows the cleaned job descriptions, where the words are combined back into strings.

Text preprocessing plays a vital role in NLP tasks as it helps eliminate unnecessary noise and allows us to concentrate on the meaningful content of the text. Before proceeding with further analysis, such as topic modelling, sentiment analysis, or inputting the data into machine learning algorithms for predictive tasks, it is crucial to complete this essential step. By cleaning and processing the text, it is significant to enhance the accuracy of the analysis. This is achieved by giving more attention to the words that hold the most tremendous significance.

4.3.5 Part-Of-Speech Tagging for Job Descriptions

Part-of-Speech (POS) tagging, which gives each word a particular grammatical category (such as noun, verb, adjective, etc.), is essential for comprehending the grammatical structure of text. The code in figure 30 tags the parts of speech for individual words in ten randomly selected job descriptions using the Natural Language Toolkit (nltk).



```
# for sake of simplicity, showing 10 job desc based on the same seed value at 5
tagged_stanzas = []
tagged = []

for word in desc_df['tokenized_desc'].sample(10, random_state = 5):
    tagged_word = nltk.pos_tag(word)
    print(tagged_word)

    tagged_stanzas.append(tagged_word)

# This format is needed for below visualizer as in takes only two values. If you skip this format, it might give you - "too many values to unpack error"
tagged.append(tagged_stanzas)
```

Figure 30: specific random state for reproducibility to display

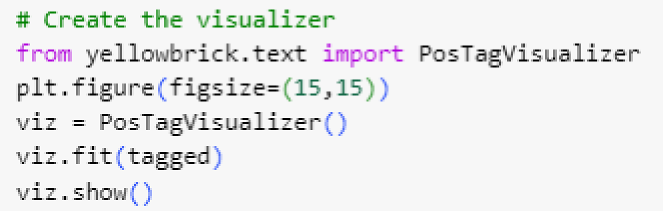
Every word in the sampled tokenized job descriptions is assigned a part of the speech tag using the nltk during the loop.pos_tag(word) function from the NLTK library. The function will give us a list of tuples. Each tuple will contain a word and its corresponding POS tag. The printed output, print(tagged_word), is most likely used for quickly visually inspecting the tagged words. In a typical data science environment, it is common practice to log or write this information to a file for future analysis.

To store tagged words, we append them to the list called "tagged_stanzas". This list aims to gather the POS-tagged words from the sampled job descriptions. This data can be used for further analysis or visualization. The code provided does not contain the actual visualization component. However, it indicates that the prepared data structure is necessary for a future visualizer tool, which will likely display the POS tags in a graphical format. The code comment suggests that the visualizer requires a specific two-level structure. If the structure is not followed correctly, an error will occur.

POS tagging is a versatile tool that has many uses. It can help with tasks like syntax parsing, converting text to speech, retrieving information, and enhancing the performance of machine learning models. By providing these models with more specific information about how words are used in different contexts, POS tagging helps them perform better. Visualizing POS tags can be beneficial for comprehending the linguistic structure of text data. This understanding plays a crucial role in developing advanced NLP systems.

4.3.6 Visualizing Part-of-Speech Tag Frequencies in Text Data

The following code is a component of a data visualization task in the data science process, specifically related to NLP (figure 31). The code utilizes a Part-of-Speech (POS) Tag Visualizer from the Yellowbrick library. Yellowbrick is a collection of visual diagnostic tools specifically created to assist with Machine Learning using Scikit-Learn in Python.



```
# Create the visualizer
from yellowbrick.text import PosTagVisualizer
plt.figure(figsize=(15,15))
viz = PosTagVisualizer()
viz.fit(tagged)
viz.show()
```

Figure 31: Code for visualizing part of speech

Figure 32 is the bar chart showing the frequency of different parts of speech (POS) tags in the analysed text corpus. Each bar in the visualization represents a specific part-of-speech (POS) tag based on the Penn Treebank POS tags. The height of each bar indicates the frequency of that particular tag in the corpus.

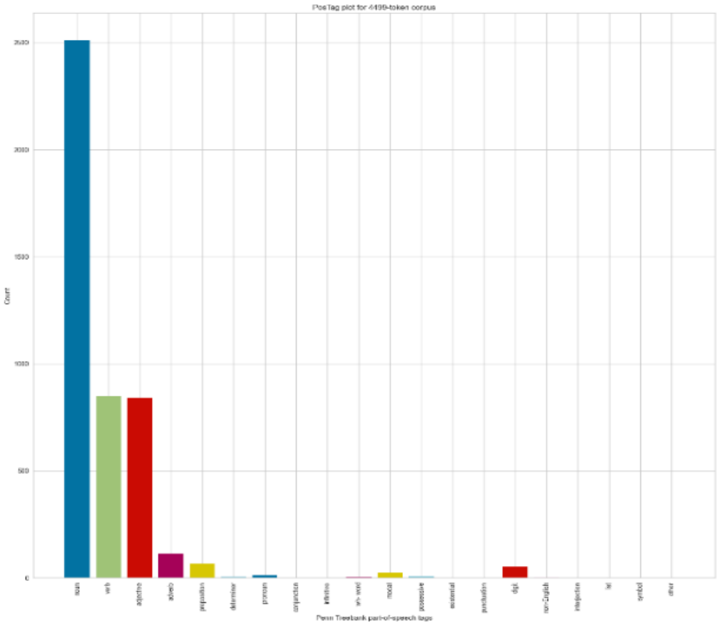


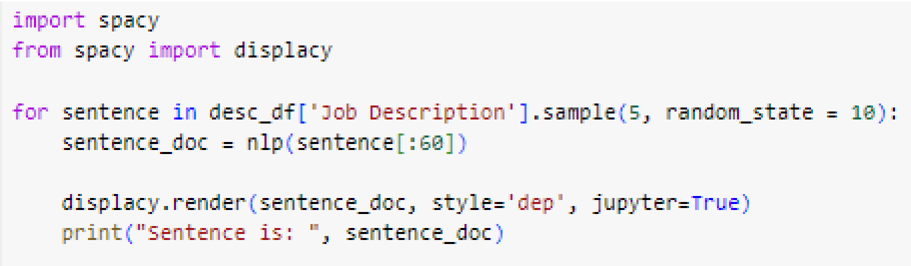
Figure 32: PoS tag visualizer

On the x-axis, we can see the various parts of speech (POS) tags. The tags that are included are nouns (NN), verbs (VB), adjectives (JJ), and adverbs (RB). The count of each POS tag appearing in the corpus is represented by the y-axis. Based on the visualization, the tallest bar represents the most frequent tag, which happens to be nouns (NN). This suggests that nouns are the most used in the job descriptions.

This kind of visualization is beneficial for comprehending the grammatical structure of the text. It can aid in tasks like feature engineering for NLP models, identifying patterns in language usage, and even informing the development of grammar-based information extraction systems.

4.3.7 Visualizing Dependency Parses in NLP

To improve the understanding of language structures, this section examines how dependency parses are visualized in NLP. This technique gives a graphical depiction of syntactic links seen in job descriptions. Figure 33 represents the code that used to generate dependency parse trees for sentences taken from job descriptions.



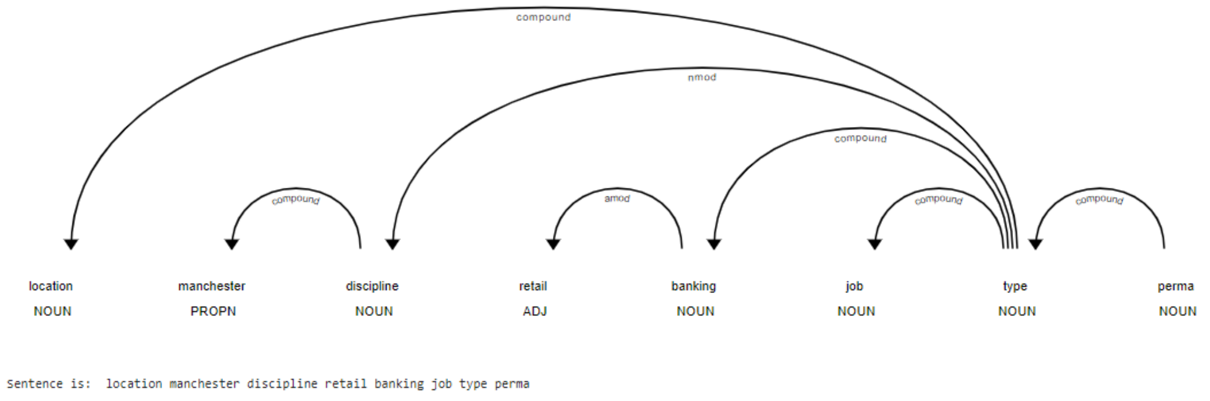
```
import spacy
from spacy import displacy

for sentence in desc_df['Job Description'].sample(5, random_state = 10):
    sentence_doc = nlp(sentence[:60])

    displacy.render(sentence_doc, style='dep', jupyter=True)
    print("Sentence is: ", sentence_doc)
```

Figure 33: Code for visualizing Natural Language Processing (NLP)

To randomly select 5 job descriptions from the 'Job Description' column in the desc_df dataframe, with a random state of 10, one can use the following code: desc_df['Job Description'].sample(5, random_state=10). A random state of 10 is used to select a random sample of 5 sentences from the 'Job Description' column in the desc_df DataFrame to ensure reproducibility.

For simplicity or clarity, the code takes each sampled sentence and truncates it to the first 60 characters. Then, it processes the truncated sentence using a spaCy NLP model, represented by the variable nlp. The model will provide a Doc object that includes the processed sentence.

The function called `displacy.render()` has used to visualize the dependency parse of a sentence. To do this, pass in the `sentence_doc` object and set the ` The sentence_doc is rendered using spaCy's displacy module to display the dependency tree. When using the style='dep' argument, we can visualize a dependency tree. Additionally, setting jupyter=True will display the visualization directly in a Jupyter notebook.



```
Sentence is:  location manchester discipline retail banking job type perma
```

Figure 34: Sentence formation visualizer

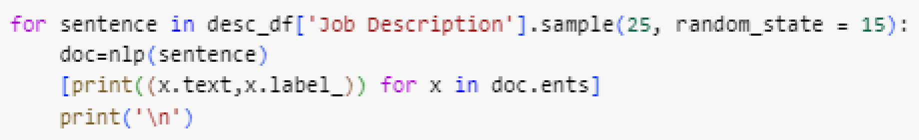
In the graph (see figure 34), every word in the sentence is represented as a node. Each word is marked with its corresponding part-of-speech (POS) tag, such as NOUN, PROPN (proper noun), ADJ (adjective). The arrows are used to connect words and indicate their grammatical relationships. The direction of the arrow points from the main word to the dependent word. The arrows' labels show the dependency type, like nmod (nominal modifier), compound, amod (adjectival modifier).

Visualizations play a crucial role in NLP as they help us grasp the connections between words in a sentence. They are beneficial for tasks like information extraction, where understanding the relationships between entities is vital. Dependency trees help comprehend the structure of sentences and can enhance the effectiveness of different NLP applications. These applications include parsing questions for chatbots, extracting information for recommendation systems, and analysing sentiment.

4.4 Named Entity Recognition Output from Job Descriptions

Named Entity Recognition (NER) is a widely used task in NLP. It aims to identify and categorize named entities mentioned in text into specific categories. These categories can include the names of individuals, organizations, locations, expressions of time, quantities, monetary values, percentages, and more.

The code used to apply Named Entity Recognition (NER) is given in figure 35.



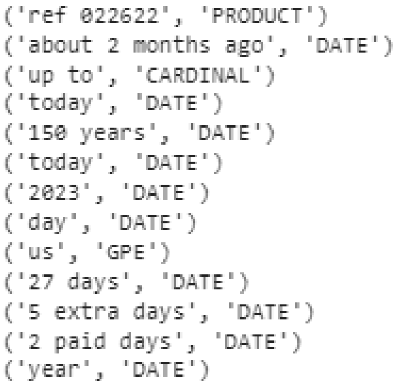
```
for sentence in desc_df['Job Description'].sample(25, random_state = 15):
    doc=nlp(sentence)
    [print((x.text,x.label_)) for x in doc.ents]
    print('\n')
```

Figure 35: Code for named entity recognition

Sentence Sampling is a technique used in NLP to select a subset of sentences from a larger text corpus. The code selects 25 sentences from the 'Job Description' column in the desc_df DataFrame. It uses a fixed random state of 15 to ensure reproducibility.

NLP models, most likely spaCy language models, processes each sentence using the command nlp(sentence). This command can identify named entities in the given text.

The recognized entities are stored in the `doc.ents` attribute of the resulting `doc` object. Every entity, denoted as x, consists of a text component (x.text) and a label (x.label_) representing the entity type. Figure 36 represents the part of results that have been printed:



```
('ref 022622', 'PRODUCT')
('about 2 months ago', 'DATE')
('up to', 'CARDINAL')
('today', 'DATE')
('150 years', 'DATE')
('today', 'DATE')
('2023', 'DATE')
('day', 'DATE')
('us', 'GPE')
('27 days', 'DATE')
('5 extra days', 'DATE')
('2 paid days', 'DATE')
('year', 'DATE')
```

Figure 36: Text and label of every entity in the sentences

After executing the code, it will display the text and label of every entity in the sentences. The result is a collection of tuples, where each tuple includes the entity and its associated label.

The NER process is crucial for tasks like information extraction. It helps use entities as features for machine learning models and automatically categorizes critical information to enrich datasets. For example, when analysing job descriptions, NER can extract and analyse various details, such as the types of positions, locations, or qualifications mentioned. This helps gain a more organized and quantitative understanding of the job market.

4.5 N-gram Frequency Analysis to Reveal Linguistic Patterns

In this part, N-gram frequency analysis has conducted to delve into the complex linguistic subtleties of job descriptions. A trigram refers to a group of three words that appear consecutively. The tri-grams that were discovered from the job descriptions, which included terms like "success equal opportunity" and "sexual orientation gender," provided insight into common language themes found in the dataset. This research plays a critical role in explaining the unique terminology used in the banking industry, highlighting important concepts like diversity, equal opportunity, and compliance with relevant legislation. It is important to recognize the frequency of these elements as well as their contextual importance within the larger framework of job descriptions. Examining the consequences allows to get important insights regarding language patterns in the industry, which will helps to understand the sector's beliefs and goals on a more accurate level.

Figure 37 is the bar chart that shows the top 20 trigrams in job descriptions.

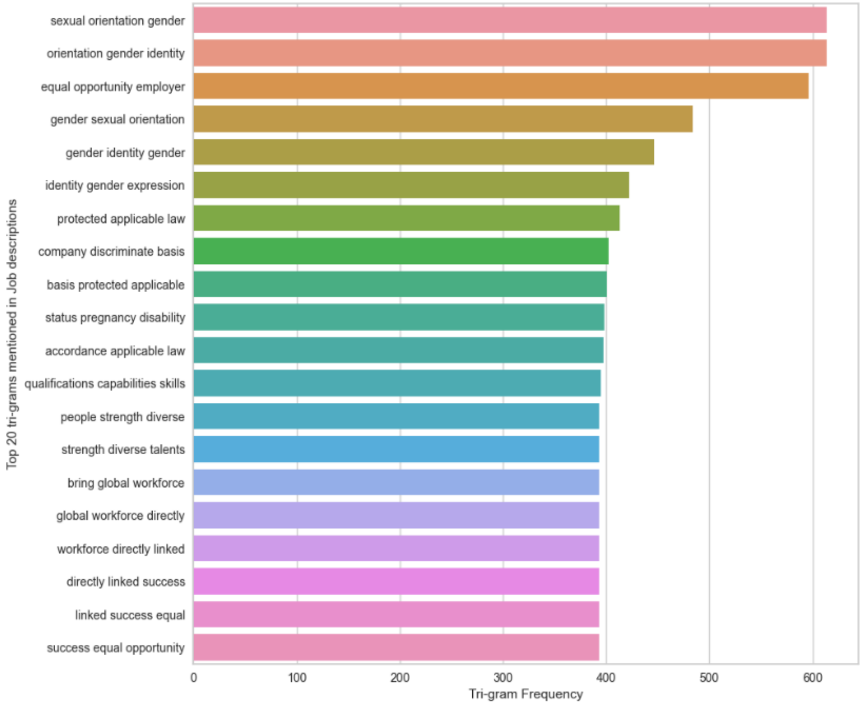
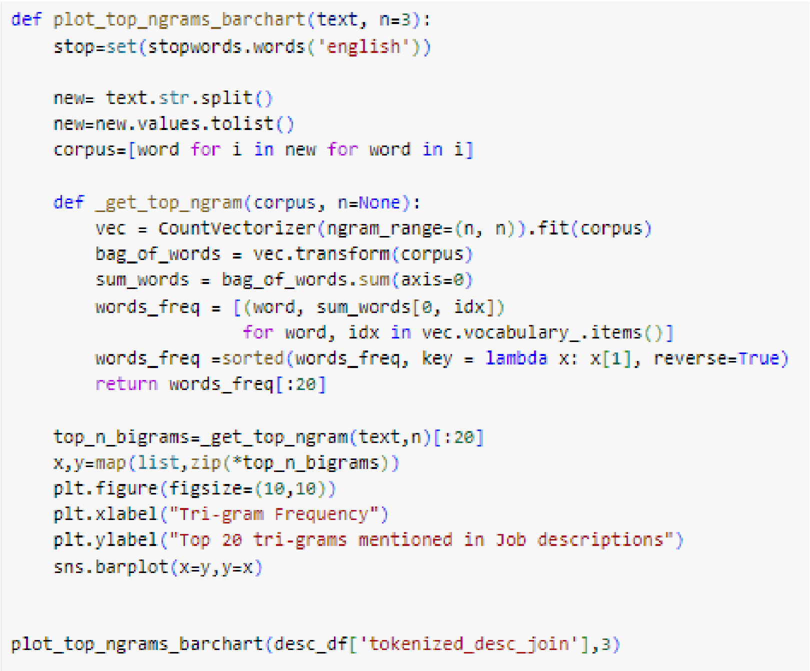


Figure 37: Most frequent tri-grams

To extract n-grams from the corpus, _get_top_ngram function has used. This function utilizes the CountVectorizer class from sci-kit-learn, which converts a collection of text documents into a matrix of token counts. Figure 38 shows the code set to extract trigrams.



```
def plot_top_ngrams_barchart(text, n=3):
    stop=set(stopwords.words('english'))

    new= text.str.split()
    new=new.values.tolist()
    corpus=[word for i in new for word in i]

    def _get_top_ngram(corpus, n=None):
        vec = CountVectorizer(ngram_range=(n, n)).fit(corpus)
        bag_of_words = vec.transform(corpus)
        sum_words = bag_of_words.sum(axis=0)
        words_freq = [(word, sum_words[0, idx])
                      for word, idx in vec.vocabulary_.items()]
        words_freq =sorted(words_freq, key = lambda x: x[1], reverse=True)
        return words_freq[:20]

    top_n_bigrams=_get_top_ngram(text,n)[:20]
    x,y=map(list,zip(*top_n_bigrams))
    plt.figure(figsize=(10,10))
    plt.xlabel("Tri-gram Frequency")
    plt.ylabel("Top 20 tri-grams mentioned in Job descriptions")
    sns.barplot(x=y,y=x)


plot_top_ngrams_barchart(desc_df['tokenized_desc_join'],3)
```

Figure 38: Code for extract n-grams

After fitting the CountVectorizer to the corpus, it calculates the frequency of each trigram. To achieve this, the frequency of each trigram across all the documents have added together.

To begin, the frequencies of the trigrams are sorted in descending order. Afterwards, the top 20 trigrams are chosen for visualization.

The `sns.barplot` function from the Seaborn library has used to create a bar chart. Further trigram frequencies on the x-axis and the trigram texts on the y-axis has defined. To enhance the chart's readability, resized the dimension to 10x10 inches. Additionally, it has ensured that the axes are properly labelled to represent the content displayed accurately.

Trigram analysis plays a crucial role in exploratory data analysis in NLP. It helps uncover language patterns in job descriptions, offering insights into organizations' cultural and social values, including their dedication to diversity and inclusivity. These insights can shape HR strategies, marketing approaches, and overall business practices.

4.6 Enhancing Job Title Analysis through Skill Annotation and Word2Vec Modelling

Being able to effectively analyse job descriptions is incredibly valuable in today's ever-changing job market. To take a more advanced approach, we can extract valuable insights from job descriptions by using techniques such as breaking down the text into smaller parts, identifying specific skills mentioned, and applying Word2Vec modelling.

To start off, this approach involves simplifying job descriptions by getting rid of common words and focusing on specific terms that truly define the role. Having a skill dictionary is important because it allows us to extract specific skills from the general text. This annotation helps connect the dots between a job description and the specific skills needed for the job. It gives a more organized way of looking at information that is not structured.

Using Word2Vec modelling takes this analysis to another level by grasping the context in which words are used. This tool helps to find similar skills and enhances the skill sets for each job role by comparing the meaning of different terms.

By normalizing job titles and use clustering techniques it can group similar roles together. This helps us identify patterns and trends in the skills needed for these roles. Clustering also helps to visualize the data, like using heatmaps to show how different skills are distributed across job roles.

The process of annotation is important in analysing job titles, since it helps to convert text into a structured format that can be easily analysed using numbers and statistics. It helps break down job descriptions into specific skills, giving a clear understanding of what employers are looking for in potential candidates.

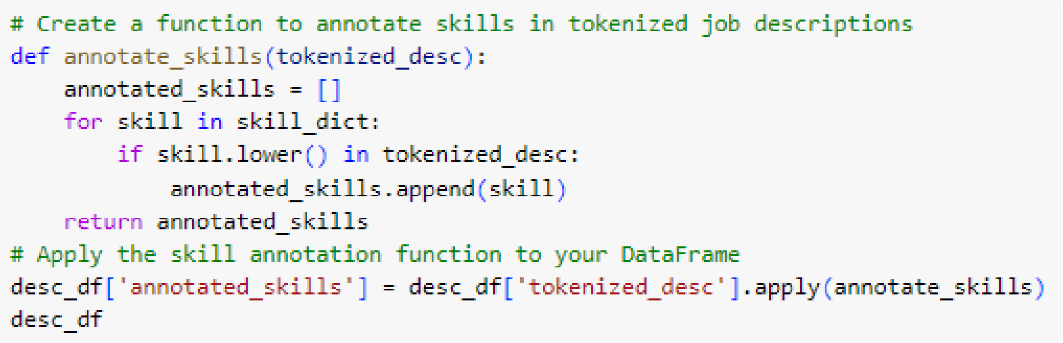
This process is important for different people involved in the job market. Employers can gain valuable insights into the skills that are currently in high demand. This allows them to customize their recruitment strategies accordingly. Job seekers can figure out the skills they need to work on to be competitive. Career counsellors and educational institutions should make sure that their curriculums are in line with what the job market needs. This way, students will have the right skills for the real world.

Data science proves to be a useful instrument in the field of job title analysis, greatly advancing our understanding of the dynamic labour market. By utilizing methods like skill annotation and modelling, data science not only offers up-to-date information but also gives various stakeholders the knowledge they need to make wise judgments in a job market that is always changing.

4.7 Skill Extraction from Job Descriptions

When it comes to skill extraction, the first step is to get job descriptions ready for in-depth examination. The first phase in the procedure is to expand the contractions in the job descriptions, which is an important way to get the language more uniform. After contraction expansion, the text is put back together into whole sentences and changed to lowercase to ensure consistency in case and remove any biases caused by different capitalization. The job descriptions are then tokenized, which is an essential step in many NLP jobs.

The code represented in figure 39, is about annotation. It's designed to find and label certain skills in job descriptions that have been broken down into individual words.



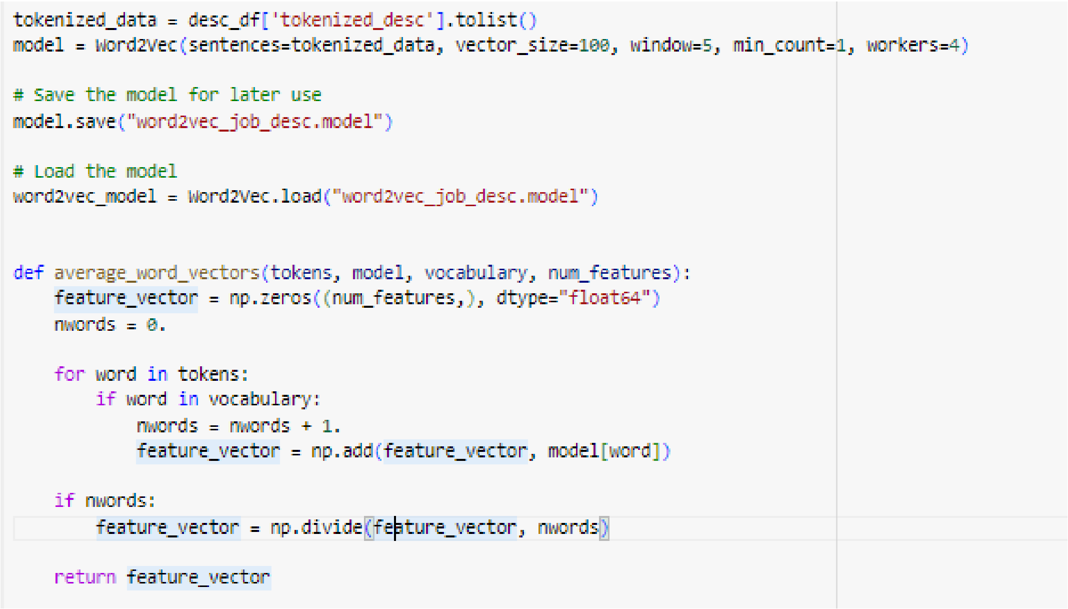
```
# Create a function to annotate skills in tokenized job descriptions
def annotate_skills(tokenized_desc):
    annotated_skills = []
    for skill in skill_dict:
        if skill.lower() in tokenized_desc:
            annotated_skills.append(skill)
    return annotated_skills
# Apply the skill annotation function to your DataFrame
desc_df['annotated_skills'] = desc_df['tokenized_desc'].apply(annotate_skills)
desc_df
```

Figure 39: Code for skill extraction from job description

The first step is to use a function called "annotate_skills" that scans through each tokenized job description. It checks each word or token in the description against a list of skills that are already defined and stored in skill dictionary. When it finds a match, it means that a specific skill from the skill dictionary is mentioned in the job description. In that case, we add this skill to a list of annotated skills for that description. In this function it goes through all the words in each description and picks out the ones that are considered important skills. Based on the definition of this function, it applies to the entire DataFrame of job descriptions (desc_df) and creates a new column called annotated_skills, which contains the list of skills identified for each job. This approach takes the messy text data from job descriptions and turns it into a more organized format. The focus is on the skills extraction, which makes it easier to analyse and understand. Having such a method is important in data-driven fields because it helps us understand the demand for skills, guides in developing the workforce, and provides valuable information for educational and training programs.

4.7.1 Enhancing Job Description Analysis with Word2Vec and Cosine Similarity

This section explores a novel method for enhancing the analysis of job descriptions by combining Word2Vec and Cosine Similarity. The fundamental component of this methodology is the use of the Word2Vec model, a widely used technique that converts words into numerical vectors that represent their context.



```
tokenized_data = desc_df['tokenized_desc'].tolist()
model = Word2Vec(sentences=tokenized_data, vector_size=100, window=5, min_count=1, workers=4)

# Save the model for later use
model.save("word2vec_job_desc.model")

# Load the model
word2vec_model = Word2Vec.load("word2vec_job_desc.model")


def average_word_vectors(tokens, model, vocabulary, num_features):
    feature_vector = np.zeros((num_features,), dtype="float64")
    nwords = 0.

    for word in tokens:
        if word in vocabulary:
            nwords = nwords + 1.
            feature_vector = np.add(feature_vector, model[word])

    if nwords:
        feature_vector = np.divide(feature_vector, nwords)

    return feature_vector
```

Figure 40:Code for enhancing job description analysis

The first step is to train a Word2Vec model using tokenized job descriptions (see figure 40). Tokenization, in this context, refers to the process of breaking down every job description into separate words or tokens. After that, the Word2Vec model learns how words are represented in a space with many dimensions. In this space, each word is represented by a vector. These vectors capture the semantic relationships between words by considering their context within the job descriptions. For example, words that show up in similar situations will have vectors that are close to each other in this space.

After the model has been trained and saved, we can use it to calculate the average word vectors for groups of tokens. This can be really helpful to represent longer phrases or even entire job descriptions as single vectors. These vectors can then be used for different comparison tasks. The average_word_vectors function creates these representative vectors by taking the average of the vectors of all the words in a token set (See Appendix G). This is a common technique used to combine information from multiple words into a single vector.

The true strength of this approach lies in using cosine similarity, which is a measure that tells how similar two vectors are. The system can figure out which skills are most relevant to a job description by comparing the vector representation of the job description to the vectors of different skills, which are also derived from the Word2Vec model.

To wrap up this procedure, the function is applied consistently to every job description in the DataFrame. Existing annotated abilities or equivalent skills produced from Word2Vec are included in each description. As a result, this produces a very valuable dataset in which each job description is linked to a wide range of skills. The extraction of these skills is made easier by the combination of machine learning and NLP techniques. Such enhanced data is extremely valuable for a variety of applications, from in-depth study of the labour market to the creation of customized job suggestion systems.

4.8 Interpreting Job Market Trends through Word Cloud Visualization

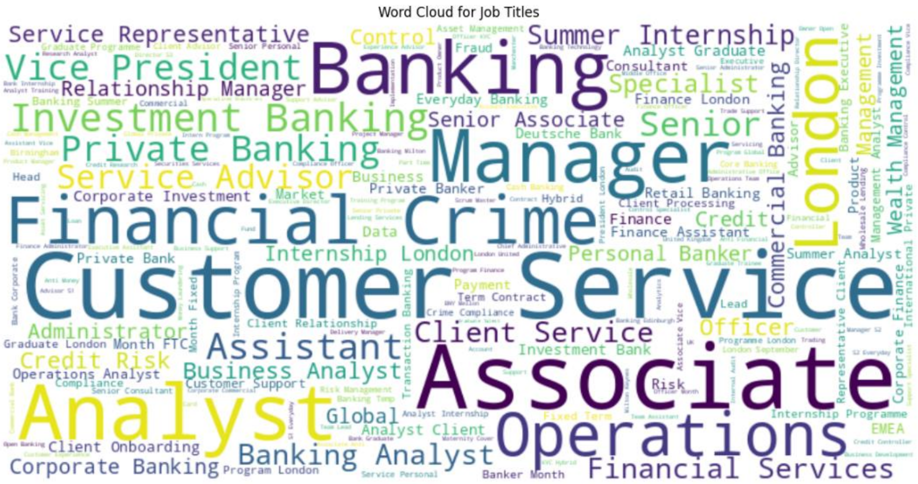


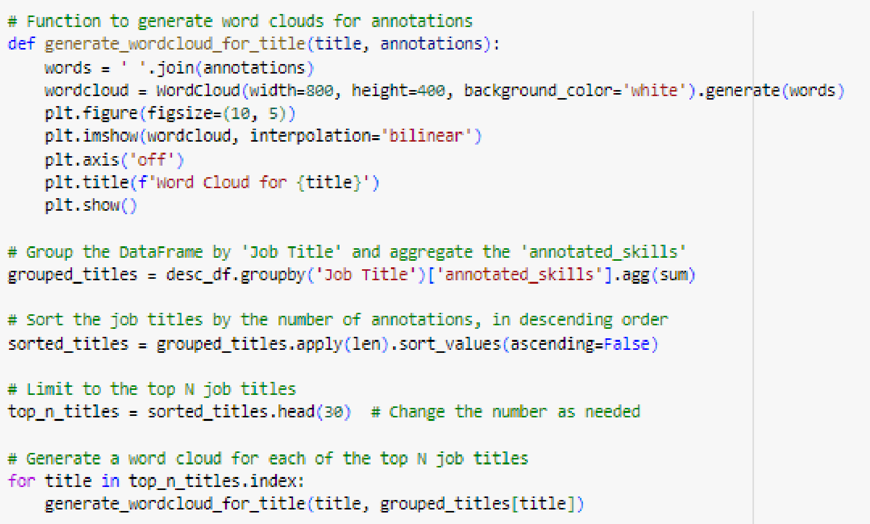
Figure 41: Word cloud for Job market

The word cloud displayed above (see figure 41) shows the terms that are often found from the list of job titles. The size of each word is represented visually in these images, with larger dimensions indicating higher occurrence rates. This graphics gives a brief yet informative summary by emphasizing the most important characteristics and trends found in the job names. Words like "customer," "service," "associate," "manager," "banking," and "financial" are often seen in the word cloud, indicating a significant frequency of phrases related to customer service and financial management jobs. This finding highlights a significant need in the examined dataset for people with experience in financial services, customer relations, and management duties. The graphic provides users with a useful tool by clearly illustrating the most important areas of attention in the labour market. Its insights, which provide a view into industry trends and the particular qualities desired by employers, can be helpful to both job searchers and companies.

4.9 Analysis of Skills Obtained from Job Descriptions

4.9.1 Visualizing Key Skills in Job Titles

This section will explore the visualization and analysis of critical competencies in job titles. A Word clouds will be utilized to present a thorough picture of the frequency and significance of annotations linked to different positions. The code in figure 42 represents the function that used to generate word cloud. It takes two things as input: a job title and a list of annotations, which can be skills or keywords related to that job. After that, it creates a word cloud using these annotations, showing them in varying sizes depending on their frequency.



```
# Function to generate word clouds for annotations
def generate_wordcloud_for_title(title, annotations):
    words = ' '.join(annotations)
    wordcloud = WordCloud(width=800, height=400, background_color='white').generate(words)
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(f'Word Cloud for {title}')
    plt.show()

# Group the DataFrame by 'Job Title' and aggregate the 'annotated_skills'
grouped_titles = desc_df.groupby('Job Title')['annotated_skills'].agg(sum)

# Sort the job titles by the number of annotations, in descending order
sorted_titles = grouped_titles.apply(len).sort_values(ascending=False)

# Limit to the top N job titles
top_n_titles = sorted_titles.head(30)  # Change the number as needed

# Generate a word cloud for each of the top N job titles
for title in top_n_titles.index:
    generate_wordcloud_for_title(title, grouped_titles[title])
```

Figure 42: Code for generating word clouds for annotations

The output generated word clouds for various job titles like Customer Service Advisor, Finance Assistant, and Customer Service Representative (See figure 43). When analysing these word clouds, it was noted that words like "Empathy," "Dedication," and "Telephone" are more prominent, which means they are frequently mentioned when talking about these roles. This technique for visualization helps us understand easily which skills employers prioritize for each position. It gives us insights into the job market's demand for specific competencies. For example, when words like 'empathy' and 'dedication' are displayed in big fonts, it indicates that these qualities are considered extremely important in customer service positions.

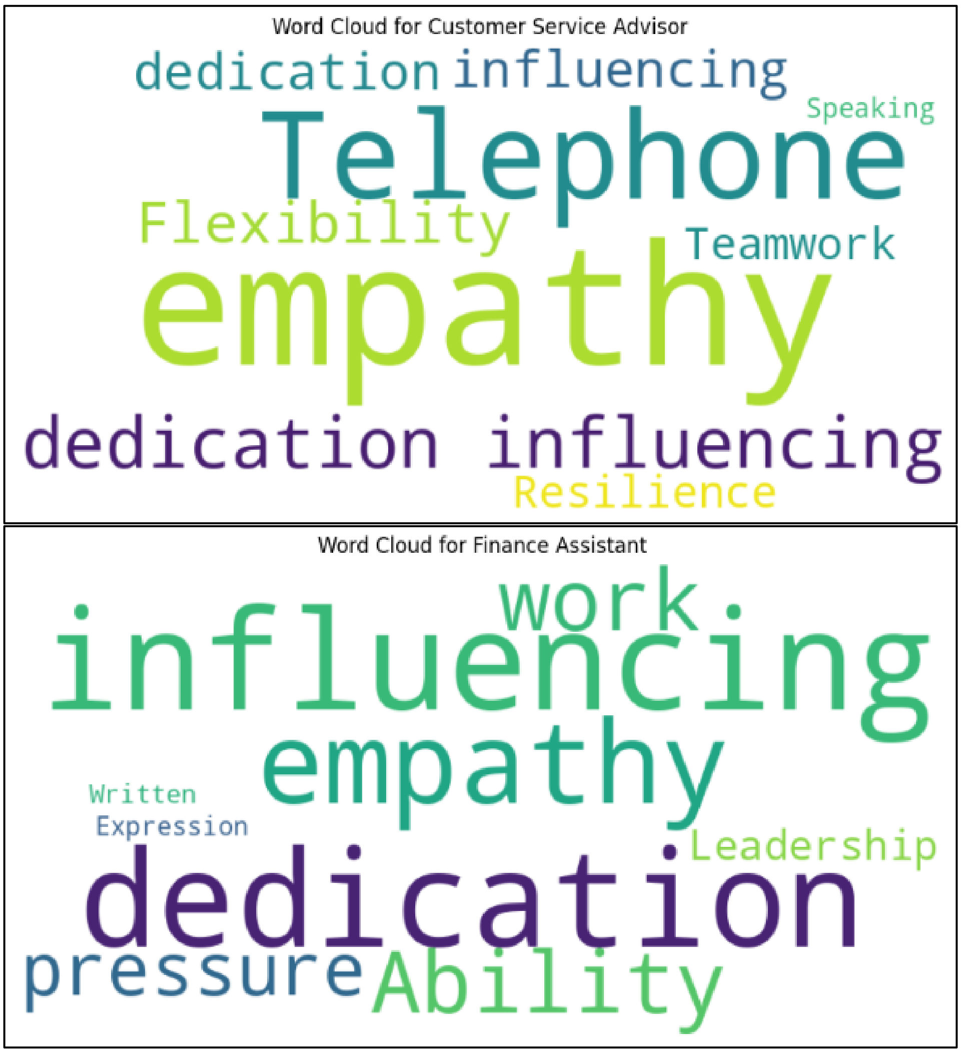


Figure 43: WordClouds represent skills for individual job titles

4.9.2 Analysing Skill Sets in Job Titles Using Jaccard-Indexed Word Clouds

This section systematically examine skill sets in job names using Jaccard-indexed word clouds. Lemmatization is used in the process to standardize job names, and related abilities are transformed into sets. We classify job titles with closely comparable skills using the Jaccard index, a statistical measure of set similarity, taking a 90% similarity threshold into consideration. The result is several groupings, each made up of job titles with related skills, offering a more accurate view of skill patterns in different types of job titles.

To visualize the data, the code generates word clouds. These word clouds show the importance of each skill based on how frequently they appear within the group. This generates word clouds for the top 50 groups that have the fewest number of titles (see figure 44). This helps to visualize and potentially identify position or specialized roles in the job market.

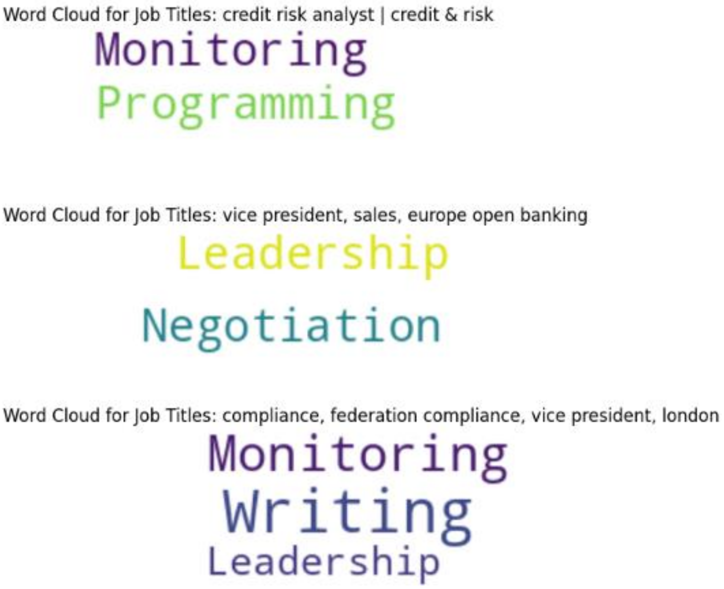
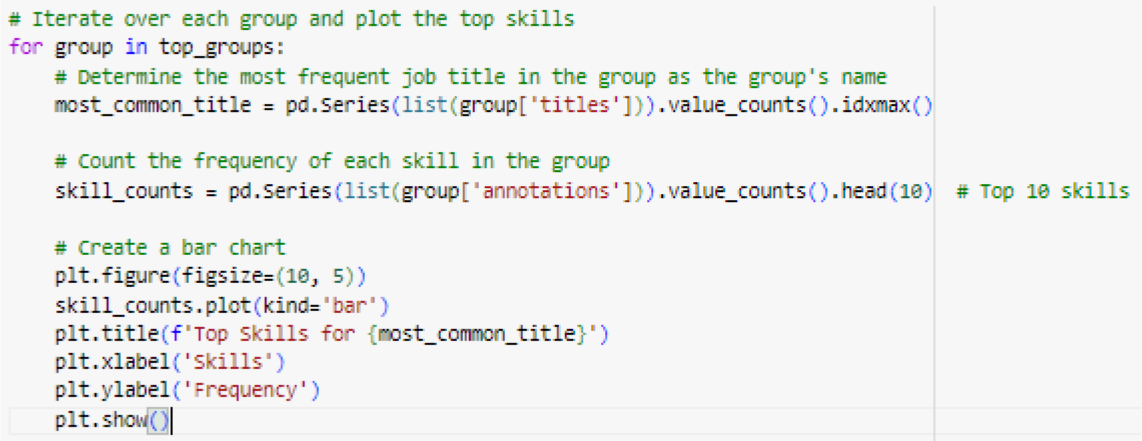


Figure 44: WordCloud for same skills required for different jobs based on jaccard indexing

Upon examining the sample output images, distinct word clouds showing various job titles, such as those associated with "commercial banking" or "vice president" roles, are observable. Every word cloud is designed to highlight the most important competencies associated with that particular job category. Skills like "negotiation," "monitoring," and "leadership" are highlighted in bigger letters, indicating their importance in the related work roles. These graphics make it easier to identify the most valued skills in a collection of job categories, which improves our understanding of the skills that are highly valued in diverse markets.

4.9.3 Visual Analysis of Top Skills for Specific Job Titles

In this section, a dedicated code has been developed to examine the most common skills linked to certain job titles or clusters of related job titles (see figure 45). The procedure is iterating over each set of job titles, finding the title that appears the most frequently in that group, and using that title as a representative label. The algorithm then counts how often each skill that is included in the job descriptions for that specific group is used. The results are shown as a bar chart that highlights the top ten most often used skills.



```
# Iterate over each group and plot the top skills
for group in top_groups:
    # Determine the most frequent job title in the group as the group's name
    most_common_title = pd.Series(list(group['titles'])).value_counts().idxmax()

    # Count the frequency of each skill in the group
    skill_counts = pd.Series(list(group['annotations'])).value_counts().head(10)  # Top 10 skills

    # Create a bar chart
    plt.figure(figsize=(10, 5))
    skill_counts.plot(kind='bar')
    plt.title(f'Top Skills for {most_common_title}')
    plt.xlabel('Skills')
    plt.ylabel('Frequency')
    plt.show()
```

Figure 45: Code for plot the top skills

The sample visualizations give a way to compare essential skills in different professional fields. In these bar graphs, each group of job titles represents a particular industry segment. For example, the graph that's specifically designed for 'commercial banking' roles (see figure 47), highlight important skills like 'Negotiation,' 'Influencing,' and 'Networking.' There is another graph that focuses on executive-level positions, like a 'vice president - model review coordinator' and it shows that the main skills needed for this role are 'Coordination' and 'Mathematics.'

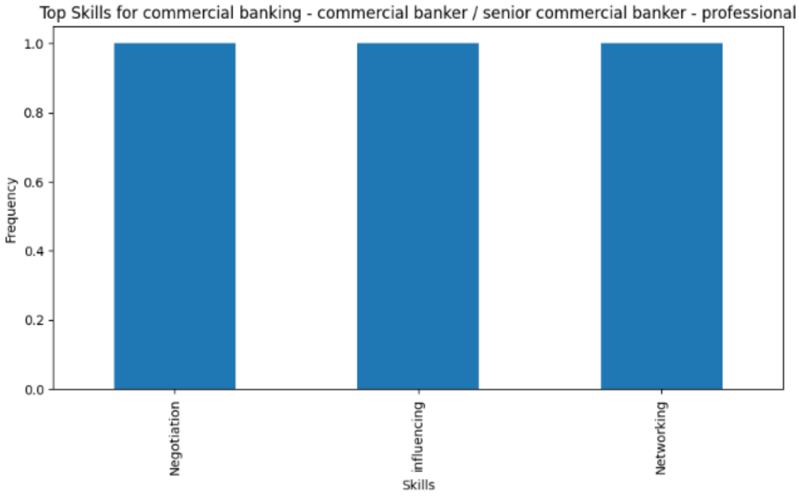


Figure 46: Graph of skills for commercial banking

These visual representations give a quick and informative overview of the important skills that are desired in various professional fields (see figure 47). They show that there is a need for interpersonal skills in banking, while strategic and analytical abilities are highly valued in senior management positions. This distilled information is important for helping job seekers with their career development and helping employers improve their recruitment criteria.

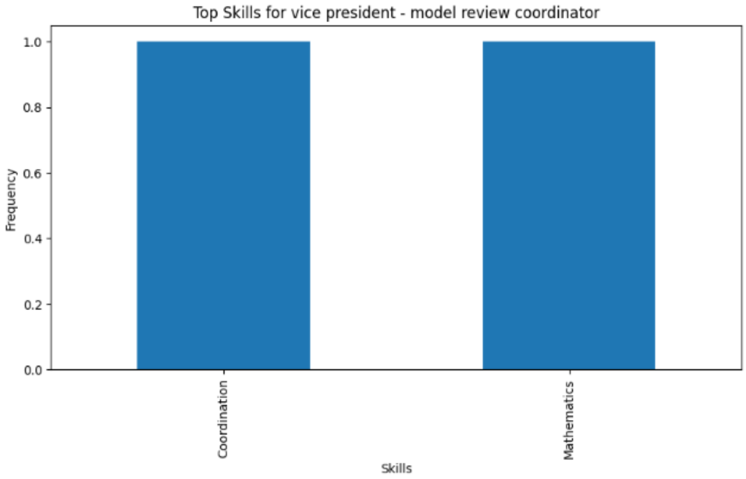
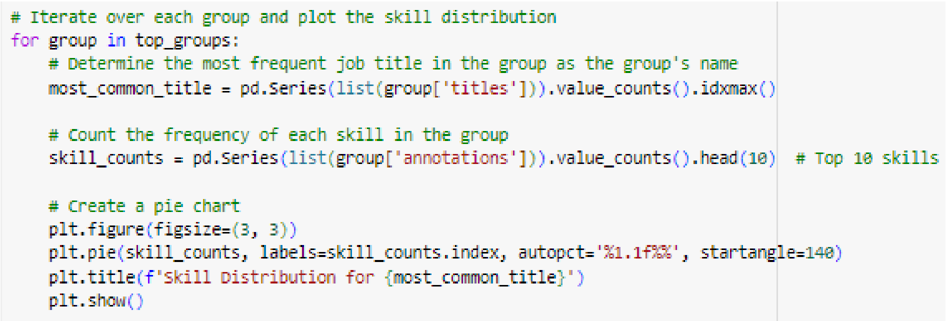


Figure 47: Top common skills for different jobs

4.10 Skill Distribution Insights across Professional Roles

This section provides a visual summary of the distribution of critical skills linked to certain job titles in the dataset. This is accomplished by creating pie charts, which clearly illustrate the importance of specific talents for various job titles.

The phases in the process include going through several categories of job titles that have already been arranged according to how comparable their necessary skills are. The code finds the job title that appears the most frequently in each group and uses that as the group name (see figure 48). It then calculates how often each skill occurs in the group and chooses the most common skill for visualization.

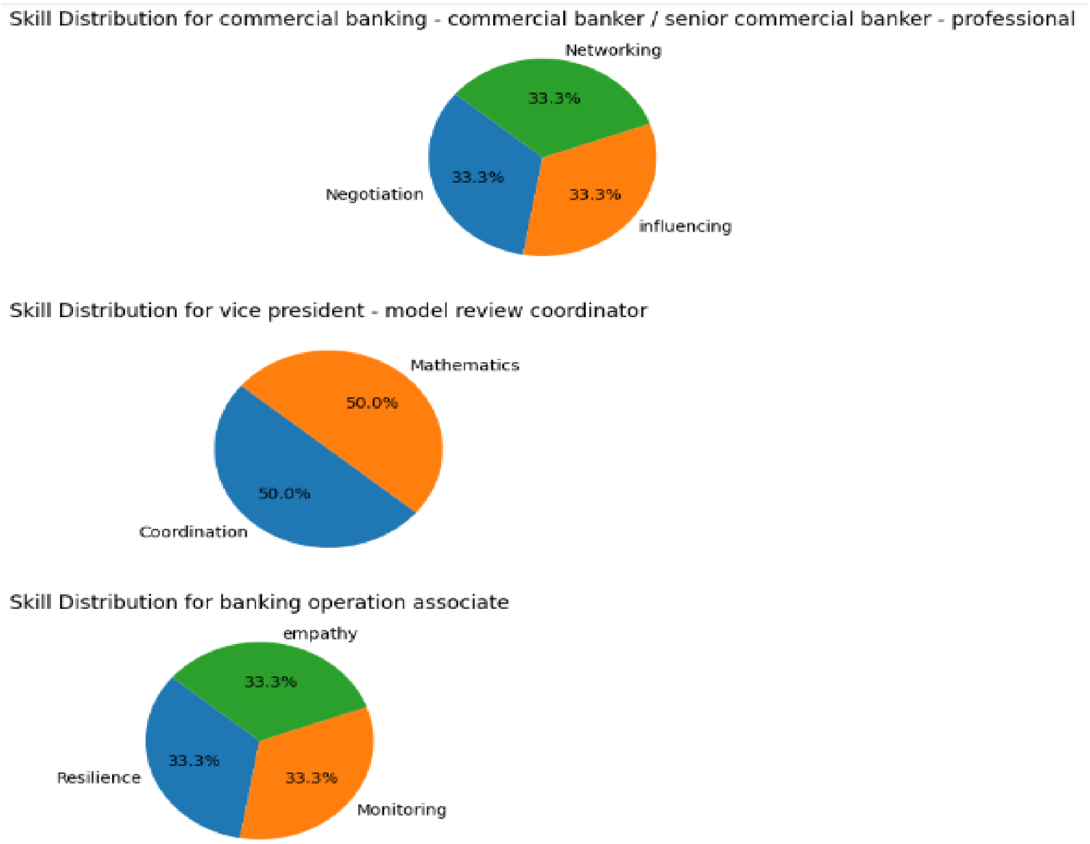


```
# Iterate over each group and plot the skill distribution
for group in top_groups:
    # Determine the most frequent job title in the group as the group's name
    most_common_title = pd.Series(list(group['titles'])).value_counts().idxmax()

    # Count the frequency of each skill in the group
    skill_counts = pd.Series(list(group['annotations'])).value_counts().head(10)  # Top 10 skills

    # Create a pie chart
    plt.figure(figsize=(3, 3))
    plt.pie(skill_counts, labels=skill_counts.index, autopct='%1.1f%%', startangle=140)
    plt.title(f'Skill Distribution for {most_common_title}')
    plt.show()
```

Figure 48: Code for iterating over each group for skill distribution

For each identified job title group, the code generates a pie chart that visually breaks down the percentage share of each top skill, giving viewers an at-a-glance understanding of which skills are most associated with that job title. The labels on the slices of the pie chart show the specific skills, and the size of each slice represents how often that skill appears compared to the others in the group.

The visualizations feature of pie charts illustrate the advanced distribution of job titles throughout various sectors, with a particular emphasis on commercial banking, model review coordination, and banking operations (refer figure 49). The graphical representations show that the distribution of skills necessary for positions in commercial banking and banking operations is balanced. Notably, in commercial banking, abilities like "Negotiation," "Influencing," and "Networking" highlights as critical, whereas banking operations place a premium on abilities like "Empathy," "Resilience," and "Monitoring”.



Skill Distribution for commercial banking - commercial banker / senior commercial banker - professional

Skill Distribution for vice president - model review coordinator

Skill Distribution for banking operation associate

Figure 49: pie charts showing skill distribution for different banking positions

Chapter 5: Implementation

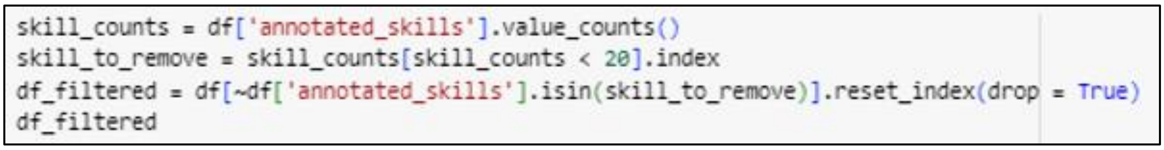
5.1 Introduction

This section the complex process of creating machine learning models to maximize skill set integration will be discussed. We travel the landscape of predictive modelling utilizing multiple machine learning paradigms, from text categorization preprocessing procedures to the use of sophisticated algorithms. This chapter unpacks the applied machine learning models, such as Multilabel Classification, OnevsRest Classifier, Logistic Regression, Gradient Boosting Machine (GBM), Random Forest Classifier, and Multinomial Naive Bayes (MNB), whether exploring supervised learning, unsupervised learning, or reinforcement learning. Together, these sections provide a thorough examination of how machine learning techniques might be used to implement the model to predict the skills required for banking sector employees.

5.2 Implementation of Machine Learning Models

This section will explain how the implementation of machine learning models has helped to predict the skills required from the job description. This has applied the NLP TF-IDF vectorizer to transform the text words into numerical form for the computer to understand the textual language. This has applied the machine learning models such as Logistic Regression, Gradient Boosting, Random Forest, and Multinomial Naïve Bayes, and they have compared their F1-score at the end to get the best-performing model. The predictions are made on the best-performing model based on their accuracy and printed the classification report to show the precision, recall, and F1-score for all the classes at the end.

Data pre-processing for modelling performed some text pre-processing steps before moving towards the final Machine learning modelling like reducing the count of less occurrence of annotated skills which does not have significant number of records for unique skills. Although it will be difficult to train the machine learning model on the skills and the model may underfit due to imbalance in the distribution of skills. Therefore, as indicated in figure 50, applied a multi-step process to deal with class imbalance in the annotated_skills column (Brownlee, 2019). This helps to improve the performance and generalization of the proposed machine learning models.



```
skill_counts = df['annotated_skills'].value_counts()
skill_to_remove = skill_counts[skill_counts < 20].index
df_filtered = df[~df['annotated_skills'].isin(skill_to_remove)].reset_index(drop = True)
df_filtered
```

Figure 50: Sample of Python Code to deal with class imbalance – Source: (Brownlee, 2019)

The next step is to identify skills that occur less frequently. This could be implemented by setting a threshold of 20 occurrences and finding the skills that fall below this threshold. Thereafter, assign these skills to the variable "skill_to_remove". Basically, what this line does is it gets rid of any skill sets that show up less than 20 times in the whole dataset. Here, it has used the .index attribute to get the specific skill sets instead of their counts.

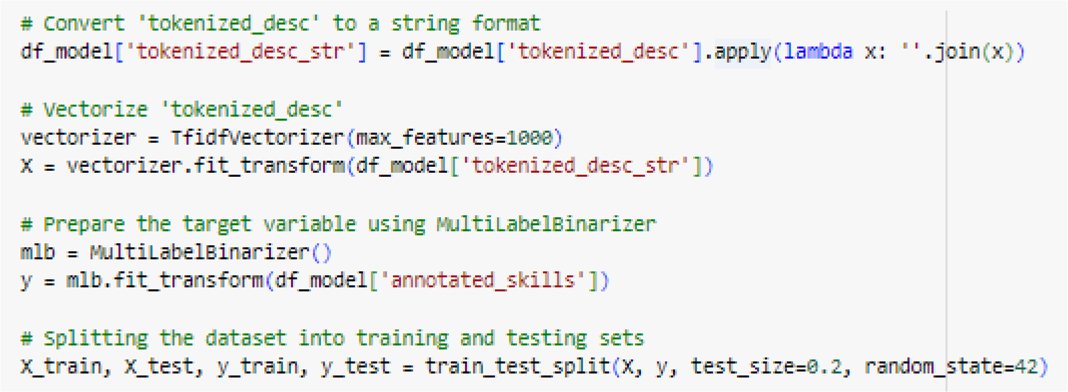
Finally, filter the DataFrame by creating a new DataFrame called df_filtered. This is accomplished by excluding rows where the values in the 'annotated_skills' exists in the list of to be removed. It will exclude any rows that contain the skill sets mentioned in the variable skill_to_remove. The ~ operator is used to flip the selection so that only rows with skill sets that are not in skill_to_remove are kept. Also, this applied reset_index(drop=True) to reset the index of the DataFrame and remove the previous index. This helps in cleaning and reorganizing the data structure. Basically, this whole process helps to address the problem of class imbalance by removing classes that occur less frequently. This can improve the model's ability to learn and make predictions based on the more common classes.

5.3 Preprocessing Steps for Text Classification in Machine Learning

The code provided in figure 51 explains several critical preparation processes intended for use on the DataFrame called df_model. These processes are designed to get the data suitable for a machine learning task, especially text categorization. The feature column, marked as the tokenized description, and the target column, representing the annotated skills, are both carefully considered in this preparation.

The first step is to transform the tokenized descriptions into string format. This modification is applied to the "tokenized_desc" column, which contains lists of tokens such as words or fragments. This transformation is required since it combines these tokens into a single string for each row. The reason for this is because most text vectorization approaches demand input data to be in string format, requiring this textual transformation.

To convert the text data from string format to numerical format, a technique called TF-IDF (Term Frequency-Inverse Document Frequency) vectorizer is used. TF-IDF is a quite common technique used in text processing. It is a process of figuring out how important a certain term is to a specific document within a larger collection of documents. When user set max_features=1000, user are specifically targeting the top 1000 terms that are most relevant.



```
# Convert 'tokenized_desc' to a string format
df_model['tokenized_desc_str'] = df_model['tokenized_desc'].apply(lambda x: ''.join(x))

# Vectorize 'tokenized_desc'
vectorizer = TfidfVectorizer(max_features=1000)
X = vectorizer.fit_transform(df_model['tokenized_desc_str'])

# Prepare the target variable using MultiLabelBinarizer
mlb = MultiLabelBinarizer()
y = mlb.fit_transform(df_model['annotated_skills'])

# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figure 51: Code for tokenizing to a string format

To prepare the target variable, it will use the MultiLabelBinarizer. It will help to convert the annotated_skills column, which may have multiple labels for each instance, into a binary matrix. It's important for tasks involving multi-label classification.

Finally, the dataset split into two sets: a training set and a testing set. This allows us to train the model on one part of the data and then validate it on another part. It's important to evaluate how well the model performs and how well it can apply what it has learned to new situations.

In general, it's crucial to do these preprocessing steps to turn the raw text data into a format that works well for training machine learning models, especially when it comes to tasks like text classification. Preprocessing is important to make sure that any machine learning model works well and gives accurate results.

5.4 Applied Machine Learning models

In this project, the chosen machine learning category is the supervised learning method with classification models, as the dataset was distributed into training and testing subsets. The applied Machine Learning algorithms are Logistic Regression, Gradient Boosting, Random Forest and Multinomial Naïve Bayes with the one vs Rest Classifier concept.

5.4.1 OnevsRest Classifier

The One-vs-Rest (OvR) strategy is also called One-vs-All, and is a technique used in multi-class classification. It helps to adapt binary classification algorithms for solving problems with multiple classes or labels. To do this, one can train a separate classifier for each class. For each classifier, it uses the samples of that class as positive examples and all other samples as negative examples. People often use this strategy when they want to apply a binary classification algorithm to a dataset that has a greater number of categories.

5.4.2 Logistic Regression

The fundamental mathematical principle that forms the basis of logistic regression is the logit, which is defined as the natural logarithm of the odds ratio. A basic illustration of a logit can be derived from a 2 × 2 contingency table. When examine a scenario where user have a binary outcome variable (if a child from an inner-city school is suggested for remedial reading classes) and a binary predictor variable (gender). Chi-square test of independence could be utilized. The obtained data indicates a chi-square value of 3.43, with 1 degree of freedom. Alternatively, one might choose to evaluate the likelihood of a boy being suggested for remedial reading instruction compared to a girl's likelihood. The outcome yields an odds ratio of 2.33, indicating that boys have a 2.33-fold higher likelihood of being referred for remedial reading programs in comparison to girls. The odds ratio is calculated by comparing the probabilities of an event occurring in two different groups. In this case, the odds ratio is determined from the probability of an event occurring in boys (73/23) and girls (15/11). The natural logarithm of the odds ratio, ln(2.33), is equal to 0.85, which is also known as a logit. The regression coefficient of the gender predictor in a logistic regression model for the two outcomes of a remedial recommendation, as it pertains to gender, would be 0.85 (Peng et al., 2002).

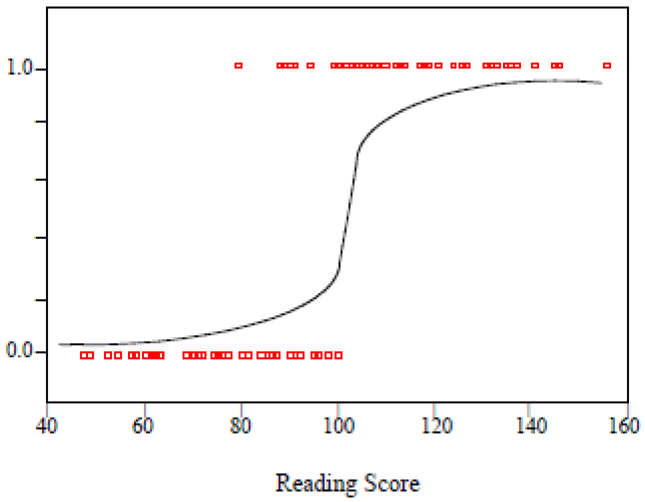


Figure 52: Logistic regression Sigmoid function

Logistic regression is a method, for examining and testing hypotheses about the relationships, between an outcome variable and one or more categorical or continuous predictor factors. In a linear regression scenario, we have one predictor variable X (a child's reading score on a standardized test) and one binary outcome variable Y (indicating whether the child is recommended for remedial reading classes). When we plot this data, we notice two lines, each representing an outcome value (refer to figure 52). Because of the nature of the outcomes, it becomes challenging to describe these lines using an ordinary least squares regression equation. Instead, one can categorize the predictor and calculate the mean of the outcome variable for each category. The resulting plot of the means of the categories will have a linear pattern in the middle, resembling what one would typically observe in a regular scatter plot. However, it will exhibit a curved shape at the ends, forming an S-shaped curve. Describing such a form, commonly known as sigmoidal or S-shaped, is challenging due to two reasons that make it incompatible with a linear equation. Initially, the extremes do not exhibit a linear pattern. Furthermore, the mistakes exhibit neither a normal distribution nor a consistent pattern throughout the entire range of data (Peng et al., 2001).

Integration of Logistic Regression with One-vs-Rest classification

One-vs-Rest: The One-vs-Rest strategy involves creating multiple binary logistic regression models, with each model representing a different class. If the dataset has N classes, then train N individual logistic regression models.

Training Process: The training process involves training a logistic regression model for each class Ci, where the model is trained against all the other classes. In this model, consider class Ci as the positive class (1) and subsequently group together all the other classes to form the negative class (0). Then repeat this process for each class, which gives N different logistic regression models.

Prediction: Each of the N logistic regression models forecasts the likelihood that a new instance will belong to its designated class when forecasting the class of the instance. The class for which the relevant logistic regression model predicts the highest probability is the final predicted class for the occurrence.

Limitation: The One-vs-Rest approach using logistic regression may not perform well if the classes cannot be easily separated linearly, as logistic regression is a type of linear classifier. Also, if there are a lot of classes, this method may not be very efficient because it needs to train many models.

5.4.3 Gradient Boosting Machine (GBM)

The boosting strategy shares similarities with the bagging approach, as it combines the base learners through voting. The primary distinction between the boosting and bagging approaches is in the incorporation of attributes from an additional classifier (Natekin & Knoll, 2013).

Gradient Boosting Machine (GBM) is a type of ensemble machine learning algorithm used for both regression and classification challenges. In GBM, the data is allocated weights. A learning algorithm is developed to train a model using a combination of weak learners. Therefore, the improvement is determined by the creation of the new base-learners. The parameters for GBM, are determined using appropriate parameters obtained from a parametric study on GBM. Afterwards, the GBM model is refined to assess its performance with different parameter sets. By conducting additional fine-tuning, the accuracy and performance of GBM can be improved to a satisfactory level on both the training and validation dataset. Efforts have been made to identify and optimize the key parameters that influence the default settings, the optimal number of trees, and the initial learning rate for GBM.

Gradient Boosting Machines (GBM) may still have over-fitting as a result of the number of iterations used to create a new model using a combination of weak learners that have been identified as new base-learners. Thus, GBM mitigates the risk of over-fitting by explicitly specifying the number of trees and the learning rate (Natekin & Knoll, 2013). To mitigate over-fitting, the learning rate is adjusted by decreasing its value to a range between 0 and 1. The depth parameter, known as the 'tree depth' controls how complex the tree is. A chosen tree depth allows the base learner to perform optimally and accurately capture the characteristics of the training data. Finding the balance, between tree complexity and the number of iterations is essential, for determining the tree depth.

Integration of GBM with One-vs-Rest classification

One-vs-Rest: The One-vs-Rest strategy works by training a separate model for each class. Each model is responsible for distinguishing one class from all the other classes. If dataset has N classes, then train N separate GBM models.

Training: During the training process, a GBM model for each class Ci. In this model, Ci is considered as the positive class, while all the other classes are combined to form the negative class. Then that user ends up with a binary classification problem for each class. When repeat the process for each class, which gives us N different GBM models.

Prediction: When making a prediction for a new instance, each of the N GBM models will give a score or probability for its own class. The predicted class for the instance is determined by the model that provides the highest score or probability.

Limitation: Using the OvR approach with GBMs can be quite computationally expensive, especially when dealing with many classes. This is because it requires training multiple models. Additionally, if the classes have a significant imbalance or if there are intricate interdependencies between them, this approach may not produce optimal outcomes.

5.4.4 Random Forest Classifier

The Random Forest algorithm, as described by Breiman (Breiman, 2001), is a method of combining classifiers which employ a set of L tree-structured base classifiers {h(X,ϴn), N=1,2,3,…L}. Here, X represents the input data and {ϴn} represents a group of vectors of randomness that are identical and dependent. Each Decision Tree is built by at random sampling data drawn from the presented dataset. A Random Forest, for example, can be built by randomly picking a subset of features for each Decision Tree (known as Random Subspaces) and/or randomly selecting a portion of training data for each Decision Tree (known as Bagging).

Within a Random Forest, the features undergo a random selection process during each decision split. Randomly picking features minimizes the connection between trees, enhancing prediction accuracy and increasing efficiency. The benefits of Random Forest are addressing the issue of overfitting, exhibit lower sensitivity towards outlier data, parameters setting is straightforward, and variable importance and accuracy are automatically generated.

The Random Forest algorithm incorporates the advantages of Decision Trees, such as their ability to make decisions based on many features. Additionally, it utilizes bagging to train on different subsets of the data and employs a voting method to make final decisions. By randomly selecting subsets of variables, Random Forest often outperforms Decision Trees in terms of accuracy (Ali et al., 2012).

The Random Forest algorithm is great for modelling data with variables since it can handle missing values effectively. It's also versatile, in dealing with types of data including continuous, categorical, and binary formats. The use of ensemble techniques in Random Forest improves its reliability making it effective in addressing overfitting concerns. As a result, there is no need, for tree pruning. Apart from its great accuracy in forecasting, Random Forest is also effective, explainable, and non-parametric when applied to different types of datasets (Devlin & Chang, 2018). The Random Forest algorithm stands out among popular machine learning approaches due to its distinctive combination of model interpretability and prediction accuracy. The application of collective techniques and arbitrary selection leads to more precise forecasts and improved simplifications (Breiman, 2001).

Integration of Random Forest with One-vs-Rest classification

One-vs-Rest: The One-vs-Rest strategy works by training a separate model for each class, where each model is trained to distinguish that class from all the other classes. If dataset has N classes, user train N individual Random Forest models.

Training: creating a Random Forest model for each class Ci. In each model, Ci is considered as the positive class, while all the other classes combine to create the negative class. When repeat this process for every class, which gives us N different Random Forest models. Each model is trained as a binary classifier.

Prediction: To predict the class of a new instance, each of the N Random Forest models will make a prediction on whether the instance belongs to its class or not. To figure out the final predicted class for instance, one can rely on the model that gives the highest probability or confidence in its prediction. Another option is to use a voting mechanism.

Limitation: The main downside is that it becomes more computationally expensive and complex because user have to train and store multiple models. Also, if there are a lot of classes, this method can become impractical.

5.4.5 Multinomial Naive Bayes (MNB)

A Multinomial Naive Bayes (MNB) classifier is a specific sort of Naive Bayes classifier that frequently serves as a benchmark for text classification tasks. MNB expects a document to be a collection of words and considers both word frequency and information. Naive Bayes classifiers are a group of classifiers that use Bayes' well-known probability theorem to construct straightforward yet efficient models, especially in the fields of document categorization & illness prediction. The primary use of NB is text categorization due to its efficiency and simplicity in implementation. Less flawed algorithms often exhibit slower performance and more complexity (Abbas et al., 2019).

The Naive Bayes algorithm is a kind of machine learning method. Its primary function is to classify text, which involves multidimensional sets of training data. Notable instances include document categorization, spam filtering, sentiment analysis. By using the naive Bayesian method, one may efficiently generate and forecast models. To determine the necessary parameters, a minimal quantity of training data is needed. The NB method is referred described as "naive" because of its assumption of feature independence, disregarding any potential correlation between the presentation of different characteristics.

Choosing the Naive Bayes classifier (NBC) is preferable because to its exceptional speed. It is recommended to utilize the Naive Bayes (NB) method for this issue due to its concurrent map-reduce implementation, which is particularly suitable for problems of this magnitude. NBC has achieved outstanding outcomes in the field of text analysis, namely in areas such as NLP (Abbas et al., 2019).

Integration of MNB with One-vs-Rest classification

One-vs-Rest: This strategy, also known as the OvR strategy, works by training a separate model for each class. Each model is designed to distinguish one specific class from all the other classes. If user have N classes, user train N separate MNB models.

Training: user train an MNB model for each class Ci. In this model, user consider Ci as the positive class, while combining all other classes to form the negative class. Thereafter, the multi-class problem is transformed into several separate binary classification problems. It repeats the process for each class, which gives N different MNB models.

Predictions: each of the MNB models calculates the probability of a new instance belonging to its specific class. The predicted class for the instance is the one that has the highest probability according to the corresponding MNB model.

Limitations: One thing to consider is that there is a downside to using this approach, and that is the number of computational resources it requires. One has to train and store multiple models, which can be quite costly. Also, it's worth noting that the assumption made by naive Bayes that features are independent may not always be accurate, which could impact how well each individual model performs.

5.6 Chapter Summary

The study shows how machine learning models use NLP and the TF-IDF vectorizer to turn job descriptions into numbers. This conversion allows to predict important skills. This study uses models such as Logistic Regression, Gradient Boosting, Random Forest, and Multinomial Naïve Bayes. This evaluates their performance by using the F1-score. After finding the best performing model, it is used to make predictions about the necessary job skills. Finally, a classification report is generated, which provides information about precision, recall, and F1-scores for each skill class.

To make sure that the model is accurate and handle the imbalance in classes, it is important to do data pre-processing. Skills with fewer occurrences have removed to make sure the model is not too simple or too complex, which helps improve the efficiency of training the model. The pre-processing steps involve converting the tokenized descriptions into strings and then using the TF-IDF vectorizer to transform the text data into a numerical format that can be used for machine learning tasks.

The TF-IDF mechanism is quite detailed. It explains how we calculate the term frequency and inverse document frequency to figure out the importance of terms in documents. Finally, we split the dataset into two subsets, one for training and one for testing. This is done for supervised learning. The dataset has 28 different classes, and we use multi-class classification for this task. The study uses advanced machine learning algorithms to accurately predict the skills that will be in demand in job markets.

Chapter 6: Results and Evaluation

6.1 Introduction

This chapter provides a comprehensive evaluation of both the expert assessment on data analysis and the performance evaluation of machine learning models.

6.2 Expert Feedback and Evaluation

Feedback from the industry experts stands as a keystone in refining the identified skills through text processing techniques. Invaluable insights of experts of the industry not only confirm the approach but also ensure the indicated abilities correspond smoothly with the sector's complex expectations.

During this step, expert opinions were gathered using a structured Google Form (See appendix D), which provided a thorough perspective of the recognized talents versus various job titles. The feedback form was designed to assess the comprehensiveness of the listed skills, identify any overemphasis or underemphasis, and solicit ideas for other critical skills within the sector. Furthermore, feedback was obtained based on the summary of data analysis provided as a report, which consisted of data visualizations detailed in chapter 4.

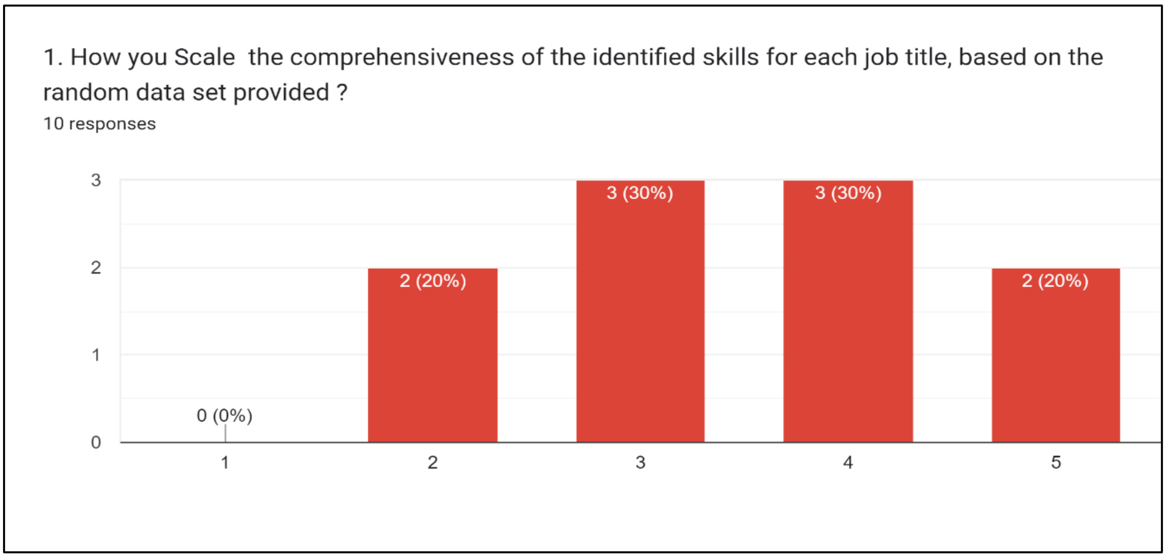
Table 4: Profile Information of Expert Evaluations

Experts	**Experience in Banking sector in Years**	**Job title**	**Specific areas within Banking sector specialized or have expertise**	**Have been involved in the hiring process or talent management**
Respond-1	10- 15 Years	HR Operations Manager	Human Resources Operations, Talent Management	Yes
Respond- 2	10- 15 Years	Branch Manager - Corporate Banking	Branch Banking, Branch Operations	No
Respond- 3	More than 15 Years	Enterprise Risk Manager	Treasury Management, Money Market, Risk Assessment	Yes
Respond- 4	1-5 Years	Senior Data Analyst	Core Banking Operations, IT	No
Respond-5	10- 15 Years	Credit Risk Analyst	Credit Risk, Branch Operations, Treasury Function	Yes
Respond-6	10- 15 Years	Senior Business Development Manager	Business Development, Customer Service, Branch Operations	Yes

Respond-7	More than 15 Years	Vice President (Branch Manager)	Branch Operations, Business Management	Yes
Respond-8	10- 15 Years	Personal Banker	Branch Operations, customer Service	No
Respond-9	More than 15 Years	Vice President Sales	Sales, Business Development, Talent Management	Yes
Respond- 10	5-10 Years	Investment Analyst	Business Development, Data analysis	Yes

The executive profiles who provided feedback on this research represent a wide range industry experience, from Human Resources Operations and Branch Banking to Risk Management and Data Analytics (refer table 5). Notably, 70% of these experienced professionals are actively participating in the recruiting process or personnel management, demonstrating the relevance and space of insights received from industry key decision-makers. This thorough portrayal offers a well-rounded view of the specified talents in the banking sector (See Appendix E).

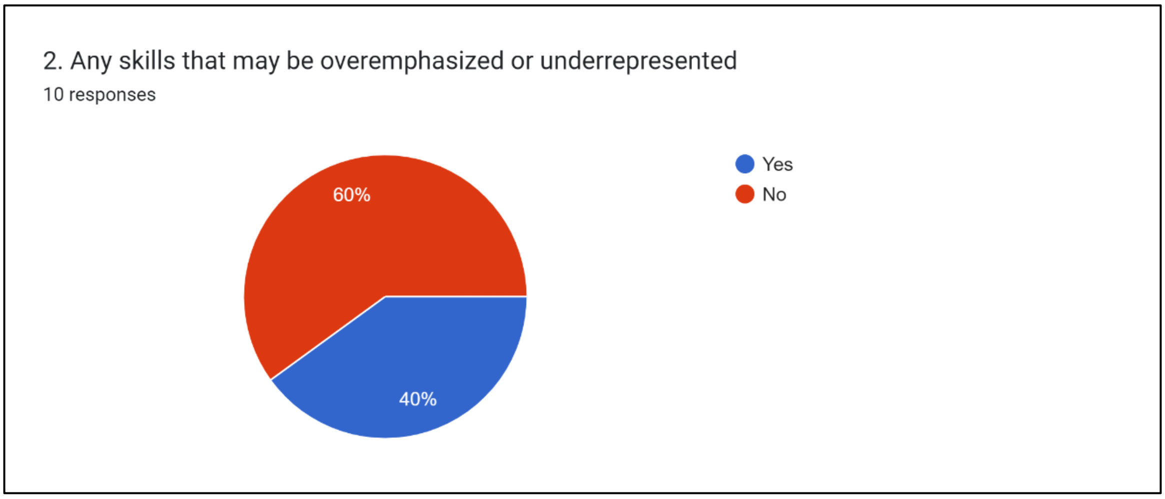
Respondents rated the comprehensiveness of indicated skills for each job title on a scale of 1 to 5, demonstrating the complicated nature of the dataset's skill comprehensiveness perceptions. However, 80% of professionals were rated on or above 3 scalers highlighting the comprehensiveness as valid as indicated in figure 53.


1. How you Scale the comprehensiveness of the identified skills for each job title, based on the random data set provided ?

10 responses

Figure 53: Expert Evaluation on Comprehensiveness of skills for each job tile

A significant 60% of experts acknowledged the possibility of overemphasis or underrepresentation in the indicated skill areas. When asked for additional insights, experts provided helpful suggestions, emphasizing the importance of skills such as Customer Service, Communication, Commercial Awareness, Business Acumen, Analytical Skills, the ability to work under pressure, and specific knowledge of Banking Systems (refer figure 54). These replies provide nuanced viewpoints on the perceived validity of stated talents, pointing out areas for improvement and introducing critical attributes highlighted by industry specialists.

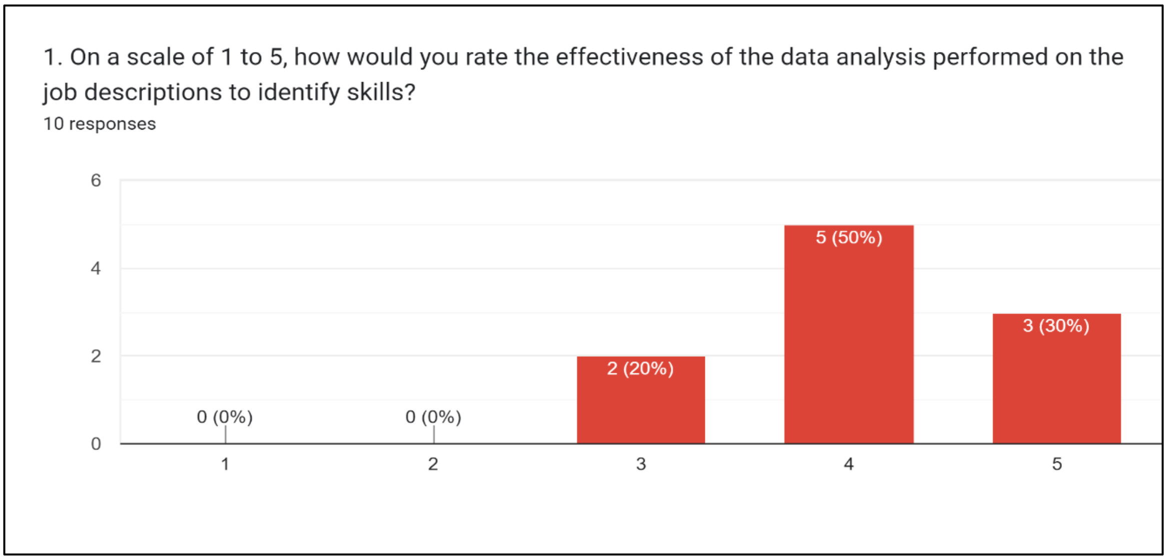


2. Any skills that may be overemphasized or underrepresented

10 responses

Figure 54: Expert feedback on overemphasis or underrepresentation of skills

On a scale of 1 to 5, the effectiveness of data analysis was rated positively. Respondents demonstrated strong performance, with 20% giving a rating of 3, 50% giving a rating of 4, and 20% giving the greatest degree of satisfaction with a rating of 5 (refer 55). These findings provide a substantial acknowledgement of the data analysis's usefulness in identifying skills inside job descriptions.



1. On a scale of 1 to 5, how would you rate the effectiveness of the data analysis performed on the job descriptions to identify skills?

10 responses

Figure 55: Expert Evaluation of effectiveness of data analysis

Overall, the feedback is encouraging, with experts typically confirming the validity of skill comprehensiveness and expressing pleasure with the usefulness of data analysis in finding skills inside job descriptions.

6.3 Performance Evaluation of Machine Learning Models

This section contains the information about the results obtained after applying different machine learning models. Different metrics like recall, precision, and F1-score have been used for evaluating the performance of all the models employed above in chapter 5.

6.3.1 Recall Analysis

The recall is calculated by dividing the count of samples correctly classified as Positive by the overall count of Positive samples (Swamynathan, 2019). The recall measure evaluates how well the model can identify examples. A higher recall indicates a greater number of positive samples discovered. The recall measure remains unaffected by the number of sample classifications. In addition, if the model correctly identifies all positive samples as positive, the Recall metric will have a value of 1.

Recall = True Positive/True Positive + False Negative

Recall = TP/ (TP+FN)

6.3.2 Precision Evaluation

Precision is the quotient obtained by dividing the number of properly identified positive samples (True Positive) by the total number of categorized positive samples (whether categorized correctly or wrongly). By using precision, it may evaluate the level of correctness shown by a machine learning model in classifying positive cases.

Precision = True Positive/True Positive + False Positive

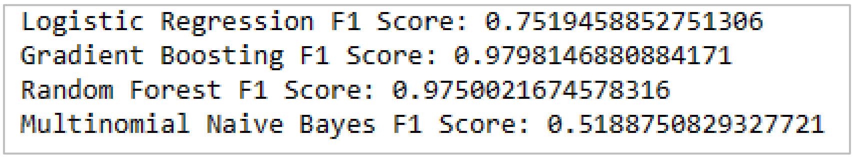
Precision = TP/TP+FP

6.3.3 F1– Score Assessment

The F1 score is basically a way to combine precision and recall into a single metric. The F1 score is often used to evaluate models in binary and multi-class classification. It combines precision and recall providing a comprehensive measure of the effectiveness of the model. The F1 score is calculated as the harmonic mean of the accuracy and recall scores. It assigns equal importance to both Precision and Recall when evaluating its correctness, making it a viable alternative to correctness metrics. It is often used as a singular metric that offers a concise assessment of the model's output accuracy.

F1 Score = 2* Precision Score * Recall Score/ (Precision Score + Recall Score)

Let’s understand the results of models created for the predictions of the annotated skills in multilabel dataset. From the below output (figure 56 and figure 57) user can understand that Gradient Boosting has performed best and performed all the other algorithms. However, the accuracy of GBM and Random Forest were very close. But the F1-score of Logistic Regression and MNB are significantly. While trying to choose a model for deployment, one would probably go with either Gradient Boosting or Random Forest based on these F1 scores. Although while developing the model one must consider other factors such as model complexity, training time, and interpretability.



```
Logistic Regression F1 Score: 0.7519458852751306
Gradient Boosting F1 Score: 0.9798146880884171
Random Forest F1 Score: 0.9750021674578316
Multinomial Naive Bayes F1 Score: 0.5188750829327721
```

Figure 56: F1 Score received for each model

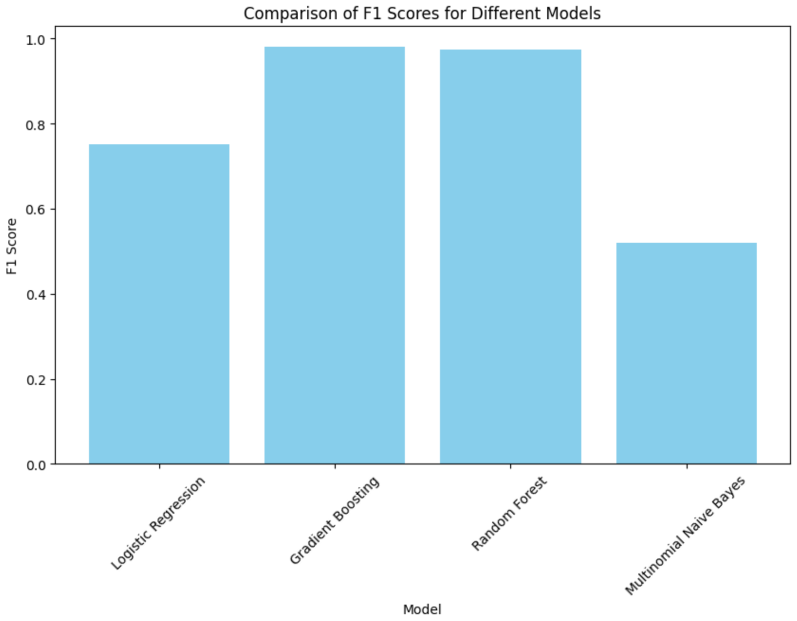
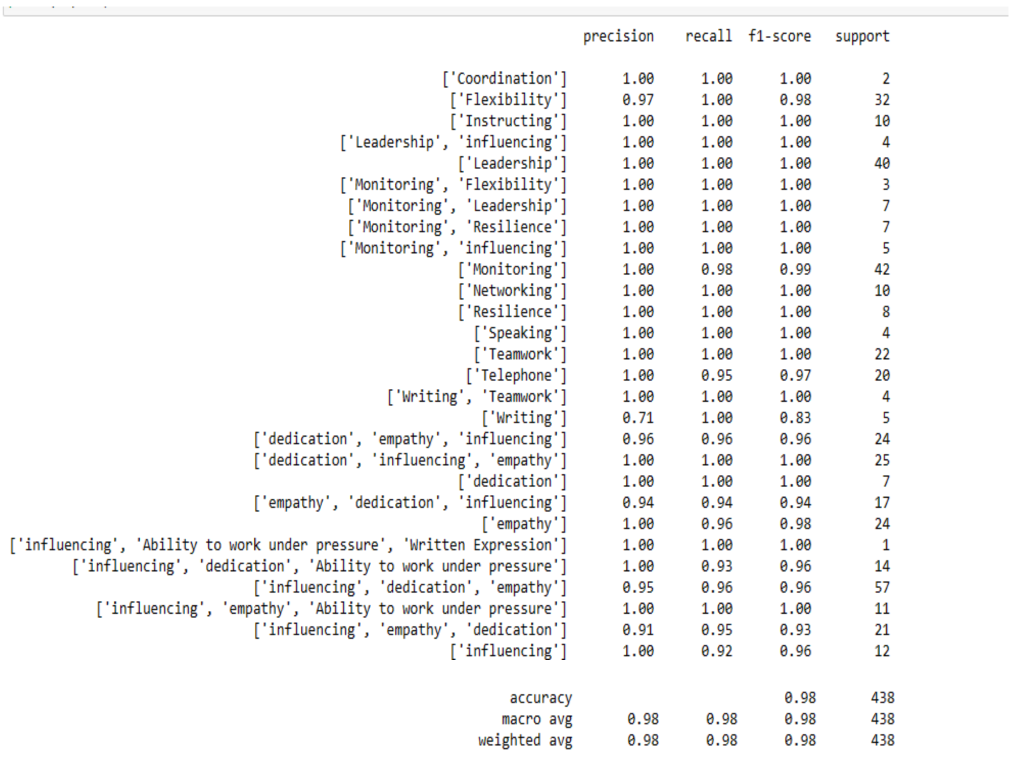


Figure 57: Comparison of F1 Scores for Different Models

After understanding that which model has performed best, the next step would be to do the predictions on the testing dataset, and to understand whether the model has performed well over all the 28 unique classes identified in 'annotated_skill' column of the dataset.



```
                                                                         precision    recall  f1-score   support

                                                       ['Coordination']       1.00      1.00      1.00         2
                                                        ['Flexibility']       0.97      1.00      0.98        32
                                                        ['Instructing']       1.00      1.00      1.00        10
                                          ['Leadership', 'influencing']       1.00      1.00      1.00         4
                                                         ['Leadership']       1.00      1.00      1.00        40
                                          ['Monitoring', 'Flexibility']       1.00      1.00      1.00         3
                                           ['Monitoring', 'Leadership']       1.00      1.00      1.00         7
                                           ['Monitoring', 'Resilience']       1.00      1.00      1.00         7
                                          ['Monitoring', 'influencing']       1.00      1.00      1.00         5
                                                         ['Monitoring']       1.00      0.98      0.99        42
                                                         ['Networking']       1.00      1.00      1.00        10
                                                         ['Resilience']       1.00      1.00      1.00         8
                                                           ['Speaking']       1.00      1.00      1.00         4
                                                           ['Teamwork']       1.00      1.00      1.00        22
                                                          ['Telephone']       1.00      0.95      0.97        20
                                                ['Writing', 'Teamwork']       1.00      1.00      1.00         4
                                                            ['Writing']       0.71      1.00      0.83         5
                               ['dedication', 'empathy', 'influencing']       0.96      0.96      0.96        24
                               ['dedication', 'influencing', 'empathy']       1.00      1.00      1.00        25
                                                         ['dedication']       1.00      1.00      1.00         7
                               ['empathy', 'dedication', 'influencing']       0.94      0.94      0.94        17
                                                            ['empathy']       1.00      0.96      0.98        24
['influencing', 'Ability to work under pressure', 'Written Expression']       1.00      1.00      1.00         1
        ['influencing', 'dedication', 'Ability to work under pressure']       1.00      0.93      0.96        14
                               ['influencing', 'dedication', 'empathy']       0.95      0.96      0.96        57
           ['influencing', 'empathy', 'Ability to work under pressure']       1.00      1.00      1.00        11
                               ['influencing', 'empathy', 'dedication']       0.91      0.95      0.93        21
                                                        ['influencing']       1.00      0.92      0.96        12

                                                               accuracy                           0.98       438
                                                              macro avg       0.98      0.98      0.98       438
                                                           weighted avg       0.98      0.98      0.98       438
```

Figure 58: Results Evaluation and Accuracy

The above classification report presented in Figure 58 provides a comprehensive overview of the model's performance across various metrics, including precision, recall, and F1-score. The remarkable scores attained in these metrics suggest that the model has learned the underlying patterns in the dataset effectively.

Since there is not substantial differences between training and testing performance indicators further supports the claim that the model has not encountered issues of overfitting or underfitting. Overfitting happens when a model becomes too focused on the training data, even the noisy and unusual parts, and ends up not performing well on new, unseen data, while underfitting is characterized by poor performance on both training and testing sets. However, the consistently high precision, recall, and F1-score values across multiple classes indicate a balanced and robust model.

The Gradient Boosting Machine (GBM) model has demonstrated exceptional performance across all classes in the dataset. Its ability to generalize well to unseen instances, as evidenced by the consistent and high performance on the testing set, is indicative of a model that has not memorized the training data but has learned to capture meaningful patterns.

Chapter 7: Conclusions and Future Work

7.1 Introduction

The findings of this research represent a significant advancement in the management of the banking industry's workforce, driven by the development of a robust framework for skill identification. This chapter summarizes the key findings, implications, and outlines potential future research directions.

7.2 Review of Research Aims and Objectives

The research objectives were meticulously structured to address crucial aspects of skill identification using NLP techniques. Each objective was aimed at enhancing our understanding of skill requirements in the banking industry and developing an effective framework for skill extraction and prediction.

Objective 1 - Investigate existing literature on NLP techniques and their applications in talent management within the banking industry, aiming to inform the development of a robust skill identification process.

Achievement: The research conducted a thorough review of existing literature on NLP techniques, specifically focusing on their relevance to talent management within the banking industry. This review provided valuable insights into the theoretical foundations of NLP and its practical applications in skill identification processes. By synthesizing relevant studies and methodologies, the research established a solid theoretical framework for subsequent objectives.

Objective 2 - Develop a comprehensive framework for identifying the competencies required by banking industry professionals through data extraction from UK job postings, leveraging web-based annotation tools and NLP techniques.

Achievement: Building upon the insights gained from the literature review, the research developed a comprehensive framework for skill identification in the banking sector. This framework integrated web-based annotation tools and NLP techniques to extract and categorize competencies from job postings sourced from UK websites. By systematically organizing the extracted data, the framework provided a structured approach for analysing and understanding the skill requirements of banking industry professionals.

Objective 3 - Utilize advanced algorithms and NLP approaches to automatically extract and categorize skills from unstructured textual data obtained from banking sector job advertisements, ensuring accuracy and relevance to industry needs.

Achievement: The research implemented advanced algorithms and NLP approaches to automate the process of skill extraction from unstructured textual data obtained from banking sector job advertisements. By leveraging techniques such as Bidirectional Encoder Representations from Transformers (BERT), the research ensured the accuracy and relevance of the extracted skills to the specific needs of the banking industry. This automated approach facilitated the efficient analysis of large volumes of textual data, enhancing the scalability and effectiveness of the skill identification process.

Objective 4 - Implement and evaluate various machine learning models within the established framework to predict the skills required for banking sector employees, utilizing metrics such as recall, precision, and F1-score to assess the effectiveness of the models.

Achievement: The research implemented a range of machine learning models within the established framework to predict the skills required for banking sector employees. These models were trained and evaluated using metrics such as recall, precision, and F1-score to assess their effectiveness in skill prediction. By comparing the performance of different models, the research identified the most accurate and reliable approaches for predicting banking-related skills, providing valuable insights for talent management and recruitment processes in the banking industry. The models demonstrated remarkable accuracy and performance, particularly Gradient Boosting, which showcased exceptional F1 scores.

7.3 Implications and Future Work

The study's implications extend beyond academia, offering practical insights for job seekers and facilitating more effective recruitment processes for banking institutions. Future research directions include improving the accuracy and context-awareness of skill extraction, addressing biases in job descriptions, and enhancing personalization in the skill matching process.

7.4 Contribution of the Research

By considering multiple factors, this study fills a vacuum by offering scholarly perspectives on automating the personnel management procedure in the banking sector. By using this method, a macro-level database of talents relevant to various banking industry occupations can be created. As a result, our research helps talent managers minimize subjectivity and prejudice in the hiring process by effectively identifying the necessary talents for job categories. Furthermore, the baseline model can be modified and updated on a regular basis to account for changing trends in the labour market. This approach is easily adaptable to both domestic and global personnel management systems and may be effectively duplicated in various industries.

In conclusion, this research provides a structured approach to continuously improving skill identification processes in specific industries, laying the foundation for more efficient and fair talent management practices.

8. References

Abbas, M., Ali, K., Memon, S., Jamali, A., Memon, S., & Ahmed, A. (2019). Multinomial Naive Bayes classification model for sentiment analysis. https://doi.org/10.13140/RG.2.2.30021.40169

Alex Graves. Sequence Transduction with Recurrent Neural Networks. Department of Computer Science, University of Toronto, Canada. 14 Nov 2012. https://arxiv.org/abs/1211.3711. Accessed 22 Sept 2023.

Alexis Megan Votto, R. V., Peyman Najafirad, H. Raghav Rao. (2021). Artificial Intelligence in Tactical Human Resource Management: A Systematic Literature Review. International Journal of Information Management Data Insight.

Ali, J., Khan, R., Ahmad, N., & Maqsood, I. (2012). Random Forests and Decision Trees. International Journal of Computer Science Issues (IJCSI), 9.

Anderson, R. W., & Joeveer, K. (2022). Bankers' pay and the evolving structure of US banking. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.4022036

Aqel, D. M. A. (2014). A framework for employee appraisals based on inductive logic programming and data mining methods ProQuest Dissertations & Theses]. Ann Arbor.

Arumugam, R., & Shanmugamani, R. (2018). Hands-on natural language processing with Python : a practical guide to applying deep learning architectures to user NLP applications (1st edition ed.). Packt Publishing.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. 12 Jun 2017. https://arxiv.org/abs/1706.03762. Accessed 23 Sept 2023.

Barducci, A., Iannaccone, S., La Gatta, V., Moscato, V., Sperlì, G., & Zavota, S. (2022). An end-to-end framework for information extraction from Italian resumes. Expert Systems with Applications, 210, 118487. https://doi.org/https://doi.org/10.1016/j.eswa.2022.118487

Bastian, M.; Hayes, M.; Vaughan, W.; Shah, S.; Skomoroch, P.; Kim, H.; Uryasev, S.; and Lloyd, C. 2014. LinkedIn Skills: Large-Scale Topic Extraction and Inference. In Proceedings of the 8th ACM Conference on Recommender Systems, 1–8. New York: Association for Computing Machinery.

Beliga, S. (2014). Keyword extraction: a review of methods and approaches. University of Rijeka, Department of Informatics, Rijeka, 1(9).

Bengio Y., Simard P., Frasconi P., Learning Long-Term Dependencies with Gradient Descent is Difficult. 2 March 1994. http://ai.dinfo.unifi.it/paolo//ps/tnn-94-gradient.pdf. Accessed 23 Sept 2023.

Biswas, S.Kr., Bordoloi, M. and Shreya, J. (2018) ‘A graph based keyword extraction model using collective node weight’, Expert Systems with Applications, 97, pp. 51–59. doi:10.1016/j.eswa.2017.12.025.

Bondielli, A. and Marcelloni, F. (2021) ‘On the use of summarization and transformer architectures for profiling résumés’, Expert Systems with Applications, 184, p. 115521. doi:10.1016/j.eswa.2021.115521.

Boselli, R., Cesarini, M., Mercorio, F., & Mezzanzanica, M. (2018). Classifying online Job Advertisements through Machine Learning. Future generation computer systems, 86, 319-328. https://doi.org/10.1016/j.future.2018.03.035

Breugel, G. v. (2017). Identification and anticipation of skill requirements

Brownlee, J. (2019, August 7). How to Clean Text for Machine Learning with Python. MachineLearningMastery.com. https://machinelearningmastery.com/clean-text-machine-learning-python/

C. Zahang, H. Wang, Y. Liu, D. Wu, Y. Liao, B. Wang, “Automatic Keyword Extraction from Documents Using Conditional Random Fields” in Journal of CIS 4:3(2008), pp. 1169-1180, 2008.

Califf, M. E., & Mooney, R. J. (2003). "Bottom-Up Relational Learning of Pattern Matching Rules for Information Extraction." Journal of Machine Learning Research, 4, 177-210.

Canning, J., & Found, P. A. (2015, June 15). The effect of resistance in organizational change programmes. International Journal of Quality and Service Sciences, 7(2/3), 274–295. https://doi.org/10.1108/ijqss-02-2015-0018

Changmao Li, E. F., Rebecca Thomas, Steve Pittard, Vicki Hertzberg, Jinho D. Choi. (2020). Competence-Level Prediction and Resume & Job Description Matching Using Context-Aware Transformer Models.

Chernova, M. (2020) Occupational skills extraction with FinBERT, pp. 16–17.

City of London Corporation. (2022). Financial service hubs around the UK. Retrieved from [https://www.cityoflondon.gov.uk/supporting-businesses/economic-research/research-publications/financial-services-hubs-around-the-uk]

Clark, T., Foster, L., Sloan, L., & Bryman, A. (2021). Bryman's social research methods (Sixth edition. ed.). Oxford University Press.

Collings, D. G., Wood, G. T., & Szamosi, L. T. (Eds.). (2018). Human Resource Management: A Critical Approach (2nd ed.). Routledge. 10.4324/9781315299556 .

Creswell, J. W., & Creswell, J. D. (2018). Research design : qualitative, quantitative, and mixed method approaches (Fifth edition. ed.). SAGE.

Dario Radecic, Softmax Activation Function Explained, 18 Jun 2020. https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60. Accessed: 3 Oct 2023

Deng, L. and Liu, Y. (2017) Deep Learning in Natural Language Processing. Available at: https://lidengsite.files.wordpress.com/2018/03/chapter-6.pdf (Accessed: 31 May 2023).

Deng, L., & Liu, Y. (2018). Deep Learning in Natural Language Processing. Springer Singapore Pte. Limited. https://doi.org/10.1007/978-981-10-5209-5

Devlin, J. und Chang, M.-W. (2018) Google AI Blog: Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing, Google AI Blog.Available here: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html. Accessed: 4 Oct 2023

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. https://arXivpreprint arXiv:1810.04805. Accessed: 9 Oct 2023

Di Vaio, A., Palladino, R., Hassan, R., & Escobar, O. (2020). Artificial intelligence and business models in the sustainable development goals perspective: A systematic literature review. Journal of Business Research . 10.1016/j.jbusres.2020.08.019 .

Editor, S.B.F. et al. (2021) 3 Ways Natural Language Processing (NLP) will transform HR in 2019, Spiceworks. Available at: https://www.spiceworks.com/hr/hr-strategy/articles/3-ways-natural-language-processing-nlp-will-transform-hr-in-2019/ (Accessed: 20 July 2023).

ESCO: European Skills, Competences, Qualifications and Occupations https://ec.europa.eu/esco/ . Accessed 17 Dec 2015. Madely du Preez

Fernandes, J. (2022). Transformers: Text Classification for NLP Using BERT. linkedin.com.

Global City. (2022). State of the Sector: Annual Review of UK Financial Services 2022. Retrieved from [https://www.theglobalcity.uk/PositiveWebsite/media/Research-reports/State-of-the-sector_annual-review-of-UK-financial-services-2022.pdf]

Goyal, A., Gupta, V. and Kumar, M. (2018) ‘Recent named entity recognition and Classification Techniques: A Systematic Review’, Computer Science Review, 29, pp. 21–43. doi:10.1016/j.cosrev.2018.06.001.

Goyal, A., Gupta, V., & Kumar, M. (2018). Recent Named Entity Recognition and Classification techniques: A systematic review. Computer Science Review, 29, 21-43. https://doi.org/https://doi.org/10.1016/j.cosrev.2018.06.001

Gruetzemacher, R. (2022). The Power of Natural Language Processing. Harvard Business Review.

Gugnani, A. Implicit Skills Extraction Using Document Embedding and Its Use in Job Recommendation. Conference: AAAI - Innovative Applications of Artificial Intelligence (IAAI). February 2020.

Guliano Giacaglia. How Transformers Work. The Neural Network used by Open AI and DeepMind. 11 March 2019. https://towardsdatascience.com/transformers-141e32e69591. Accessed 2 Oct 2023

Hakim, A. A., Erwin, A., Eng, K. I., Galinium, M., & Muliady, W. (2015). Automated document classification for news article in Bahasa Indonesia based on term frequency inverse document frequency (TF-IDF) approach. In 6th International Conference on Information Technology and Electrical Engineering: Leveraging Research and Technology, (ICITEE), 2014.

Hmoud, B. I. , & Várallyai, L. (2020). Artificial Intelligence in Human Resources Information Systems: Investigating its Trust and Adoption Determinants. International Journal of Engineering and Management Sciences, 5 (1), 749–765 .

Hoang, P., Mahoney, T., Javed, F., & McNair, M. (2018, March). Large-Scale Occupational Skills Normalization for Online Recruitment. AI Magazine, 39(1), 5–14. https://doi.org/10.1609/aimag.v39i1.2775

Instruments used by international institutions and developed countries https://repositorio.cepal.org/bitstream/handle/11362/42233/S1700483_en.pdf?sequence=1&isAllowed=y

International Monetary Fund. (2022). World Economic Outlook Report October 2022. Retrieved from [https://www.imf.org/en/Publications/WEO/Issues/2022/10/11/world-economic-outlook-october-2022]

Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Google AI Language. 24 May 2019. https://arxiv.org/pdf/1810.04805.pdf. Accessed 28 Sept 2023.

Jacob Devlin, Ming-Wei Chang, Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing, Google AI Language, 2 Nov 2018,https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html. Accessed: 3 Oct 2023.

Jagannathan, M., Roy, D., & Delhi, V. S. K. (2022). Application of NLP-based topic modelling to analyse unstructured text data in annual reports of construction contracting companies. CSI Transactions on ICT, 10(2), 97-106. https://doi.org/10.1007/s40012-022-00355-w

Jain, S.M. (2022) ‘Bert’, Introduction to Transformers for NLP, pp. 37–49. doi:10.1007/978-1-4842-8844-3_3.

Jaspreet Singha, G. S., Muskan Gahlawatb, Chander Prabha. (2023). Big Data as a Service and Application for Indian Banking Sector 4th International Conference on Innovative Data Communication Technology and Application, Jay Alammar, The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning) https://jalammar.github.io/illustrated-bert/ . Accessed 29 Sept 2023.

Job search | indeed. Available at: https://uk.indeed.com/ Accessed: 10 Oct 2023.

Jurafsky, D., & Martin, J. H. (2009). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Pearson. [Chapter 2: Regular Expressions, Text Normalization, and Edit Distance]

Kastrati, Z., Dalipi, F., Imran, A. S., Nuci, K. P., & Wani, M. A. (2021). Sentiment analysis of students' feedback with nlp and deep learning: A systematic mapping study. Applied sciences, 11(9), 3986. https://doi.org/10.3390/app11093986

Kivimäki, I.; Panchenko, A.; Dessy, A.; Verdegem, D.; Francq, P.; Fairon, C.; Bersi- ni, H.; and Saerens, M. 2013. A Graph-Based Approach to Skill Extraction from Text. Paper presented at the Graph-Based Methods for Natural Language Processing workshop, 18 October, Seattle, WA.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text Classification Algorithms: A Survey. Information, 10(4).

Ku, L. (2021) An introduction of NLP and how it's changing the future of HR, Plug and Play Tech Center. Available at: https://www.plugandplaytechcenter.com/resources/introduction-nlp-and-how-its-changing-future-hr/ (Accessed: 10 July 2023).

Kuhlman, D. (2012, April 22). A Python Book: Beginning Python, Advanced Python, and Python Exercises. Retrieved from https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python_book_01.html

Kuodytė, V., & Petkevičius, L. (2021). Education-to-Skill Mapping Using Hierarchical Classification and Transformer Neural Network. Applied sciences, 11(13), 5868. https://doi.org/10.3390/app11135868

Liu, B. (2012). "Sentiment Analysis and Opinion Mining." Synthesis Lectures on Human Language Technologies, 5(1), 1-167.

Lubanovic, B. (2019) Introducing python: Modern computing in simple packages. Sebastopol, CA: O'Reilly Media, Inc.

Lubanovic, B. (2019). Introducing Python: Modern Computing in Simple Packages (Second edition). O'Reilly Media, Incorporated.

Madely du Preez. Taxonomies, folksonomies, ontologies: what are they and how do they support information retrieval? The Indexer The International Journal of Indexing 33. March 2015

Majumder, S., & Mondal, A. (2021). Are chatbots really useful for human resource management? International Journal of Speech Technology, 24(4), 969–977. https://doi.org/10.1007/s10772-021-09834-y

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

Mazurchenko, A., Zelenka, M., & Maršíková, K. (2022, June). DEMAND FOR EMPLOYEES' DIGITAL SKILLS IN THE CONTEXT OF BANKING 4.0. E+M Ekonomie a Management, 25(2), 41–58. https://doi.org/10.15240/tul/001/2022-2-003

MediaWiki API help. https://en.wikipedia.org/w/api.php Accessed 15 Sept 2023

Monsters, D. (2022, June 28). Text Preprocessing in Python: Steps, Tools, and Examples. Medium. https://medium.com/product-ai/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908

Montejo-Ráez, A., Montejo-Ráez, A., & Jiménez-Zafra, S. M. (2022). Current Approaches and Applications in Natural Language Processing. MDPI Books.

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: an introduction. Journal of the American Medical Informatics Association : JAMIA, 18(5), 544-551. https://doi.org/10.1136/amiajnl-2011-000464

National Research Council. (2012). Education for Life and Work: Developing Transferable Knowledge and Skills in the 21st Century. Washington, DC: The National Academies Press.

O*NET Analyst Ratings of Occupational Skills: Analysis Cycle 23 Results Appendix B Descriptive Statistics for O*NET Analysis Cycle 23 Importance and Level Ratings by Occupations. (n.d.).

Och, F. J. (2003). "Minimum Error Rate Training in Statistical Machine Translation." In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL).

Office for National Statistics. (2022). Labour Market Overview UK, November 2022. Retrieved from [https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/employmentandemployeetypes/bulletins/uklabourmarket/november2022]

Organisation for Economic Co-operation and Development. (2021). OECD Economic Outlook, Interim Report September 2021. Retrieved from [https://www.oecd.org/economic-outlook/september-2021]

Patrick, D. S. (2018). Hands-on artificial intelligence for beginners : an introduction to AI concepts, algorithms, and their implementation (1st edition ed.). Packt.

Peng, C. Y., Manz, B. D., & Keck, J. (2001). Modelling categorical variables by logistic regression. American Journal of Health Behavior, 25(3), 278–284.

Peng, C.-Y. J., Lee, K. L., &amp; Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. The Journal of Educational Research, 96(1), 3–14. https://doi.org/10.1080/00220670209598786

Popov, D., Snelson, S., & Baily, T. (2022). Review of skills taxonomies Report prepared for the Skills and Productivity Board.

Qaiser, S., &amp; Ali, R. (2018). Text mining: Use of TF-IDF to examine the relevance of words to documents. International Journal of Computer Applications, 181(1), 25–29. https://doi.org/10.5120/ijca2018917395

Rahul, Adhikari, S., & Monika. (2020, 11-13 March 2020). NLP based Machine Learning Approaches for Text Summarization. 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC),

Razno, M. (2019). Machine learning text classification model with NLP approach. Computational Linguistics and Intelligent Systems, 2, 71-73.

Reese, R., & Bhatia, A. (2018). Introduction to NLP. In Natural Language Processing with Java. Packt Publishing, Limited.

Renu, B. (2021). Basic and Modern Employability Skills: Bankers' Viewpoint. The ICFAI journal of soft skills, 15(3), 30-42.

Ricks, D. A., Toyne, B., & Martinez, J. A. (1990). The Role of Literature Review in the Research Process. In Annual Meeting of the Association for Information Systems (Vol. 1, p. 220)

Robles, M. M. (2012). Executive perceptions of the top 10 soft skills needed in today's workplace. Business Communication Quarterly, 75(4), 453-465.

Roy D & Jagannathan M (2021) Exploring the reach of lean philosophy in indian construction industry. In Proceedings of the fourth biennial conference of the Indian Lean Community Indian Lean Construction Conference (pp. 203–212). Ahmedabad: CEPT University Press.

Selenium. Available at: https://www.selenium.dev/ Accessed: 10 Oct 2023.

Selimović, J., Pilav-Velić, A., & Krndžija, L. (2021). Digital Workplace Transformation in the financial service sector: Investigating the relationship between employees’ expectations and intentions. Technology in Society, 66, 101640. https://doi.org/10.1016/j.techsoc.2021.101640

Sharma, A., Singhal, S., & Ajudia, D. (2021, 24-26 Sept. 2021). Intelligent Recruitment System Using NLP. 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV),

Sharma, N. Job Skills extraction with LSTM and Word Embeddings. University of Technology Sydney - UTS, Sydney, Australia. Sept 2019

Shubham Singh (2023, May 22). How to Get Started with NLP – 6 Unique Methods to Perform Tokenization. https://www.analyticsvidhya.com/blog/2019/07/how-get-started-nlp-6-unique-ways-perform-tokenization

Smith, P. D. (2018). Hands-on Artificial Intelligence for Beginners: An introduction to ai concepts, algorithms, and their implementation. Packt Publishing.

Speer, A. B. (2020). Scoring Dimension-Level Job Performance From Narrative Comments: Validity and Generalizability When Using Natural Language Processing.

Swamynathan, M. (2019) 'Step 2: Introduction to machine learning', Mastering Machine Learning with Python in Six Steps, pp. 65–143. doi:10.1007/978-1-4842-4947-5_2.

Taoffik, O. B., Oyintola Isiaka, A., Akinkunmi Oluseun, O., & Olukunle Akinbola, O. (2016). Influence of Information Literacy Skills on Information Needs and use among Banking Personnel in Ogun State, Nigeria. Library philosophy and practice, 1.

The Lightcast Open Skills Taxonomy Understanding a Fast-Changing Labor Market. (n.d.).

Treasury, H. (2022). State of the sector: annual review of UK financial Services 2022. London: HM Treasury. Retrieved from https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1092788/State_of_the_sector_annual_review_of_UK_financial_services_2022.pdf

Tunstall, L., Von Werra, L., & Wolf, T. (2022). Natural language processing with transformers. " O'Reilly Media, Inc.".

Tunstall, L., Werra, L. von and Wolf, T. (no date) Natural language processing with transformers, Revised edition, O'Reilly Online Learning. Available at: https://www.oreilly.com/library/view/natural-language-processing/9781098136789/ch01.html (Accessed: 15 July 2023).

Vidya, A. (2020). Beginners Guide to Topic Modelling in Python. https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modelling-in-python/

Wang, Z.; Li, S.; Shi, H.; and Zhou, G. 2014. Skill Inference with Personal and Skill Connections. In Proceedings of the 25th International Conference on Computational Linguistics (COLING), 520–529. Stroudsberg, PA: Association for Computational Linguistics.

Weber, J. R. (2002). "Speech and Language Processing: Creating Robust and Effective Tools for Speech Recognition." In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP).

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., & Funtowicz, M. (2020). Transformers: State-of-the-art natural language processing. Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations,

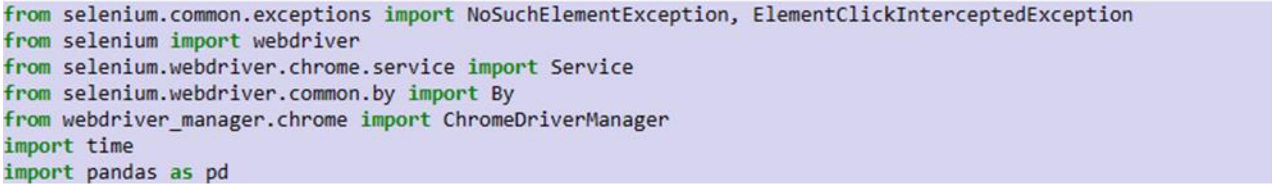
Y. Wu, M. Schuster, Z. Chen, Q. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes and J. Dean Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, 2016.

Yue Kang, Z. C., Chee-Wee Tan, Qian Huang & Hefu Liu. (2020). Natural language processing (NLP) in management research: A literature review. Journal of Management Analytics.

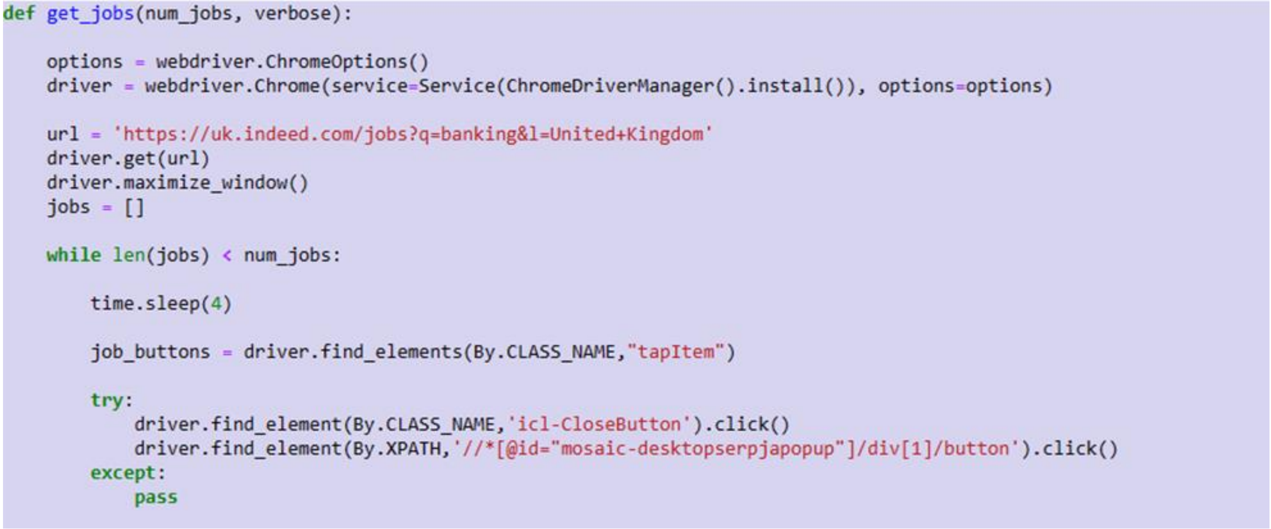
Zong, C., Zhang, J., & Xia, R. (2021). Text data mining. Springer.

9. Appendices

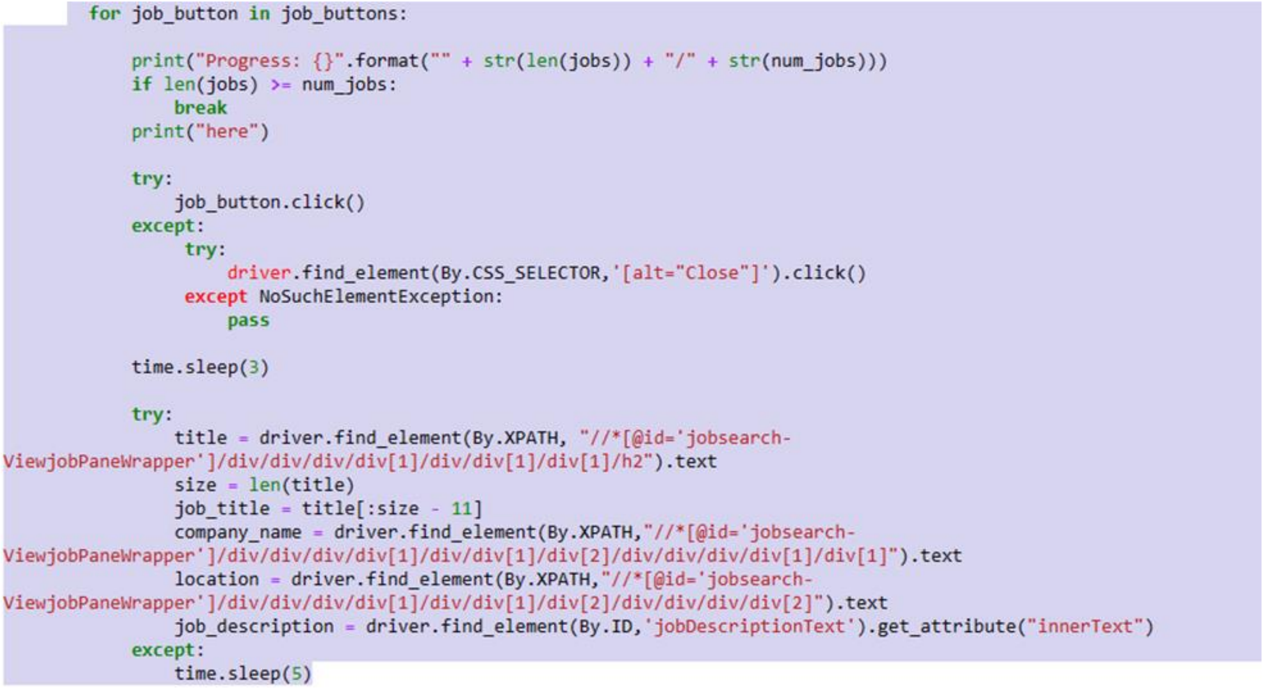
Appendix A - Python Code Implementation for Web Scraping



```python
from selenium.common.exceptions import NoSuchElementException, ElementClickInterceptedException
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from webdriver_manager.chrome import ChromeDriverManager
import time
import pandas as pd
```

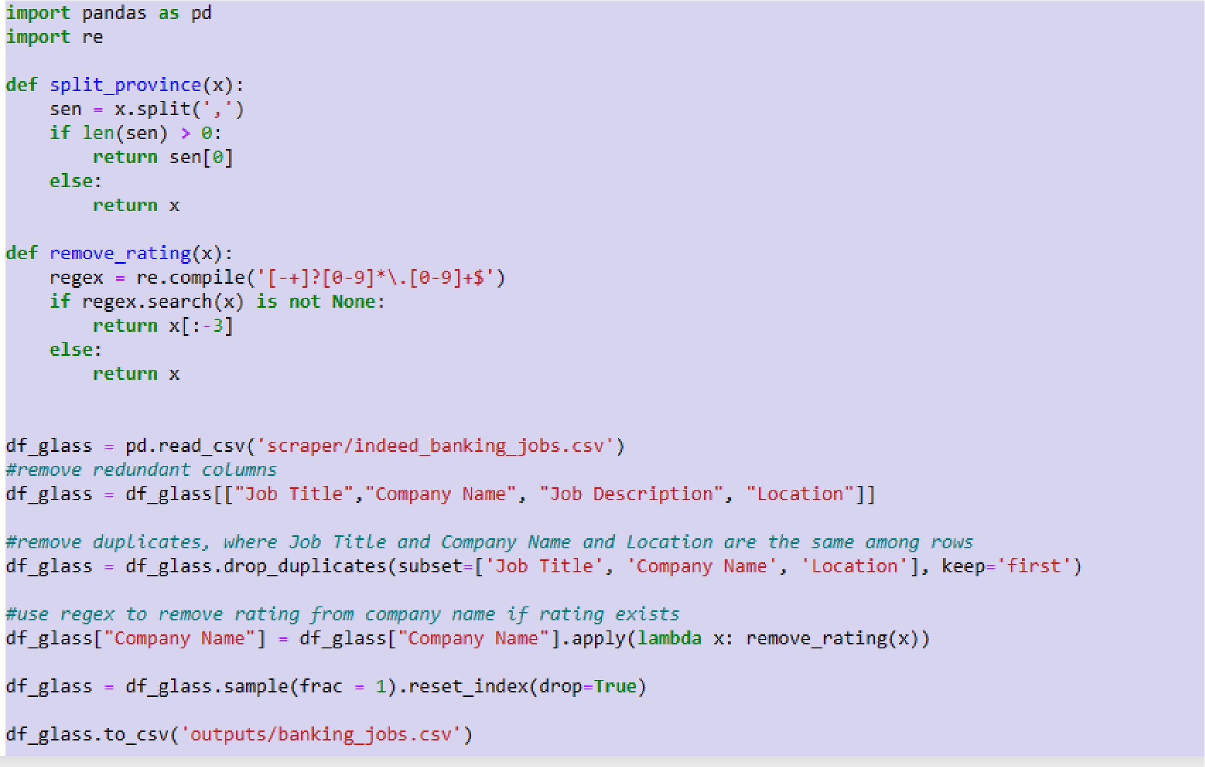


```python
def get_jobs(num_jobs, verbose):

    options = webdriver.ChromeOptions()
    driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=options)

    url = 'https://uk.indeed.com/jobs?q=banking&l=United+Kingdom'
    driver.get(url)
    driver.maximize_window()
    jobs = []

    while len(jobs) < num_jobs:

        time.sleep(4)

        job_buttons = driver.find_elements(By.CLASS_NAME,"tapItem")

        try:
            driver.find_element(By.CLASS_NAME,'icl-CloseButton').click()
            driver.find_element(By.XPATH,'//*[@id="mosaic-desktopserpjapopup"]/div[1]/button').click()
        except:
            pass
```



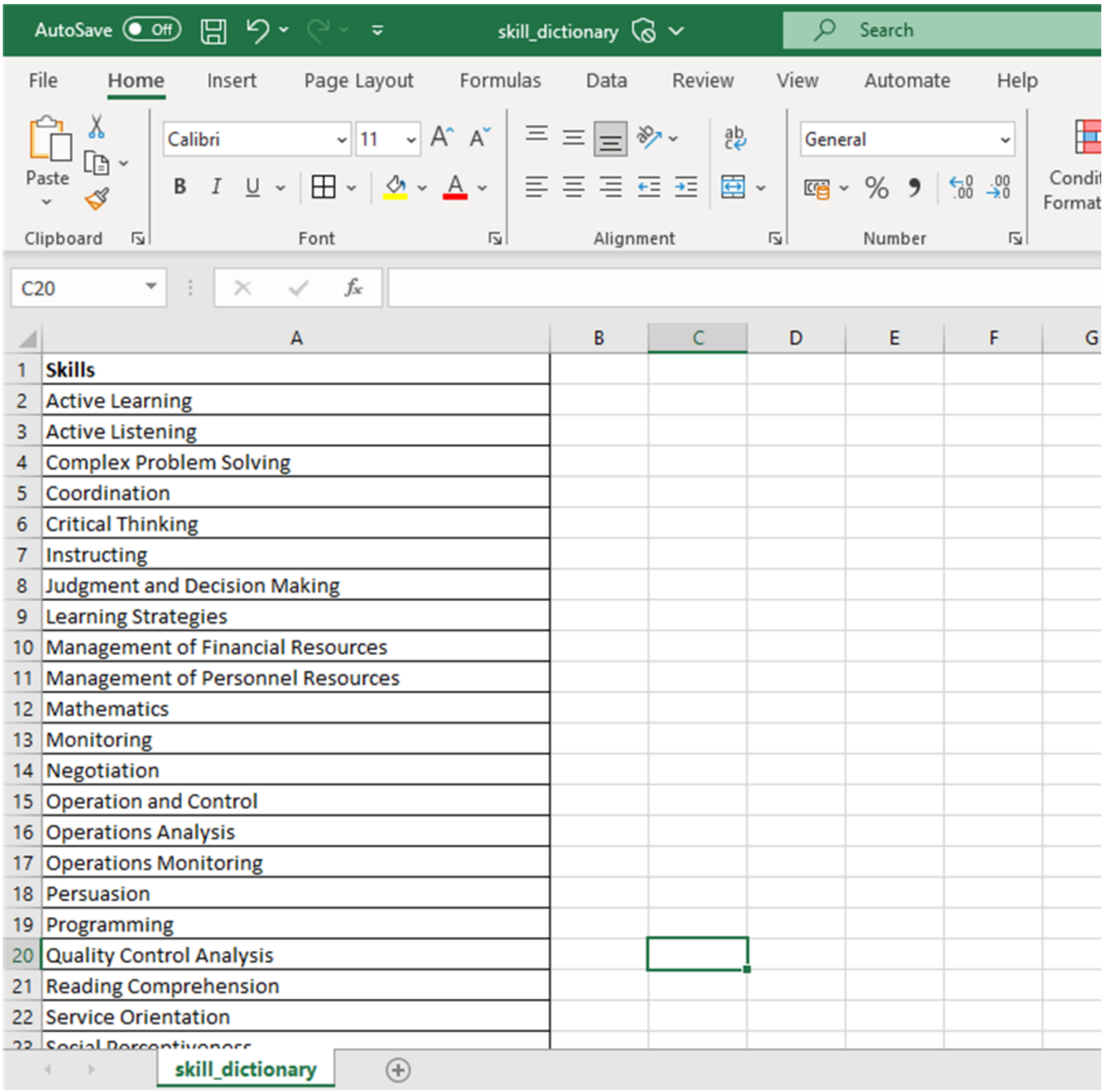
```python
        for job_button in job_buttons:

            print("Progress: {}".format("" + str(len(jobs)) + "/" + str(num_jobs)))
            if len(jobs) >= num_jobs:
                break
            print("here")

            try:
                job_button.click()
            except:
                 try:
                     driver.find_element(By.CSS_SELECTOR,'[alt="Close"]').click()
                 except NoSuchElementException:
                     pass

            time.sleep(3)

            try:
                title = driver.find_element(By.XPATH, "//*[@id='jobsearch-
ViewjobPaneWrapper']/div/div/div[1]/div/div[1]/div[1]/h2").text
                size = len(title)
                job_title = title[:size - 11]
                company_name = driver.find_element(By.XPATH,"//*[@id='jobsearch-
ViewjobPaneWrapper']/div/div/div[1]/div/div[1]/div[2]/div/div/div[1]/div[1]").text
                location = driver.find_element(By.XPATH,"//*[@id='jobsearch-
ViewjobPaneWrapper']/div/div/div[1]/div/div[1]/div[2]/div/div/div[2]").text
                job_description = driver.find_element(By.ID,'jobDescriptionText').get_attribute("innerText")
            except:
                time.sleep(5)
            print("Job Title: {}".format(job_title))
            print("Job Description: {}".format(job_description[:500]))
            print("Company Name: {}".format(company_name))
            print("Location: {}".format(location))


            jobs.append({"Job Title" : job_title,
            "Job Description" : job_description,
            "Company Name" : company_name,
            "Location" : location
            })

        print('should move to next page')
        try:
            driver.find_element(By.XPATH,'//*[@id="jobsearch-JapanPage"]/div/div/div[5]/div[1]/nav/div[6]/a').click()
            print('moved to next page')
        except NoSuchElementException:
            print("Scraping terminated before reaching target number of jobs. Needed {}, got {}.".format(num_jobs, len(jobs)))
            break

    return pd.DataFrame(jobs)

df = get_jobs(5000, False)

df.to_csv('scaper/indeed_banking_jobs.csv')
```

Appendix B – Python Coding for Data Cleaning



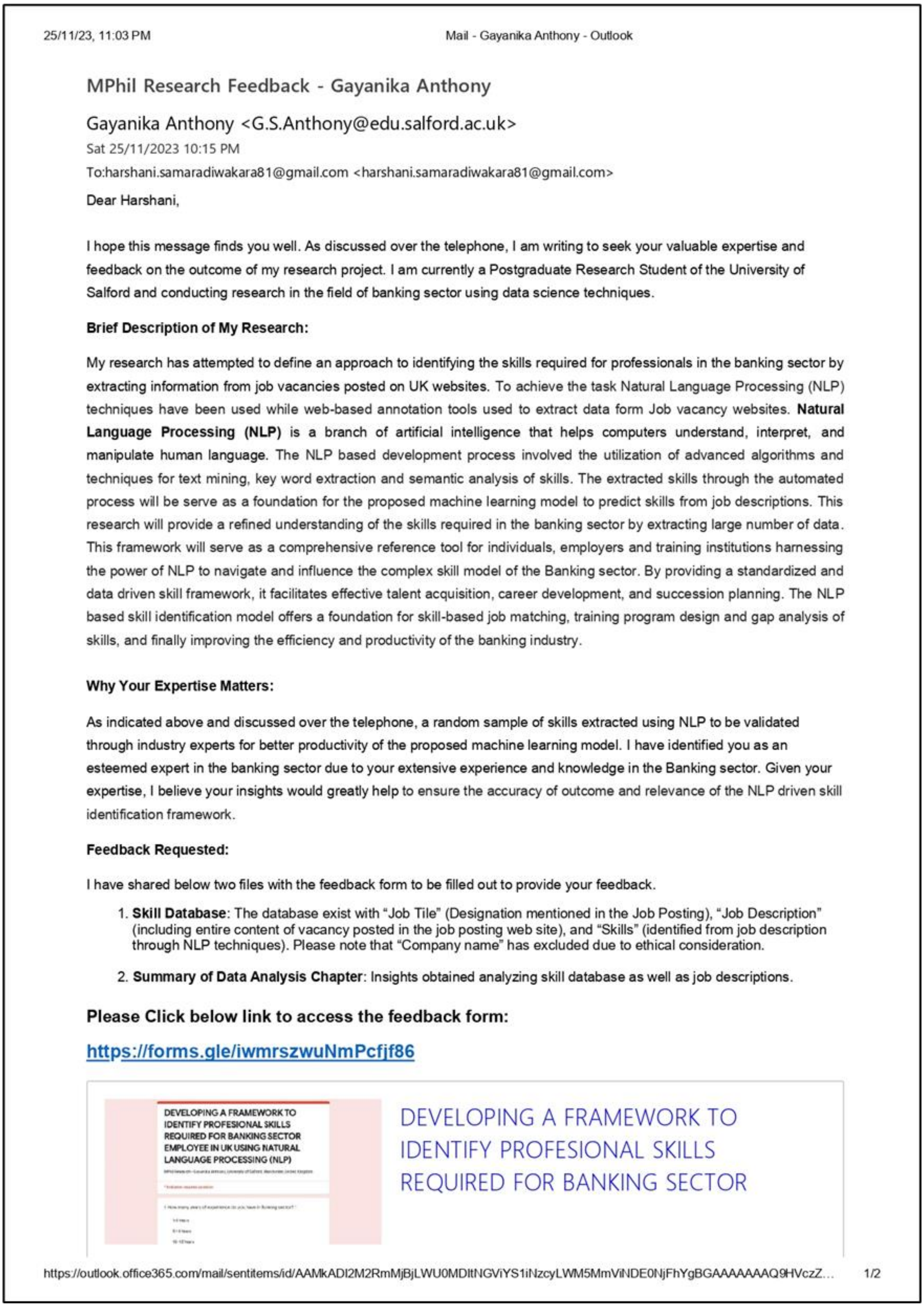
```python
import pandas as pd
import re

def split_province(x):
    sen = x.split(',')
    if len(sen) > 0:
        return sen[0]
    else:
        return x

def remove_rating(x):
    regex = re.compile('[-+]?[0-9]*\.[0-9]+$')
    if regex.search(x) is not None:
        return x[:-3]
    else:
        return x


df_glass = pd.read_csv('scraper/indeed_banking_jobs.csv')
#remove redundant columns
df_glass = df_glass[["Job Title","Company Name", "Job Description", "Location"]]

#remove duplicates, where Job Title and Company Name and Location are the same among rows
df_glass = df_glass.drop_duplicates(subset=['Job Title', 'Company Name', 'Location'], keep='first')

#use regex to remove rating from company name if rating exists
df_glass["Company Name"] = df_glass["Company Name"].apply(lambda x: remove_rating(x))

df_glass = df_glass.sample(frac = 1).reset_index(drop=True)

df_glass.to_csv('outputs/banking_jobs.csv')
```

Appendix C – Skill Dictionary



	A	B	C	D	E	F	G
1	**Skills**						
2	Active Learning						
3	Active Listening						
4	Complex Problem Solving						
5	Coordination						
6	Critical Thinking						
7	Instructing						
8	Judgment and Decision Making						
9	Learning Strategies						
10	Management of Financial Resources						
11	Management of Personnel Resources						
12	Mathematics						
13	Monitoring						
14	Negotiation						
15	Operation and Control						
16	Operations Analysis						
17	Operations Monitoring						
18	Persuasion						
19	Programming						
20	Quality Control Analysis						
21	Reading Comprehension						
22	Service Orientation						
23	Social Perceptiveness						

Appendix D – Email sent to Experts to obtain Feedback



25/11/23, 11:03 PM Mail - Gayanika Anthony - Outlook

MPhil Research Feedback - Gayanika Anthony

Gayanika Anthony <G.S.Anthony@edu.salford.ac.uk>
Sat 25/11/2023 10:15 PM
To:harshani.samaradiwakara81@gmail.com <harshani.samaradiwakara81@gmail.com>

Dear Harshani,

I hope this message finds you well. As discussed over the telephone, I am writing to seek your valuable expertise and feedback on the outcome of my research project. I am currently a Postgraduate Research Student of the University of Salford and conducting research in the field of banking sector using data science techniques.

Brief Description of My Research:

My research has attempted to define an approach to identifying the skills required for professionals in the banking sector by extracting information from job vacancies posted on UK websites. To achieve the task Natural Language Processing (NLP) techniques have been used while web-based annotation tools used to extract data form Job vacancy websites. **Natural Language Processing (NLP)** is a branch of artificial intelligence that helps computers understand, interpret, and manipulate human language. The NLP based development process involved the utilization of advanced algorithms and techniques for text mining, key word extraction and semantic analysis of skills. The extracted skills through the automated process will be serve as a foundation for the proposed machine learning model to predict skills from job descriptions. This research will provide a refined understanding of the skills required in the banking sector by extracting large number of data. This framework will serve as a comprehensive reference tool for individuals, employers and training institutions harnessing the power of NLP to navigate and influence the complex skill model of the Banking sector. By providing a standardized and data driven skill framework, it facilitates effective talent acquisition, career development, and succession planning. The NLP based skill identification model offers a foundation for skill-based job matching, training program design and gap analysis of skills, and finally improving the efficiency and productivity of the banking industry.

Why Your Expertise Matters:

As indicated above and discussed over the telephone, a random sample of skills extracted using NLP to be validated through industry experts for better productivity of the proposed machine learning model. I have identified you as an esteemed expert in the banking sector due to your extensive experience and knowledge in the Banking sector. Given your expertise, I believe your insights would greatly help to ensure the accuracy of outcome and relevance of the NLP driven skill identification framework.

Feedback Requested:

I have shared below two files with the feedback form to be filled out to provide your feedback.

1. **Skill Database**: The database exist with "Job Tile" (Designation mentioned in the Job Posting), "Job Description" (including entire content of vacancy posted in the job posting web site), and "Skills" (identified from job description through NLP techniques). Please note that "Company name" has excluded due to ethical consideration.
2. **Summary of Data Analysis Chapter**: Insights obtained analyzing skill database as well as job descriptions.

Please Click below link to access the feedback form:

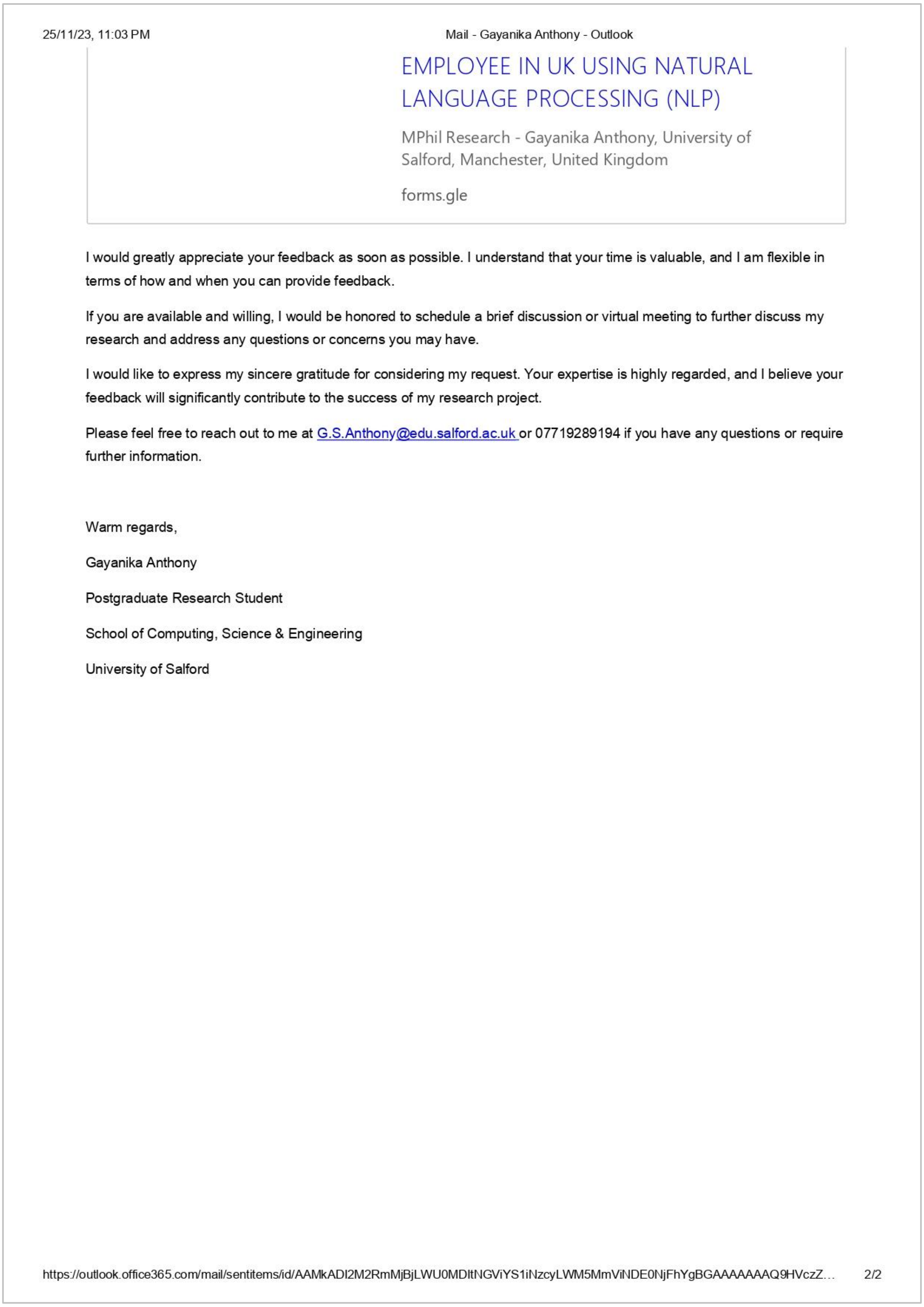
https://forms.gle/iwmrszwuNmPcfjf86

DEVELOPING A FRAMEWORK TO IDENTIFY PROFESIONAL SKILLS REQUIRED FOR BANKING SECTOR EMPLOYEE IN UK USING NATURAL LANGUAGE PROCESSING (NLP)

DEVELOPING A FRAMEWORK TO IDENTIFY PROFESIONAL SKILLS REQUIRED FOR BANKING SECTOR

https://outlook.office365.com/mail/sentitems/id/AAMkADI2M2RmMjBjLWU0MDItNGViYS1iNzcyLWM5MmViNDE0NjFhYgBGAAAAAAAQ9HVczZ… 1/2

Appendix D– Email sent to Experts to obtain Feedback (Continued)



25/11/23, 11:03 PM Mail - Gayanika Anthony - Outlook

EMPLOYEE IN UK USING NATURAL LANGUAGE PROCESSING (NLP)

MPhil Research - Gayanika Anthony, University of Salford, Manchester, United Kingdom

forms.gle

I would greatly appreciate your feedback as soon as possible. I understand that your time is valuable, and I am flexible in terms of how and when you can provide feedback.

If you are available and willing, I would be honored to schedule a brief discussion or virtual meeting to further discuss my research and address any questions or concerns you may have.

I would like to express my sincere gratitude for considering my request. Your expertise is highly regarded, and I believe your feedback will significantly contribute to the success of my research project.

Please feel free to reach out to me at G.S.Anthony@edu.salford.ac.uk or 07719289194 if you have any questions or require further information.

Warm regards,

Gayanika Anthony

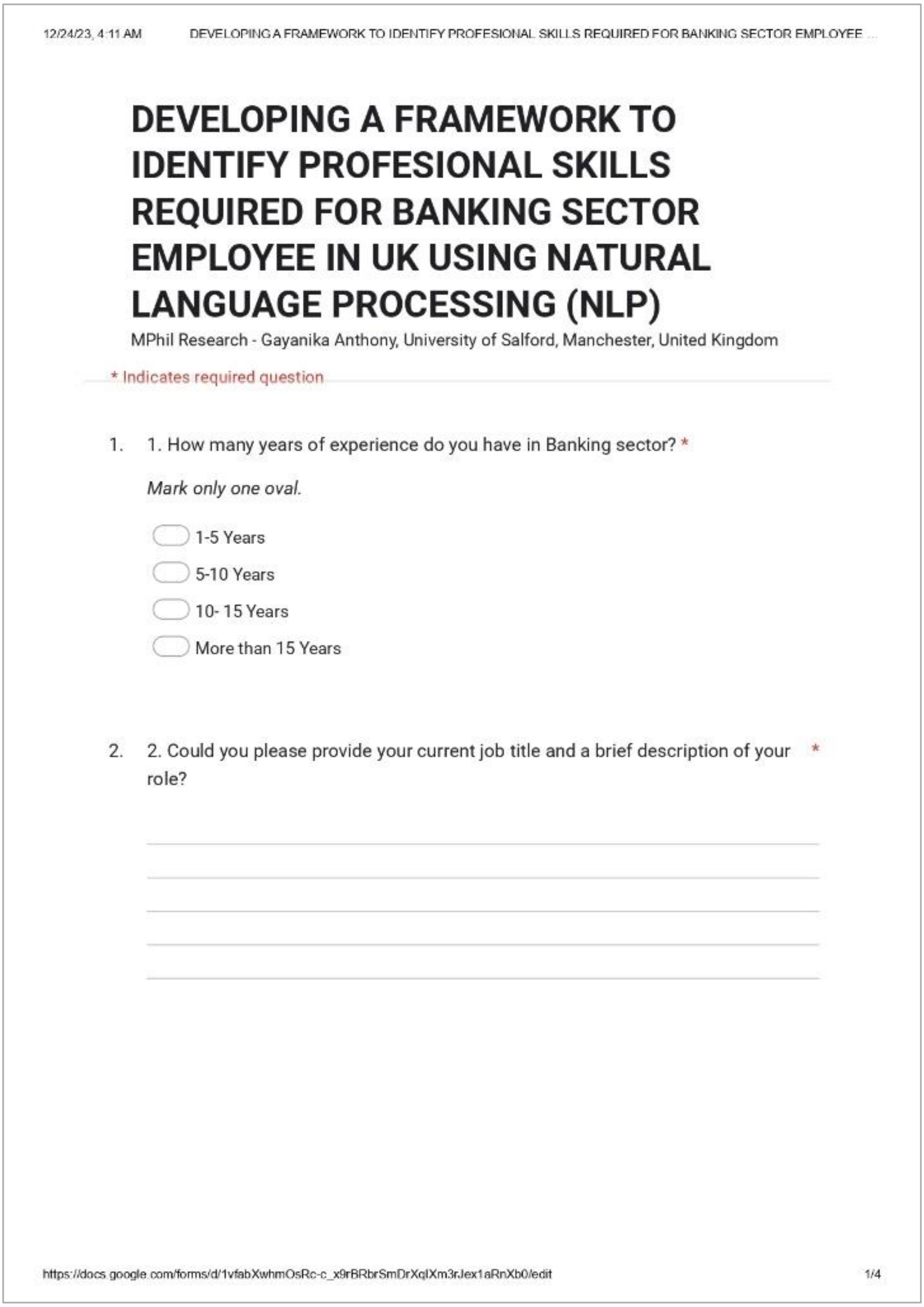
Postgraduate Research Student

School of Computing, Science & Engineering

University of Salford

https://outlook.office365.com/mail/sentitems/id/AAMkADI2M2RmMjBjLWU0MDItNGViYS1iNzcyLWM5MmViNDE0NjFhYgBGAAAAAAAQ9HVczZ… 2/2

Appendix E - Google Form Generated to obtain Expert Feedback



12/24/23, 4:11 AM DEVELOPING A FRAMEWORK TO IDENTIFY PROFESIONAL SKILLS REQUIRED FOR BANKING SECTOR EMPLOYEE ...

DEVELOPING A FRAMEWORK TO IDENTIFY PROFESIONAL SKILLS REQUIRED FOR BANKING SECTOR EMPLOYEE IN UK USING NATURAL LANGUAGE PROCESSING (NLP)

MPhil Research - Gayanika Anthony, University of Salford, Manchester, United Kingdom

* Indicates required question

1. 1. How many years of experience do you have in Banking sector? *

   Mark only one oval.

   ○ 1-5 Years

   ○ 5-10 Years

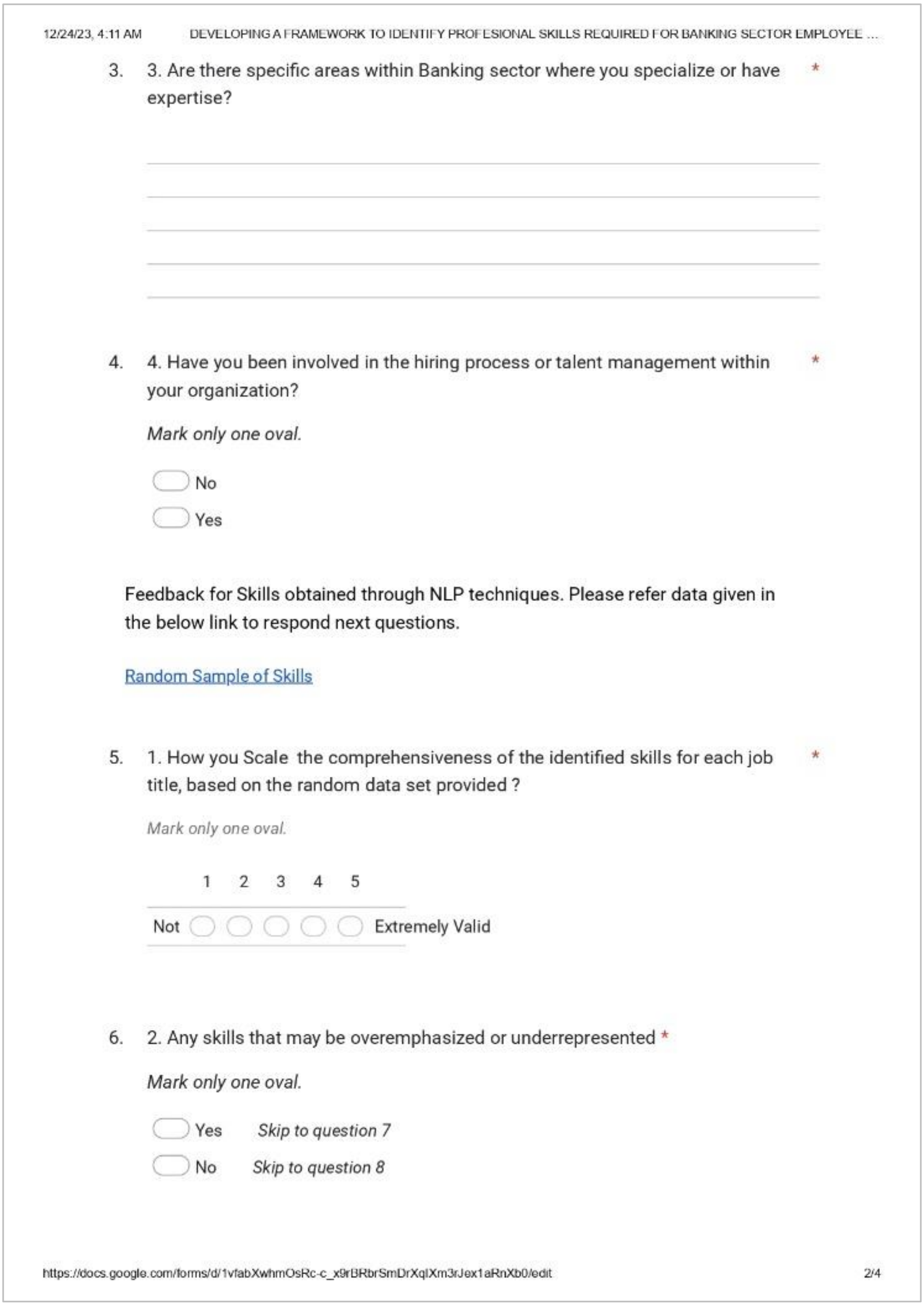
   ○ 10- 15 Years

   ○ More than 15 Years

2. 2. Could you please provide your current job title and a brief description of your role? *

https://docs.google.com/forms/d/1vfabXwhmOsRc-c_x9rBRbrSmDrXqlXm3rJex1aRnXb0/edit 1/4

Appendix E - Google Form Generated to obtain Expert Feedback (continued)



12/24/23, 4:11 AM DEVELOPING A FRAMEWORK TO IDENTIFY PROFESIONAL SKILLS REQUIRED FOR BANKING SECTOR EMPLOYEE ...

3. 3. Are there specific areas within Banking sector where you specialize or have expertise? *

____

____

____

____

____

4. 4. Have you been involved in the hiring process or talent management within your organization? *

Mark only one oval.

○ No

○ Yes

Feedback for Skills obtained through NLP techniques. Please refer data given in the below link to respond next questions.

Random Sample of Skills

5. 1. How you Scale the comprehensiveness of the identified skills for each job title, based on the random data set provided ? *

Mark only one oval.

	1	2	3	4	5	
Not	○	○	○	○	○	Extremely Valid

6. 2. Any skills that may be overemphasized or underrepresented *

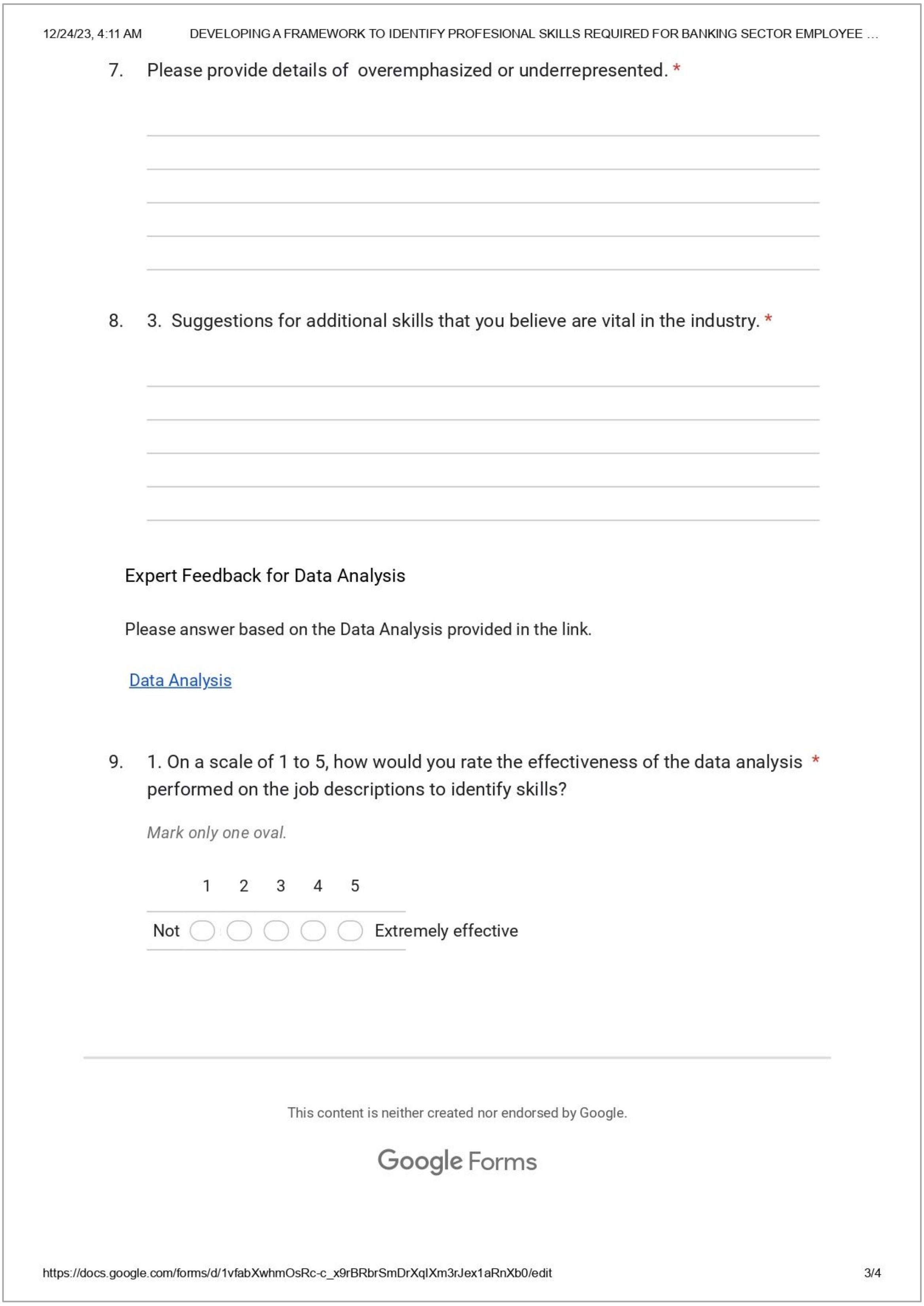
Mark only one oval.

○ Yes *Skip to question 7*

○ No *Skip to question 8*

https://docs.google.com/forms/d/1vfabXwhmOsRc-c_x9rBRbrSmDrXqlXm3rJex1aRnXb0/edit 2/4

Appendix E - Google Form Generated to obtain Expert Feedback (continued)



12/24/23, 4:11 AM DEVELOPING A FRAMEWORK TO IDENTIFY PROFESIONAL SKILLS REQUIRED FOR BANKING SECTOR EMPLOYEE ...

7. Please provide details of overemphasized or underrepresented. *

8. 3. Suggestions for additional skills that you believe are vital in the industry. *

Expert Feedback for Data Analysis

Please answer based on the Data Analysis provided in the link.

Data Analysis

9. 1. On a scale of 1 to 5, how would you rate the effectiveness of the data analysis performed on the job descriptions to identify skills? *

Mark only one oval.

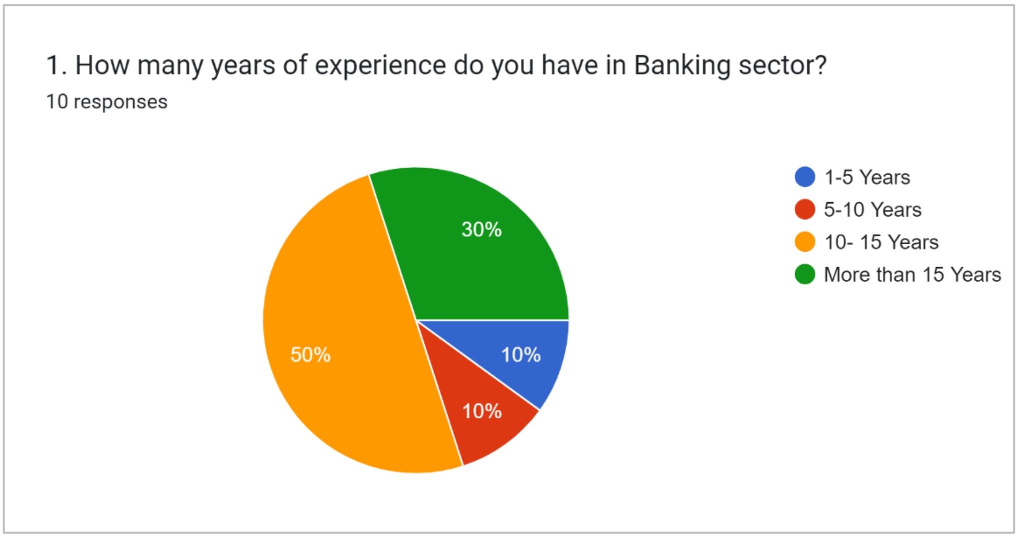
	1	2	3	4	5	
Not	○	○	○	○	○	Extremely effective

This content is neither created nor endorsed by Google.

Google Forms

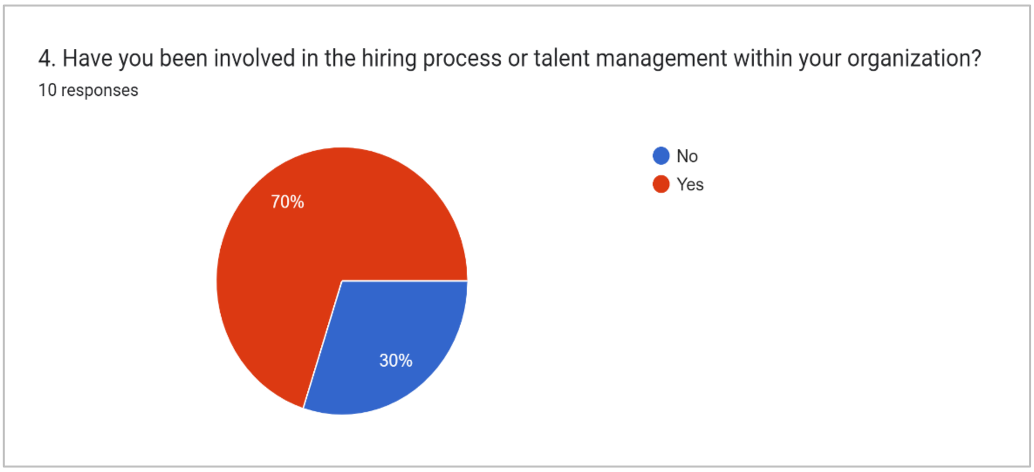
https://docs.google.com/forms/d/1vfabXwhmOsRc-c_x9rBRbrSmDrXqlXm3rJex1aRnXb0/edit 3/4

Appendix F - Executive Feedback – Profile Information



1. How many years of experience do you have in Banking sector?

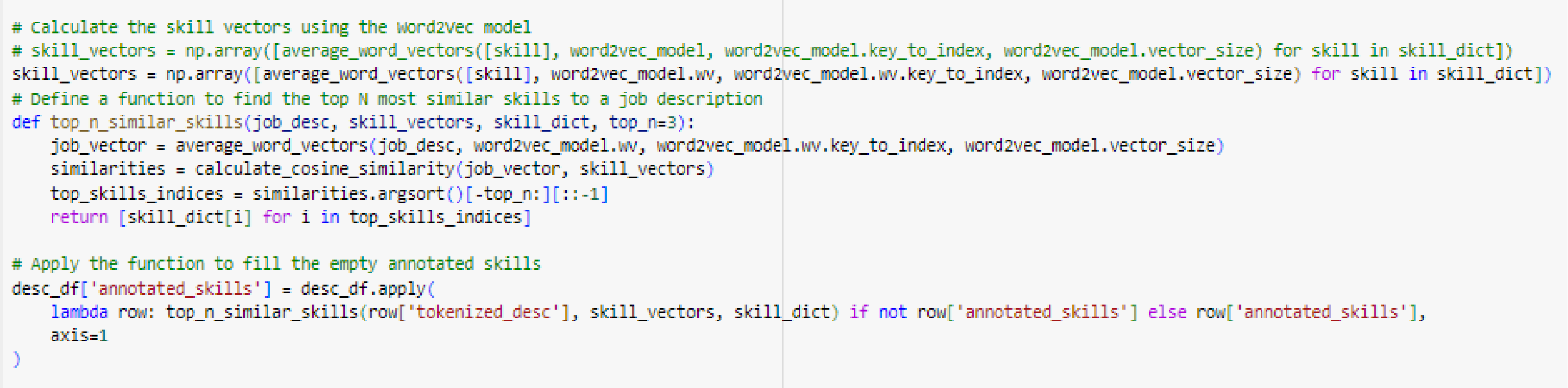
10 responses



4. Have you been involved in the hiring process or talent management within your organization?

10 responses

Appendix G – Python Code for Function Cosine Similarity



```python
# Calculate the skill vectors using the Word2Vec model
# skill_vectors = np.array([average_word_vectors([skill], word2vec_model, word2vec_model.key_to_index, word2vec_model.vector_size) for skill in skill_dict])
skill_vectors = np.array([average_word_vectors([skill], word2vec_model.wv, word2vec_model.wv.key_to_index, word2vec_model.vector_size) for skill in skill_dict])
# Define a function to find the top N most similar skills to a job description
def top_n_similar_skills(job_desc, skill_vectors, skill_dict, top_n=3):
    job_vector = average_word_vectors(job_desc, word2vec_model.wv, word2vec_model.wv.key_to_index, word2vec_model.vector_size)
    similarities = calculate_cosine_similarity(job_vector, skill_vectors)
    top_skills_indices = similarities.argsort()[-top_n:][::-1]
    return [skill_dict[i] for i in top_skills_indices]

# Apply the function to fill the empty annotated skills
desc_df['annotated_skills'] = desc_df.apply(
    lambda row: top_n_similar_skills(row['tokenized_desc'], skill_vectors, skill_dict) if not row['annotated_skills'] else row['annotated_skills'],
    axis=1
)
```