A COMPARATIVE EVALUATION OF

TEXT REPRESENTATION TECHNIQUES

FOR CONTENT-BASED JOB RECOMMENDATION SYSTEM

TEMUULEN Bulgan

A Thesis Submitted in Partial Fulfilment

of the requirements for the

Degree of

Master of Science in Data Analytics

Diagram

Description automatically generated with medium confidence

February 2024

Supervisor: Kislay Raj

# TABLE OF CONTENTS

**ABSTRACT** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - **i**

**TABLE OF CONTENTS** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - **iii**

**TABLE OF FIGURES** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - **iv**

**LIST OF TABLES** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - **v**

**LIST OF ABBREVIATIONS** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - **vi**

**Chapter 1. Introduction** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

1. Introduction - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
2. Research Question and Research Objectives - - - - - - - - - - - - - - - - - - -
3. Thesis Organization - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

**Chapter 2. Background Study and Related Works - - - - - - - - - - - - - - - - - - - - - - -**

1. Background Study
   * 1. Information Overload - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     2. Recommendation as a Problem - - - - - - - - - - - - - - - - - - - - - -
     3. Job Recommendation System - - - - - - - - - - - - - - - - - - - - - - -

Content-Based Recommendations System - - - - - - - - - - - - - -

* + 1. Text Representation - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Bag of Words - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

Term Frequency - Inverse Document Frequency - - - - - - - - - - -

Word Embedding - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - Bidirectional Encoder Representations from Transformers - - - -

* + 1. Similarity Function - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    2. Evaluation - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    3. Model Evaluation - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    4. Statistic Test - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
    5. Web Scraping - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

1. Related Works - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

2.2.1 TF-IDF in Comparative Experiments for RS

2.2.2 Word2Vec in Comparative Experiments for RS

2.2.3 Bert in Comparative Experiments for Embedding

1. Research Gaps **-**

**Chapter 3. Methodology** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

1. Sampling Strategy - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   * 1. Job Seekers’ Dataset - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     2. Job Offers’ Dataset - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
2. Experimental setup - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
3. Implementation of Experiment 1 - - - - - - - - - - - - - - - - - - - - - - - - -
   1. Experiment Objective - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   2. Model Design - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   3. Data preparation - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   4. Train-test split - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   5. Procedure - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   6. Evaluation metrics - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
   7. Benchmarking - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
4. Implementation of Experiment 2 - - - - - - - - - - - - - - - - - - - - - - - - - - - -

3.3.1 Experiment Objective - - - - - - - - - - - - - - - - - - - - - - - - - - - -

3.3.2 Model Design - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

1. Implementation of Experiment 3 - - - - - - - - - - - - - - - - - - - - - - - -
2. Ethical Considerations - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

**Chapter 4. Evaluation of Experiment Results** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

1. Experiment 1 Results

4.1.1 Evaluation of BERT Encoding

4.1.2 Evaluation of Word2Wec Embedding

4.1.3 Evaluation of TF-IDF Embedding

4.1.4 Experiment Conclusion

1. Job Ranking Results
2. Human Evaluation results

**Chapter 5. Conclusion** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -- - - - - - -

5.1 Conclusion - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

**References** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

**Appendix** - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

# List of Abbreviations

**ANN**  Artificial Neural Networks

**BERT** Bidirectional Encoder Representations from Transformers

**BoW** Bag of Words

**CBF** Content Based Filtering

**CF** Collaborative Filtering

**CV** Curriculum Vitae

**DL** Deep Learning

**DNN** Deep Neural Networks

**FNN** Feedforward Neural Networks

**GloVe** Global Vectors for Word Embeddings

**HF** Hybrid Filtering

**IDF** Inverse Document Frequency

**IF** Information Filtering

**IR** Information Retrieval

**JRS**  Job recommendations System

**NLP** Natural Language Processing

**NN** Neural Network

**PNN** Probabilistic Neural Networks

**RBO** Rank Biased Overlap

**TF** Term Frequency

**TF-IDF** Term Frequency Inverse Document Frequency

**RS** Recommendation System

**WMD** Word Mover's Distance

# Chapter 1. Introduction

## Introduction

Matching human resources effectively with specific job roles is of great significance to the sustainable and healthy development of human resources management and contemporary enterprises (Ni, 2022). However, the rapid growth of online job portals and the adoption of e-recruitment in the current global panorama, along with the massive digital data they contain, has directed to a notable issue of information overload, affecting both individuals seeking employment opportunities and recruiters who are presented with an abundance of options and choices, thereby making the decision-making process more complex. In addition, the ongoing advancement of technology and shifts in industries are continually generating novel employment prospects that demand a wide range of skills, sometimes leading to a realignment of individuals' career paths and contributing to a misalignment between job seekers and the available job market. In response to these challenges, there have been numerous developments in state-of-the-art technology and systems in the field of online talent management. One notable achievement in this domain is the Job Recommendation System (JRS), which falls under the broader category of Recommendation Systems (RS)—algorithms engineered to offer personalized suggestions to users across diverse content or items (Isinkaye et al., 2015). JRS was introduced as a bidirectional technology-driven solution for job recruiting (Yu et al., 2011), designed to match or rank candidates with relevant job openings via online job portals while considering their profiles, employment histories, and behavioural patterns. As Farber et al. recognized in 2003, JRS represented a tremendous leap forward in this capacity when compared to the traditional and limited Boolean search strategy commonly used in talent management.

Over the last decade, JRS has been studied from various perspectives, and it continues to be one of the hot topics in RS that requires specific attention (Mashayekhi et al., 2022). A large number of research publications have delved into a wide spectrum of algorithms, all with the ultimate goal of developing the most effective structure in the JRS domain. This endeavour has resulted in the development of various technical approaches, each designed to fulfil a diverse need across a range of scenarios and continuously adapting to technological advancements. Even though the classification and ideas are changing swiftly, the foundational principles of most JRS, including the entire RS, have remained relatively consistent, predominantly centred around Collaborative Filtering (CF) and Content-Based Filtering (CBF) recommendations (Guo et al., 2017). CF, in particular, has achieved notable success in various domains of traditional RS in terms of accuracy and has demonstrated its effectiveness in handling different situations where both users (also later referred to as 'job seekers' in JRS) and items (also later referred to as 'jobs' in JRS) undergo daily updates. However, this method comes with inherent limitations in JRS, with CF struggling to address cold-start problems where job seekers have yet to encounter job ads on many occasions, making collaborative methods inapplicable (Su and Khoshgoftaar, 2009).

On the other hand, CBF has proven to be a valuable solution for addressing this challenge, mainly when user and item features and profiles are known and its fundamental concept centres on recommending items with shared contextual similarities (Tran et al., 2017). A noteworthy real-world application of these techniques in JRS can be observed from the 2017 ACM RecSys Conference Recommender Systems Challenge (Abel et al., 2016). This challenge was specifically focused on the task of the job recommendations, with the goal of identifying users interested in job advertisements and matching them with suitable job openings. The challenge attracted a total of 262 participating teams from all over the world, who collectively submitted over 6,100 solutions. The winning system for this challenge was proposed by a team of researchers who advocated for the Content-Based Neighbor Model (Volkovs et al., 2017) as the most effective approach for addressing the challenge in JRS.

While CBF offers its advantages in the field of JRS, it is not without its drawbacks. One major limitation is the hand-engineering of features for job and job seeker profiles, which demands substantial domain knowledge. Consequently, the system's performance depends on the quality of these hand-crafted features (Google, 2023). The second major challenge is the absence of a definitive method for constructing the optimal person-job matching architecture. This challenge is closely tied to the domain's nature, which essentially presents a text mining problem, alternatively addressable as a Natural Language Processing (NLP) challenge. Currently, researchers and enterprises have proposed a plethora of different NLP techniques for CBF, primarily focusing on text representations approaches. This encompasses not only traditional statistical techniques such as Term Frequency - Inverse Document Frequency (TF-IDF) (Bansal et al., 2017), Bag-of-Words (BoW) (Guo et al., 2014), and well-established embedding methods like GloVe (Brahushi and Ahmad, 2022), Word2Vec (Gugnani and Misra, 2020), ELMo (Kurdija et al., 2020), but also transformer-based models like BERT (Panchasara et al., 2023), XLNET (Nguyen et al., 2022), along with large language models (LLM) such as LLaMa and ChatGPT (Salinas et al., 2023), implying their suitability for person-job matching tasks. Unfortunately, the existing body of literature in the field of JRS has inadequately addressed this matter, as there is a notable absence of a comprehensive comparative analysis of the most frequently employed approaches for matching in the content-based e-recruitment domain.

Consequently, a pivotal research questions arise: “Which NLP’s text representation architecture for person-job ranking in the content-based job RS domain demonstrates the highest level of effectiveness?”, “Can novel approaches outperform intermediary and conventional approaches?”. To answer these questions and bridge this research gap, my study aims to pursue a review of academic papers and conduct a comprehensive comparative evaluation experiment, with a particular focus on assessing the performance of the most commonly used NLP techniques for person-job matching or ranking based systems. Through this research undertaking, I aim to make a minor contribution to the realm of JRS by enhancing the current state of research concerning automating job markets and career opportunities in the digital era.

## 1.2 Research Question and Research Objectives

The research aims to address the following *research question*:

***Can the application of Transformer’s encoders, known for their state-of-the-art performance across various NLP tasks, outperform distributed word embeddings or traditional statistical representation methods in improving the accuracy of person-job ranking within job recommendation systems?***

*Research Hypothesis:*

To address the research question raised by this study, it is necessary to formulate it as a set of hypotheses. Depending on the outcomes of the experiment, these hypotheses will either be rejected or accepted. Following the execution of statistical tests to assess differences between the models, if the difference is statistically significant (p<0.05), we reject the null hypothesis and accept the alternate hypothesis.

*H0 (null hypothesis):* A job recommendation system utilising Transformer's encoders for text representation in its person-job ranking algorithm will not show significantly improved accuracy outcomes compared to systems that rely on distributed word embeddings or traditional statistical representations.

*H1 (alternate hypothesis):* A job recommendation system utilising Transformer’s encoders for text representation in its person-job ranking algorithm will show significantly improved accuracy outcomes compared to systems that rely on distributed word embeddings or traditional statistical representations*.*

In addition to the primary hypothesis, this research integrates several supportive objectives. These additional objectives are purposefully crafted to enhance the study, offering greater depth of understanding and supplementary evidence:

* Objective 1: *To conduct a comparative model evaluation experiment among different text representation architectures.* The first objective of the study focuses on conducting a thorough comparative analysis through an experiment. This involves using both a validated benchmark dataset from secondary sources and primary collected data. The experiment will deploy the designed architectures to assess their effectiveness in ranking individuals for jobs. The objective is to identify the strengths and weaknesses of the employed methods and how well they perform in real-world situations.
* Objective 2: *To conduct a comparative human evaluation experiment among different text representation architectures in real-life situations.* The secondary objective of the research entails conducting a detailed comparative experiment involving human evaluation. This experiment utilises primary collected data as well as data obtained through web scraping. It involves implementing the created architectures to test their effectiveness in ranking and matching people with jobs under real-world conditions. The objective also includes determining the strengths and drawbacks of the techniques used, as well as evaluating their practical performance in real-life scenarios.
* Objective 3: *To the best of my knowledge draw conclusions and formulate evidence-based recommendations.* The third and final concluding objective of the study is focused on integrating the results of the research. The primary aim here is to draw informed conclusions from the results of the comparative evaluation studies. These conclusions are intended to lay the groundwork for recommendations based on evidence, offering key insights to researchers, practitioners, and decision-makers working within the dynamic landscape of JRS. The overarching purpose of this research is to enrich the field by providing data-driven guidance, enhancing the precision and efficiency of person-job ranking, and contributing to the evolving domain of JRS.

## 1.3 Thesis Organization

The flow of the thesis goes in following manner:

*Chapter 1:* Consists of introduction and research question with its objectives.

*Chapter 2*: Describes the background study for the research and the summary of related works by different authors.

*Chapter 3:* Includes the overview of the methodology will detail the experiment proposed address the research questions. It will define the methods, libraries and tools that is used to test the hypothesis as well as the metrics required to evaluate the experiment results and ultimately answer the research questions.

*Chapter 4:* Provides a comprehensive description of the implementation processes undertaken to execute the experiments, along with a presentation of various experimental results and evaluation findings.

*Chapter 5:* Conclusion.

# Chapter 2. Background Study and Related Works

## 2. 1 Information Overload:

According to Wu et al. (2022), the rapid expansion of digital technology and rising internet usage have led to a substantial information overload issue, where customers face an excessive number of options and decisions. Finding relevant and distinctive items and content among the abundance of similar online possibilities has become a pervasive challenge (Roy and Dutta, 2022). As a result, it has necessitated the expansion of one of today's powerful digital tools known as RS. RS are advanced technologies that aid in determining the ranking or user preference for specific items. They have been developed to address the challenge of information overload, sifting through vast and constantly changing data (Konstan and Riedl, 2012), based on the user's preferences, interests, or observed behaviour regarding the product (Karypis, 2001; Pan and Li, 2010). In the present day, RS have evolved into a distinct field of research and stand as one of the most successful web applications, catering to billions of users daily by providing recommendations for diverse content such as news feeds, videos, e-commerce products, music, movies, books, games, friends, jobs, and more (Dong et al., 2022). Their extensive use spans various industries and platforms, including e-commerce websites, streaming services, social media platforms, and content platforms. Prominent e-commerce websites like Amazon (Ha et al., 2017), Netflix (Gomez-Uribe et al., 2016), YouTube (Davidson et al., 2010), Spotify (Millecamp et al., 2018), Facebook (Shapira et al., 2013), LinkedIn (Kenthapadi et al., 2017), Airbnb (Mauro et al. 2021), and Tinder (Bartlett et al., 2023) all leverage recommendation engines to enhance their user experiences. In recent years, there has been an upsurge of different methodologies and approaches for building RS, each varying in the problem domain, the type of information used, and the recommendation algorithm employed for making predictions with the most commonly used methods, such as CF, CBF, and Hybrid Filtering (HF)

## 2.2 Recommendation as a Problem

According to Celma (2010), the issue of making recommendations is divided into two distinct problems: predicting and recommending. The prediction part deals with guessing how likely it is that a user will favor certain items, while the recommendation part concerns with suggesting a list of N items to the user, which also can be reduced as listing of top N items once the system is capable of ranking items in a complete order.

Sarwar and his colleagues in 2001 described the prediction challenge by identifying a group of *m* users and a set of *n* items that could be recommended. Each user has a list of items they have provided feedback on, either directly or indirectly. This list of items for a user is a subset of the total items and could even be empty. They also introduced a function to estimate how likely it is that an active user would prefer an item not already familiar to them.

Furthermore, Sarwar and his team in 2001 explained the recommendation problem as the process of creating a list of N items, chosen from the total set of items, that the user is most likely to enjoy. Specifically, these are the N items ranked highest in terms of the user's estimated preference. This list is carefully ordered and excludes items the user has already shown interest in. Considering the vast number of potential items and users, this process can be quite complex.

## 2.3 Job Recommendation System:

Long before the internet, Vega (Vega, 1990) proposed a novel approach that would connect job seekers with open employment. This system was available via the Minitel terminal, a precursor to the internet, by dialling 3615 and browsing to the LM/EMPLOI service. It allowed users to enter search queries and submit digital resumes via text messaging over the phone line. The system would then match the input against a database using a predefined job taxonomy, identifying a list of job vacancies that could potentially interest the applicant. But it was a complex task that was urging for innovation. Then in the early stages of information systems, the focus of human resource management was largely on the storage and tracking of applicant data through applicant management systems. These systems facilitated internal workflows and communications between the HR department and other departments, as noted by Al-Otaibi and Ykhlef (2012). Initially, elementary methods such as posting job advertisements on the career pages of corporate websites were employed. However, as experience with these initial systems grew, the realization of their potential led to the development of more sophisticated e-recruitment platforms, marking an evolution from simple job posting to more complex job matching solutions. In recent times, job recommendation has gained significant attention and importance on online recruiting platforms (Mashayekhi et al., 2022). Unlike traditional RS that suggest items to users, JRS focus on recommending job applicants to recruiters where the contexts of user and item are likely to be symmetrical. These systems aim to deliver personalized lists of job descriptions to applicants based on their preferences or present recruiters with a curated list of potential candidates who match the job requirements. In order to enhance the quality of recommendations, various recommendation approaches have been devised and incorporated into JRSs.

### 2.3.1 Content-based job recommendation system:

The core idea behind a content based JRS is to suggest jobs or positions to users by comparing the content of job listings to the user's profile, focusing on how similar they are. The definition of ‘content’ varies widely, depending on the specific field the system is applied to, resulting in a variety of variables that can be utilized for this purpose (Gao et al., 2012). Within JRSs, the content-based approach mainly depends on measuring the semantic similarity between the user's profile and job descriptions to generate recommendations. This similarity estimation helps in determining the relevance of each job opening to the job seeker. The fundamental process involves gathering content details from both job seekers and job descriptions, followed by evaluating their similarities. This typically involves selecting relevant features to use and converting this information into formats understandable by computational devices for subsequent analysis. Various methods have been employed for this conversion, including the Bag-of-Words (BoW) model with TF-IDF weighting (Mpela and Zuva, 2020), Latent Dirichlet Allocation (Bansal, Srivastava & Arora, 2017), along with more modern approaches like Word2vec (Gugnani and Misra, 2020; Vaverde-Rebaza et al., 2018; Janusz et al., 2018) and BERT (Vanetik, 2023). Despite the emergence of new techniques, content-based recommendation's contributions have remained notably consistent over the past decade, with ongoing recommendations from researchers for its use in JRSs. The key steps in CB methods typically involve:

1. *[u1, u2 … um] ∈ U, [ j1, j2 … jm] ∈ J*
2. Begin
3. Modelling *jobs j ∈ J, Content(j)*
4. Modelling user’s preference *u ∈ U, UserProfile(u)*
5. Estimate utility value of job *j* for user *u*.

*fcb(u, j) = Sim(user profile(u), content(j))*

1. Return *Top-N* of jobs based on value of *f(u,j)*
2. End.

*Where: Input:*

* *U = {u},* set of users
* *J = {j},* set of all available jobs

*Output:*

* *[u1, u2 … um] ∈ U,* return a ranked list (*Jto pn*) of potential jobs to recommend for *u*.

## 2.4 Text representation:

Human language is characterized by its intricate complexity and variability, featuring a broad spectrum of words, grammatical constructions, phrases, and subtle linguistic distinctions. For computers to interpret and handle these complexities effectively, text needs to be converted into a structured, numerical form that is accessible to algorithms. This conversion is achieved through text representation, a key element of NLP, enabling the transformation of textual information into a machine-readable format (Szymański, 2014). For a given set of text documents *D = {di, i=1, 2..., n},* where each di stands for a document, the problem of text representation is to transform each *di* of *D* as a point *si* in a numerical space *S*, where the distance or similarity between each pair of points in space *S* is well defined. Various methods exist for representing text features, each offering unique benefits and limitations.

### 2.4.1 Bag of Words (Bow):

BoW is the simplest form of statistical representation of text and one of the earliest (Baeza-Yates and Ribeiro-Neto, 1999). This approach views each document as a bag containing words, ignoring the sequence and grammatical structure of the text. It constructs a fixed-length vector, with each dimension representing a distinct word from the overall text collection. The value in each dimension reflects the occurrence frequency of the corresponding word within the document. While BoW is straightforward and effective for basic tasks, its major drawback is the lack of contextual understanding; it cannot grasp the semantics of words or phrases. BoW simplifies the transformation of unstructured text into a structured, numerical format suitable for machine learning algorithms. In the BOW model, a text document is represented as a collection of unordered terms. Given the document collection <*D = {di, i=1, 2..., n}>,* suppose there are m unique terms appeared in this collection. Mathematically, this corpus of documents can be represented by a *m* by *n* matrix <*S ∈ Rm\*n>*. Each text document is denoted by a column vector <*si, i = 1, 2 …, n>* and each term is denoted by a row vector. The *jth* entry of *si*is denoted by <*sji , j = 1, 2, …, m> (Liu and Özsu, 2009).*

Because of its simplicity and speed, BoW is often used as a preprocessing stage in more advanced statistical methods. However, the major drawback of it is that it only retains the frequency of the words in the document and loses the sequence information, disregarding grammar, and word order.

### 2.4.2 Term Frequency - Inverse Document Frequency

BoW can be extended by using weighting schemes that relate documents with words. One of the most popular extensions is TF-IDF. It enhances its predecessor model by considering the importance of words in a document relative to their frequency across the entire corpus (Aizawa, 2003). It operates on the principle of assigning weights to words based on their occurrence rate within a specific document (term frequency) and their rarity or commonality across the entire corpus (inverse document frequency). To determine the weight of any given word, TF-IDF calculates both the term frequency (TF) and inverse document frequency (IDF) and then multiplies these two values, as demonstrated in the following formula:

*TF (t, d) =*

*IDF(t) = log*

*TF-IDF (t, d) = TF (t, d)\*IDF (t)*

Where:

* TF refers to the ratio of the count of a specific word in a document to the total number of words in that document.
* IDF refers logarithm of the quotient of the total number of documents in the corpus by the number of documents containing the term T.

While TF-IDF reduces the influence of common words, thereby enhancing the contribution of more significant terms to a document's representation, the method remains straightforward and scalable, which is advantageous for analyzing large volumes of text. However, it shares a limitation with the BoW model in that it does not capture the positional context of words within the document (Ramos, 2003). Additionally, TF-IDF's effectiveness is highly dependent on the specific corpus it's applied to. For instance, a matrix representation derived from football data would not be applicable to tennis or basketball data. This underscores the importance of using high-quality training data for the model to be effective (Zhang et al., 2011).

Word embedding:

## 2.2 Related Works

Recognising the broad and varied scope of this domain, the review of related works is systematically divided into three distinct subsections. Each subsection is dedicated to a specific category of embedding methods in the RS: the classic TF-IDF, the more advanced Word2Vec, and the latest transformer encoders, BERT. This structure allows for a comprehensive exploration of the embedding landscape, from the foundational techniques to the latest advanced techniques. The decision to cover a broad spectrum of RSs arises from the observation that comparative experiments specifically tailored to the job recommendation domain are scarce.

### 2.2.1 TF-IDF in Comparative Experiments for Recommendation Systems

Romadon et al. (2020) reported that combining TF-IDF with ANN for RS classification problems works better than Word2Vec embeddings because TF-IDF has a higher average accuracy rate of 80.55% compared to 71.22% for Word2Vec. The study suggests that TF-IDF's enhanced capability in extracting relevant features for the task of classifying job applicant texts contributes to its better performance. It is proposed that augmenting the dataset and concentrating on high-accuracy criteria could further enhance results. Similarly, in 2021, Zhu developed a system for recommending scholarly articles from PubMed that are relevant to public datasets from the Gene Expression Omnibus (GEO). This system utilises and evaluates various methods for text representation. The results showed that methods based on term frequency, like BM25 and TF-IDF, worked better than all the others. This included well-known NLP embedding models like Doc2Vec, ELMo, and BERT.

The utilization of TF-IDF for categorizing CVs in relation to job openings was highlighted as a significant advancement. This method proved instrumental in assisting employers to find suitable candidates whose skills and experiences aligned with the job criteria, specifically within targeted geographic locations. According to Apaza et al. (2021), the success of TF-IDF in parsing and analyzing CV data underscored its importance as a powerful tool in the development of RS for job matchmaking.

Brahushi and Ahmad (2022) conducted a study to evaluate the efficacy of a hybrid two-way RS compared to the rankings of resumes and job descriptions by human experts. The research devised four distinct scenarios—matching resumes to resumes, jobs to jobs, resumes to jobs, and jobs to resumes—using a dataset of 400 documents to create a standard based on human rankings derived from content similarity. GloVe (Global Vectors for Word Embeddings) and TF-IDF methods were used to calculate cosine similarity scores across all scenarios and see how well the system matched these human-generated rankings. The comparison of system-generated rankings to human rankings utilised the Rank Biased Overlap (RBO) similarity score as the evaluation metric. The results showed that both GloVe and TF-IDF had median RBO scores above 0.5, but TF-IDF generally did better than GloVe, especially when comparing resumes to resumes, where the difference was very clear. This indicates that although both embedding approaches are capable of reflecting human judgement in the ranking of resumes and job descriptions, TF-IDF demonstrates a slightly superior congruence with human rankings in the majority of the scenarios.

### 2.2.2 Word2Vec in Comparative Experiments for Recommendation Systems

The comparison between TF-IDF and Word2Vec vectors in the context of job postings was done by Elsafty et al. (2018). The study revealed that Word2Vec models significantly outperform the TF-IDF baseline. The precision at the P@10 score for TF-IDF was only 8.69%, while Word2Vec models achieved scores ranging from 54.84% to 56.22%. Further enhancements using Word2VecF and Doc2VecC, particularly when combined with TF-IDF weights, led to even higher precision scores, reaching up to 64.23%. Compared to sparse representations of TF-IDF, this shows that dense vector representations are better at finding semantic similarities between job descriptions.

Shrestha (2020), in her study, highlights the comparison between TF-IDF and Word2vec (including Word Mover's Distance, WMD) techniques for JRS. The study found that Word2vec offers better recommendations due to its ability to capture the semantic meanings of words in job profiles, which TF-IDF lacks. Word2vec showed an accuracy increase of 7% over TF-IDF, and even when further refined with WMD, it showed promising results with a precision and recall increase of 4% over TF-IDF. Despite the promising results, the study acknowledges the limitations due to the size and specificity of the dataset used, suggesting that a larger dataset could improve the accuracy of these models. This conclusion underlines the potential for Word2vec and similar techniques to enhance the effectiveness of JRSs, recommending further research with larger, real-world datasets and online experiments with many users for comprehensive evaluation.

### 2.2.3 Bert in Comparative Experiments for Embedding

In the exploration of novel text representation techniques for comparison in RS, several studies have made significant contributions. In 2021, Lavi et al. created ConSultantBERT, a BERT model that has been fine-tuned and built on top of the Siamese SBERT framework. It performed better than both unsupervised and supervised baselines that use TF-IDF features and pre-trained BERT embeddings. The innovation demonstrated superior capability in handling multilingual and cross-lingual matching, making it a powerful tool for feature representation in JRS. The success of ConSultantBERT is attributed to its fine-tuning on a large-scale, real-world dataset, optimising cosine similarity in embeddings for precise resume-vacancy matching. Singh et al. (2023) explored the domain of social networking, focusing specifically on LinkedIn, to enhance the site's post-recommendation mechanism. They created content-based RS using machine learning, GPT-2, and BERT to provide more personalised content to users. Their research revealed that BERT outperformed other models in identifying similarities between the content users generate and the posts recommended to them, highlighting its capability to significantly improve user engagement and satisfaction on LinkedIn.

Neelima and Mehrotra (2023) offered a thorough examination of word embedding techniques, organising them into traditional, static, and contextualised categories. They emphasized the efficiency of BERT in generating contextualized word embeddings, which, when integrated with NN models, improved accuracy in diverse NLP tasks like sentiment classification and text categorization. Their study highlighted the dynamic development of word embeddings and their vital contribution to replicating human cognitive functions within computational models. Focusing on the critical issue of Islamophobia, the study by Saeed et al. (2023) employs text data mining and NLP techniques to identify Islamophobic content on social media platforms. The study shows that transformer-based algorithms like BERT and GPT are good at classifying texts by using Latent Dirichlet Allocation for topic modelling and Word2Vec and GloVe for feature extraction. The empirical analysis conducted reveals that these advanced techniques, particularly when used in conjunction with traditional textual features, offer a solid strategy for detecting Islamophobic narratives, with the BERT and GPT models achieving notable F1 scores.

## 2.3 Research Gaps

The existing literature on the comparison of text representation techniques within JRS is somewhat limited. A detailed analysis of the available papers on comparison of different embedding techniques within the broader domain of RSs reveal that each embedding method, including traditional and novel techniques, has its own unique advantages and limitations, and their effectiveness can vary widely based on the specific application scenario. And more importantly the inconsistency of these results makes it challenging to draw definitive conclusions on the preferable choice of text representation technique for JRS, given the countless configurations and possibilities available for the system design.

For example, comparisons between TF-IDF and Word2Vec have been made in the context of JRS, yet the outcomes of their effectiveness differ greatly, influenced by the system's architecture and its various components. There are instances where TF-IDF has shown superiority, while in other scenarios, Word2Vec embeddings have been more effective. In some recent studies BERT's was highlighted with its impressive capabilities in numerous NLP tasks, including RSs, by leveraging its encoding features. Yet, the direct comparison of these three embedding models—TF-IDF, Word2Vec, and BERT—within the same JRS framework has not been explored. This oversight represents a current gap in the literature, particularly in evaluating the performance of JRSs using the vector representations provided by these techniques.

This research seeks to address this gap by conducting a thorough comparative analysis of these prominent embedding techniques within a self-designed JRS architecture. It is crafted to evaluate the embeddings through tests including classification accuracy, job ranking efficiency, and human-centric evaluation. By doing so, this study aims to provide a clear, empirical basis for selecting embedding techniques in the development of more effective and nuanced JRS, thus contributing subtle insights to the field.

# Chapter 3. Research Methodology

To address the primary research question, it is essential to conduct an evaluation procedure, and the evaluation of the system is the cornerstone of research methodology in RS, which evolved from experimental practice in Machine Learning (ML) and IR (Castells and Moffat, 2022). Both RS and JRS are typically assessed either online or offline manner. Online assessment involves the real-time evaluation and testing of the recommendation algorithms and models while they are actively being used by users in a live or production environment. This approach is used to measure the system's performance and efficiency under real-world conditions. While it is deemed effective in measuring experimental outcomes, it is a costly and time-consuming approach, generally favoured by only industry experts (Peska and Vojtas, 2020). On the other hand, offline evaluation and offline metrics are most commonly used in academic settings due to their ease of execution, repeatability, speed, and flexibility in accommodating various recommendation models (Gilotte et al., 2018). An offline experiment is conducted using a pre-existing dataset of users and items. Its objective is to closely replicate the data that the system is anticipated to encounter when it is deployed online (Gunawardana and Shani, 2015). And for this research, the offline evaluation and online evaluation are conducted to access and compare three different text representation models in JRS. In addition, it is important to mention that this study employs a unique methodology similar to online evaluations used by industry but diverges by integrating real, volunteer participants who remotely participate in the experiment. This approach allows to combine the scalability and accessibility of online evaluations with the authenticity and commitment of volunteer participants who have proactively agreed to contribute to our research.

To conduct these evaluations, a framework enabling experimentations is necessary. Reviewing the literature revealed the absence of a standardized framework or design to support the proposed evaluations. Consequently, the decision was made to develop a custom JRS specifically tailored to help get answers to the main research question.A close-up of a sign

Description automatically generated Figure provides an overview of the main structure of the custom architecture for JRS, specifically designed for this particular experimental research. The operation of this architecture follows a left-to-right sequence. The process begins by taking input data from job seekers and job listings, then passes through a series of iterative processing units to generate output data containing recommended profiles and job listings. These three components can be considered as the core and central components of the entire system. It’s where the text representation techniques will be applied to and rank the jobs or candidates based on the conditions and details they provided. The following sections will delve into the details of the design and the experimental processes conducted on each component of the system, aiming to find answers to the primary question. Additionally, it will cover the sampling strategies employed, the ethical guidelines adhered to, and the configuration of the experimental setup.

## Sampling Strategy

Careful selection of an appropriate sampling approach plays a crucial role in achieving the objectives of the study, as emphasized by Stratton in 2021. For this particular experiment, it is essential to have two distinct, well-structured datasets: one for individuals seeking employment and another for available job listings. As mentioned in previous section, the job seeker dataset acts as a representation of the experiment's participants, mirroring the demographic characteristics from which the subjects are drawn. Conversely, the job listings dataset represents the 'job listing pool.' These datasets provide the context within which participants engage in the matching and ranking process, employing content based JRS and various text representation methods.

Below, an overview of the selected sampling strategies for each dataset is provided, including specific considerations and underlying rationales informing these choices.

### 3.2.1 Job Seekers’ Dataset

Before selecting participants to represent the job seeking population, it is critical to consider the following sampling limitations:

1. Difficulty in defining the complete job searchers population*,* whether it refers to an entire nation or a smaller geographical region. This complication arises from the multitude of job seeker’s categories, including those who are actively seeking, passively looking, monitoring, or transitioning between these states due to their dynamic nature (Bortnick, 1992).
2. Unlike other populations such as voters, licensed professionals etc., there is often no unified registry or database that encompasses the complete information of all job seekers, which collectively contribute to the limited accessibility of this population.
3. In this situation, the information provided by participants will only be used to evaluate and compare system architectures and interact with the data in the job listing pool. Any findings and conclusions drawn from the use of the sample won’t be used for generalization of the entire population (Yin, 2009).

A circular network with many circles

Description automatically generated with medium confidenceGiven the limitations indicated above, the convenience technique is the most suitable option for selecting participants in this experiment. In addition, it is important to point out that the limitations, as mentioned earlier, fit precisely  *Figure . Advantages of convenient sampling.*

A screenshot of a cell phone

Description automatically generatedthe qualities defined by Etican et al. in 2016. They describe convenience sampling as a non-probabilistic technique in which participants are chosen based on their accessibility and proximity. It is a convenient and cost-effective method that allows researchers to quickly collect data from voluntarily available individuals ready to engage in the experiment (refer to Figure 4). Furthermore, they underscore the common use of this method in situations where obtaining a random or representative sample is challenging and that convenience sampling comes with its own set of limitations, as the samples obtained may not precisely mirror the broader population, necessitating caution when interpreting and generalising results derived from convenience sampling.

Taking into account both the benefits and drawbacks associated with volunteer recruitment, the online social media platform, particularly Facebook, has been chosen for its cost-efficiency and the rapidity with which it allows for sample collection. (Antoun et al., 2016). A link for the Google Form that is constructed for the experiment (Bulgan, 2024), along with a well-articulated post that clearly explained the objectives of the research and the anticipated time commitment, was posted on the local Facebook communities known for daily postings of job offers and job seeker ads (refer to Figure \*). These groups are ‘[Ireland Job Vacancies And Employment](https://www.facebook.com/groups/1618183571730254/)’ (162.9K members), ‘[Jobs Ireland](https://www.facebook.com/groups/jobsearchireland/)’ (128K members), ‘[DUBLIN JOBS](https://www.facebook.com/groups/1415280002101858/)’ (49.0K members) and ‘[Ирланд дахь ажлын зар](https://www.facebook.com/groups/2017664791841533/)’ (6.6K members). Once potential volunteers were attracted, detailed information about participation requirements, risks, benefits, and the rights of participants was provided through personal messages.

Two individuals expressed interest in the research by requesting further details through direct messages on Facebook, and another person completed the Google Form via the shared link. At the time of the initial online meeting, all three were actively seeking employment and after receiving more detailed information regarding to the experiment and its aims, all three people agreed to participate in it by sharing relevant personal information. Two of them gave their written consent through Facebook direct messages, while the third individual marked all queries and sections concerning confidentiality, participation, and consent to share personal information on the Google Form by selecting 'yes', which is equivalent to providing written consent. Afterwards, the volunteers were invited to engage in the first phase of their participation in the experiment – the online interview. An online video call was arranged for each participant, during which they were requested to provide information relevant to their job search by answering prepared questionnaires. The formulated questions were:

1. Name or Nickname. Please provide your name or nickname that you want to use in my research. Be assured that the information provided by you will be exclusively used for research purposes only, and any details that could identify you will remain confidential. You can provide your actual name or full name if you wish. Alternatively, if you're not comfortable with that, you can choose to go by a nickname or a made-up name.
2. Contact information. Please provide the contact details you prefer, whether it's your mobile number, personal email, or any other social media profile that I can use to contact you throughout my research. Be assured that the information provided by you will be exclusively used for research purposes only, and any details that could identify you will remain confidential.
3. Job role of interest. Please provide details about your preferred field of interest or the specific job title you have in mind. Be assured that the information provided by you will be exclusively used for research purposes only, and any details that could identify you will remain confidential.
4. Educational information. Please provide your highest level of education and details regarding your major field of study. Additionally, include information on any other degrees or certificates obtained beyond your highest educational achievement. Be assured that the information provided by you will be exclusively used for research purposes only, and any details that could identify you will remain confidential. You may choose not to mention the names and graduation year of the institution or educational organisation if you're not comfortable providing those specifics.
5. Skills information. Please provide your personal skills information, encompassing both hard and soft skills acquired through prior work experience or training. Be assured that the information provided by you will be exclusively used for research purposes only, and any details that could identify you will remain confidential.
6. Experience information. Please provide your professional experience details. You can mention in this field the length of time spent in previous roles related to your current field of interest. Additionally, highlight any skill-based experiences that are relevant. If there's no relevant experience, feel free to enter <None>. Be assured that the information provided by you will be exclusively used for research purposes only, and any details that could identify you will remain confidential. You may choose not to mention the names of the businesses and organisations where you worked previously if you're not comfortable providing those specifics.

However, the two agreed for a voice call rather than a video session and requested not to record the conversation. Another participant declined all call requests but consented to complete the Google Form. Moreover, all three participants wished to maintain the confidentiality of their age, gender, names, contact details, and any other potentially identifying information. Below, tables present the information gathered from each participant through voice calls and the Google Form. For privacy reasons, participants will henceforth be referred to as 'user\_1', 'user\_2', and 'user\_3' in this experiment.

|  |  |  |  |
| --- | --- | --- | --- |
| **1. participant** | user\_1 | user\_2 | user\_3 |
| **2. data\_collection** | Voice Call | Voice Call | Google Form |
| **3. date** | 2023-12-17T15:30:00.000000 | 2023-12-27T11:50:00.000000 | 2023-12-31T13:39:00.000000 |
| **4. location** | Dublin, Ireland | Dublin, Ireland | Dublin, Ireland |
| **5. preferred\_position** | Registered nurse | Electrician | Data analyst |
| **6. education** | Graduate diploma: Critical care nursing | High school diploma  Vocational electrician certification  Construction safety certification | Degree:  Master of Science in Data Analytics,  Bachelor of Science in BA  Certifications:  Microsoft Certified - Azure Data Scientist Associate,  Google Data Analytics Certificate |
| **7. skill** | patient care  wound care  medical procedures  adult nursing  infection control  diagnostic  time management  communication skills  attention to detail | Circuit testing  Blueprint reading  Fault finding  Electrical wiring  Troubleshooting  Inspection equipment  Installation  Organization  Maintenance  Diagnostic  Independent worker  Safety knowledge | Python  Data Mining and Extraction  Data Analytics and Visualization  ETL Pipeline  Data Reporting  Database Management Systems  SQL and NoSQL  Machine Learning  A/B Testing  Data Governance |
| **8. experience** | Registered Nurse – 3 years | Residential Electrician's Helper 1 year | Entry Level Data Analyst: 1 year  Data Coordinator: 2 years |

*Table . The job seeker’s data*

### 3.2.2 Job Offers’ Dataset

For the data collection of the job offers dataset, the web scraping technique has been used to gather publicly available data from Indeed.com. The domain of web scraping offers various techniques in different formats, each offering unique benefits. In this experiment, the Python library Selenium is chosen as the data collection tool for the job listing pool due to its user-friendliness and robust community support (Chapagain, 2019). The primary aim of employing this tool has been to extract publicly accessible information from the Indeed job platform for use in the experiment. It is crucial to acknowledge that collecting random job listings or the entire job database from the platform is not feasible. Instead, collection was done on the outcomes of a keyword search using the job title information that each participant provided in order to create a validation test that replicates real-world scenarios for our system architectures.

The rationale for the data collection method is structured in this manner: If a participant in the study indicates a preference for positions in finance during their interview, all job listings from Indeed resulting from the keyword search 'nurse' will be scraped. Accordingly On January 10, 2024, a total of 564 job ads related to nursing in Dublin and surrounding areas were gathered from Indeed in relation to the field of the first participant. Subsequently, on January 20, 2024, an additional 194 jobs were collected using the same search criteria to enrich the pool of job listings. As shown in Figure 1, the emphasis is on extracting specific information from these job postings, including the job title, required skills, necessary experience, and educational requirements. All other data were excluded from the experiment for reasons thoroughly explained in the ethical considerations section of the thesis.

A screenshot of a computer

Description automatically generatedFollowing the final extraction, a dataset consisting of 1166 unique job advertisements, corresponding to the job titles of interest to the participants, was extracted from Indeed.com on two separate occasions, January 10th and 20th. It consists of six columns, each providing details such as the job title, job ID, page URL, posting date, assigned position label, and a detailed job description (refer to Figure 0). These columns encompass all the relevant information that was available on the website link. It's important to note again that certain details like salary, contact information, organisation locations, office location, etc. have been excluded from the job descriptions as mentioned earlier.

*Figure: Selection of random rows from the final DataFrame containing the extracted job ads.*

## Experimental Setup

The experiments were conducted using a Jupyter notebook on a computer equipped with the following hardware and software:

* Operating System: Windows 10, version 10.0.22631-SP0
* Python Environment: Python 3.11.4, release date July 5, 2023
* Processor: Quad-core CPU with a speed of 2496 MHz
* Graphics Card: NVIDIA GeForce GTX 1650
* Memory: 31.87 GB RAM
* Storage: 237.45 GB on the hard drive

Detailed documentation of the experiment's execution and the chronological coding activities are available on the GitHub repository at the following URL: <https://github.com/temulenbd/capstone_project>

## Implementation of Experiment 1

### Experiment Objective

The main objective of the first experiment is to assess the quality and determine which of the three different text representation techniques is most appropriate for the Classification Unit within the proposed JRS framework.

### 3.3.2 Unit Desing

As previously discussed, the proposed JRS consists of three distinct main components, with the first being the Classification Unit. Figure 2 visually represents the structure and flow within this segment. The workflow in this unit begins by acquiring input texts from both new users and job listings. These texts are then directed into a text representation phase, where they undergo transformation into a numerical format, enabling interpretation by machine learning techniques. After this conversion, the numerically encoded texts are forwarded to the classification phase. At this stage, a variety of classification models can be employed, ranging from basic to more sophisticated ones. This phase is adaptable, allowing for the use of various classification models, from basic to more advanced. For this specific study, Logistic Regression (LR) was selected as the classifier because the primary focus of this experiment is on the influence of different text embedding methods. This classifier then evaluates the embedded numerical data of user and job information to determine the probability of a categorical outcome, such as the job's category or label. Following classification, the text is further enhanced with additional details before being added to the pool of job listings or user profiles.

### Data preparation

The primary data for this first experiment comprises two primary datasets: <job ads> and <job seekers>, both of which were specifically collected for this research. Initial preprocessing was applied to both datasets shortly after collection, given their raw state, which was directly sourced from websites and derived from interview reports with individuals. Particular focus was given to the <job ads> dataset. To adhere to ethical guidelines, personal and sensitive information such as email addresses, salary figures, phone numbers, contact details, organisational locations, and any links were eliminated from the raw text. Throughout this specific experiment, depending on the chosen text representation algorithm, various tailored preprocessing techniques were implemented.

For example, in this experiment, the <bert-base-uncased> variant is used, which inherently disregards text case by converting all input text to lowercase during its tokenization phase. And as for the tokenizer, the <AutoTokenizer> is employed to automatically determine and load the suitable tokenizer for the selected pre-trained model. Specifically, the BertTokenizer, which uses the WordPiece tokenization method, is deployed for this study. This choice is particularly relevant to the classification task, as this tokenizer appends special tokens at the beginning and end of text inputs (<padding> argument), simplifying data preparation without necessitating additional preprocessing steps. However, a notable constraint of this model is its limitation to handling only 512 tokens per input. To circumvent this issue, a custom <custom\_tokenize> function was developed. This function is designed to selectively use portions from the start and end of longer texts, thereby adhering to the token limitation and ensuring the model accommodates the full breadth of the dataset effectively.

The preprocessing approach for data intended for Word2Vec embedding involves a somewhat different methodology. Only basic preprocessing steps such as converting text to lowercase and removing punctuation marks were implemented, bypassing lemmatization and grammar correction. This is because the Word2Vec model in use has been trained on a vast corpus, possessing a comprehensive vocabulary that accommodates various forms of words. An important aspect to highlight is the application of a <custom\_function> during the embedding process. This customised function guarantees that a zero vector is returned whenever a token cannot be located in the model's vocabulary. It prevents such words from affecting the overall aggregation of the embeddings.

TF-IDF necessitates the implementation of conventional preprocessing steps such as converting text to lowercase, eliminating punctuation, removing stop words, performing lemmatization, and tokenization. These steps are facilitated through the use of functions from the NLTK (Natural Language Toolkit) and regular expressions libraries in Python.

### Train-test split

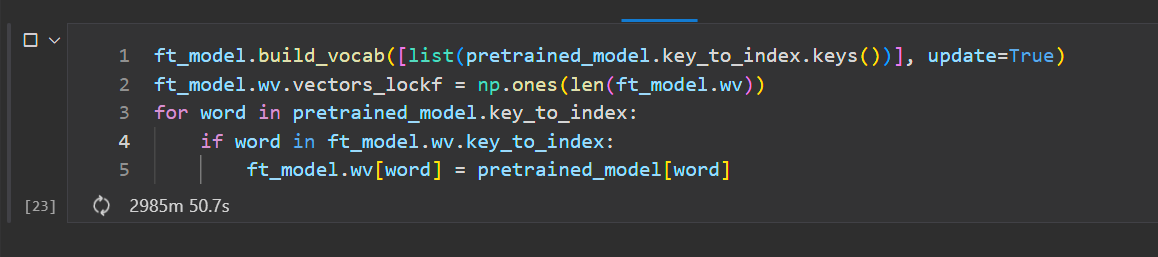
Using <Scikit-learn>'s <train\_test\_split> function, the pre-processed data was divided into train, test, and validation sets. The distribution of the data across these sets is as follows: the training set comprises roughly 62.95%, the validation set about 7.03%, and the test set approximately 30.02% of the original dataset **(refer to figure ).**

### Procedure

During the experiment the sole variable that undergoes alteration is the embedding techniques; all other components and workflows within the unit remain constant.

1. BERT: For the experimentation using Bert embeddings, the pre-trained <bert-base-uncased> model was selected and later fine-tuned for downstream task with the primary job ads dataset. The model, initially sourced from the Hugging Face library, underwent fine-tuning with the <train()> function from the transformers module. The effectiveness of the fine-tuning process was monitored via a custom <compute\_metrics> function, which provided insights into precision, recall, and F1 scores after each training epoch. Notable hyperparameters adjusted during fine-tuning included:

* Optimizer: The <Adam> optimizer was chosen due to its adaptive and efficient nature, making it highly compatible with the BERT model's optimisation needs.
* Loss Function: The <CrossEntropyLoss>, used alongside a custom trainer class named <WeightedLossTrainer>, aimed to address data imbalance issues inherent in the job ads dataset.
* Learning rate: A learning rate of 2e-5 was chosen as a smaller learning rate because it ensures the pre-training weights are adjusted gradually, preventing the model form from deviating too drastically from its pre-trained state.
* Epochs: The experiment was limited to three epochs to minimise the risk of overfitting.
* Batch Size: A batch size of 16 was determined to be optimal, as increasing to 32 led to memory constraints and potential device crashes.
* Weight Decay: Set at 0.01, weight decay was implemented as a regularisation measure to combat overfitting.
* Evaluation Strategy: An epoch-based evaluation strategy was adopted to review training outcomes after each epoch.
* FP16 Training: Enabling FP16 precision was a strategic choice to expedite the training process.
* CUDA Utilisation: The creation of a CUDA device further accelerated training, leveraging GPU capabilities for enhanced performance.

1. **Word2Vec: In the experimentation phase, an attempt was made to fine-tune the Word2Vec model by creating a new corpus and copying vectors from a pre-trained model. The aim was to enrich the corpus with these fine-tuned vectors and adjust the weights with additional training on a job ads’ dataset. Unfortunately, the vector transfer process was not completed due to its extensive duration, extending over several days without completion (refer to Figure 2). Consequently, the decision was made to utilise a pre-trained Word2Vec model, specifically trained on Google News data, featuring 300 dimensions, for the classification task.
2. Pimpalkar and Raj (2020) demonstrated in their research that combining BoW with TF-IDF significantly enhances the performance of machine learning classifiers. This experiment adopts a similar approach for embedding, utilizing the <CountVectorizer> and <TfidfVectorizer> from <Scikit-learn>'s feature extraction modules for vector transformation. The pre-processed dataset of job advertisements underwent both transformations separately, after which the <hstack> function from SciPy's library was employed to concatenate them into a single dense vector representation.

### Evaluation metrics

Regarding the assessment metrics, considering that the study employs a basic Logistic Regression (LR) model for classification with an emphasis on comparing various embedding methods only, the selection of detailed evaluation metrics—namely accuracy, precision, recall, and F1 score, in addition to the confusion matrix—was considered sufficient.

### Benchmarking:

Given that the primary dataset for this study consists of real-life data, unique to this experiment and not previously utilized in other research, concerns about the accuracy of the results naturally emerge. To address these concerns and ascertain any discrepancies between this experiment and established benchmarks, benchmarking was conducted using the well-known AG News dataset, a standard for classification tasks. Due to computational resource constraints, the experiment did not cover the entire benchmark dataset; instead, a random subset was chosen. This subset was specifically selected to reflect the imbalances present in the main dataset, enabling a fair and relevant comparison under similar conditions.

## Implementation of Experiment 2

### 3.4.1 Experiment Objective

The main objective of the second experiment is to assess the quality and determine which of the three different text representation techniques is most appropriate for the cosine calculation stage within the Ranking unit of the custom-built JRS framework.

### 3.4.2 Model Design

**Figure 223** illustrates the structure and functioning order of the Ranking Unit within the custom-built JRS. The architecture of this unit involves a job ad pool, a users’ pool, a cosine calculation stage, a ranking stage, and a ranked job list for the corresponding user. The initial process begins by receiving a pool of job advertisements labelled similarly to the user's label that is selected (it's noteworthy to mention that for this experiment, all 1166 job ads and all experiment participants are utilized). This approach aims to increase efficiency by eliminating unnecessary computations. Labelled ads and the candidate's data come from the output of the Classification Unit, where they have been sorted into categories and assigned vector values using one of three text representation methods. At this point, each job ad receives a score reflecting its similarity to the user's profile. After the calculation is completed, the data moves to the ranking stage, where jobs are ranked from 1 to 3 based on a normalisation function. The end result is a list of job ads ranked from the most to the least relevant to the selected candidate, which is then passed on to the subsequent processing unit.

### 3.4.2 Experiment Dataset

For this experiment, the dataset to be utilised originates from the output of the Classification unit, albeit in a slightly different format. As detailed in the Procedure section of Experiment No. 1, tokenization was previously performed using a custom function that divided the text into head and tail sections, adhering to the 512-token maximum limitation imposed by the BERT model. However, directly truncating the text to fit within this limit would result in a biassed and inaccurate score for the cosine similarity calculation. To circumvent this issue, a custom function named <embed\_with\_bert> was employed. The purpose of this function is to segment the text into tokens via <BertTokenizer>, calculate the total token count, and apply the <sliding window> technique if the number of tokens surpasses the model’s limit. This technique is used for processing sequences of text by moving a fixed-size window across it by continuously extracting each value on the window and is commonly used in this type of situation (Zhao, 2022). Regarding Word2Vec and TF-IDF embeddings, no adjustments were made; the same vector values used in the classification stage are applicable and suitable for this experiment.

### 3.4.3 Procedure

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedAs noted in the preceding sub-section, the sliding window technique was employed for encoding with BERT, specifically targeting the extraction of the last hidden state, which is the output from BERT's 12th encoder layer. To expedite computational speed, the format for this last hidden state was selected to be a Torch Tensor, owing to the use of a CUDA device. Furthermore, as illustrated in **figure x**, it is evident that both tensors, which are the final vector representation for each job ad, share identical dimensions in their third axis. This implies that each token in the second dimension is represented by a consistent feature vector of 768 elements. This characteristic holds true for all textual tensor representations in this context. Consequently, this uniformity across the feature vectors complicates the process of performing cosine similarity calculations for tensors derived from BERT's encoder when compared to TF-IDF or Word2Vec embeddings. Unlike Word2Vec embeddings, where it is feasible to aggregate vectors and calculate their meaning, such an operation is not directly applicable to BERT embeddings. Therefore, cosine similarity scores for BERT-encoded data were computed individually for each token's matrix, followed by an aggregation of these scores to determine their mean value (**refer to figure**).

The process of computing cosine similarity for Word2Vec and TF-IDF embeddings is more straightforward compared to BERT due to their simpler vector structures. For instance, after applying Word2Vec, every vector is uniform in shape, specifically (300), indicating that each data point is represented by a fixed 300-dimensional vector post-aggregation. This uniformity simplifies the cosine similarity calculations. Similarly, TF-IDF, despite employing a different approach, is less complex than BERT. It involves generating a corpus for each word from all occurrences during tokenization and transforming each word accordingly. As a result, each TF-IDF vector for random job advertisements maintains a consistent shape of (20162,), facilitating easier computation of cosine similarity.

### 3.4.4 Evaluation

The evaluation of the experiment is conducted differently compared to the first experiment. In this case, exploratory data analysis (EDA), correlation tests for observations, and statistical tests were employed to draw conclusions on the cosine scores calculated for each text representation technique. And the Ranking model evaluation test is conducted with human evaluation with the next experiment.

## Implementation of Experiment 3

### 3.5.1 Experiment Objective

The primary aim of the final experiment is to evaluate the quality and ascertain the most fitting text representation technique among the three for use in the Recommendation Unit and the ranking phase of the suggested JRS framework.

### 3.5.2 Experiment Design

The last unit of the proposed Job Recommendation System (JRS) that is designed for evaluation of different text representation techniques is the Recommendation Unit. This unit's structure and workflow are relatively simple: it takes in the ranked job listings and the focus candidate data from the preceding unit. Then, by applying specific custom conditions, it either selects a list of recommended jobs or a single recommendation, which can be tailored in any way. For this particular study, instead of using the conventional top-N recommendation list—which would traditionally align with the core aim of a RS—the experiment utilises a list that represents the full spectrum of ranked jobs from lowest to highest. The primary method involves using stratified random sampling to choose 30 jobs that meet certain criteria, such as:

* True Positive
* False Positive
* Negative

And most importantly, unlike the two earlier experiments, the last one encompasses both the Ranking and Recommendation Units, it also integrates human evaluation to draw conclusions about the embedding techniques used in the proposed JRS framework.

### 3.5.3 Experiment dataset:

As previously mentioned, this experiment will involve the last two units of the custom JRS, assessing the quality of both the ranking and recommendation phases through a human evaluation experiment. The dataset for this final assessment is derived from the cosine similarity scores generated during the Ranking unit's calculation stage. To evaluate the dataset obtained from various text representation techniques, a ranking system must be established for comparison against a human-generated, logically coherent ranking. Given the challenge of data imbalance, it's impractical to directly create ordinal ranks (1st, 2nd, 3rd, etc.). A more viable option is to rescale the original scores within a specific range, maintaining the relative differences between scores. This method facilitates the identification of logical discrepancies using the numerical value of labels, thus enabling the direct assignment of ranks from 1 to 3. Consequently, the procedure for directly assigning ranks from 1 to 3 is intricately linked to the classification count of job advertisements. By sorting the final dataset's rows in ascending order based on cosine scores from each method, rows that exceed the total job ad count for each label in the dataset can be interpreted to fall within a ranking spectrum of 2 to 3 (with 2 indicating the job recommendation is accurate but not an ideal match in terms of detailed qualifications, or a false positive, and 3 signifying a perfect match between the job recommendation and the participant's qualifications, or a true positive). Entries below the job ad count threshold for each label are given a rank of 1, indicating an incorrect job recommendation or a negative. This strategic approach to ranking enables the creation of lists of 30 job advertisements for human evaluation for each participant, tailored to their profile.

### 3.5.4 Evaluation

Regarding human evaluation, new datasets containing 30 job advertisements for each text representation technique were distributed to the experiment participants. Participants received a link to access the job ads along with a blank space to assign rankings ranging from 1 to 3. They were instructed to evaluate and assign a rank to each job ad based on their judgment and return their assessments. All three participants completed the ranking for each job advertisement and submitted their responses for the final analysis.

## 4.1 First Experiment Results (Classification)

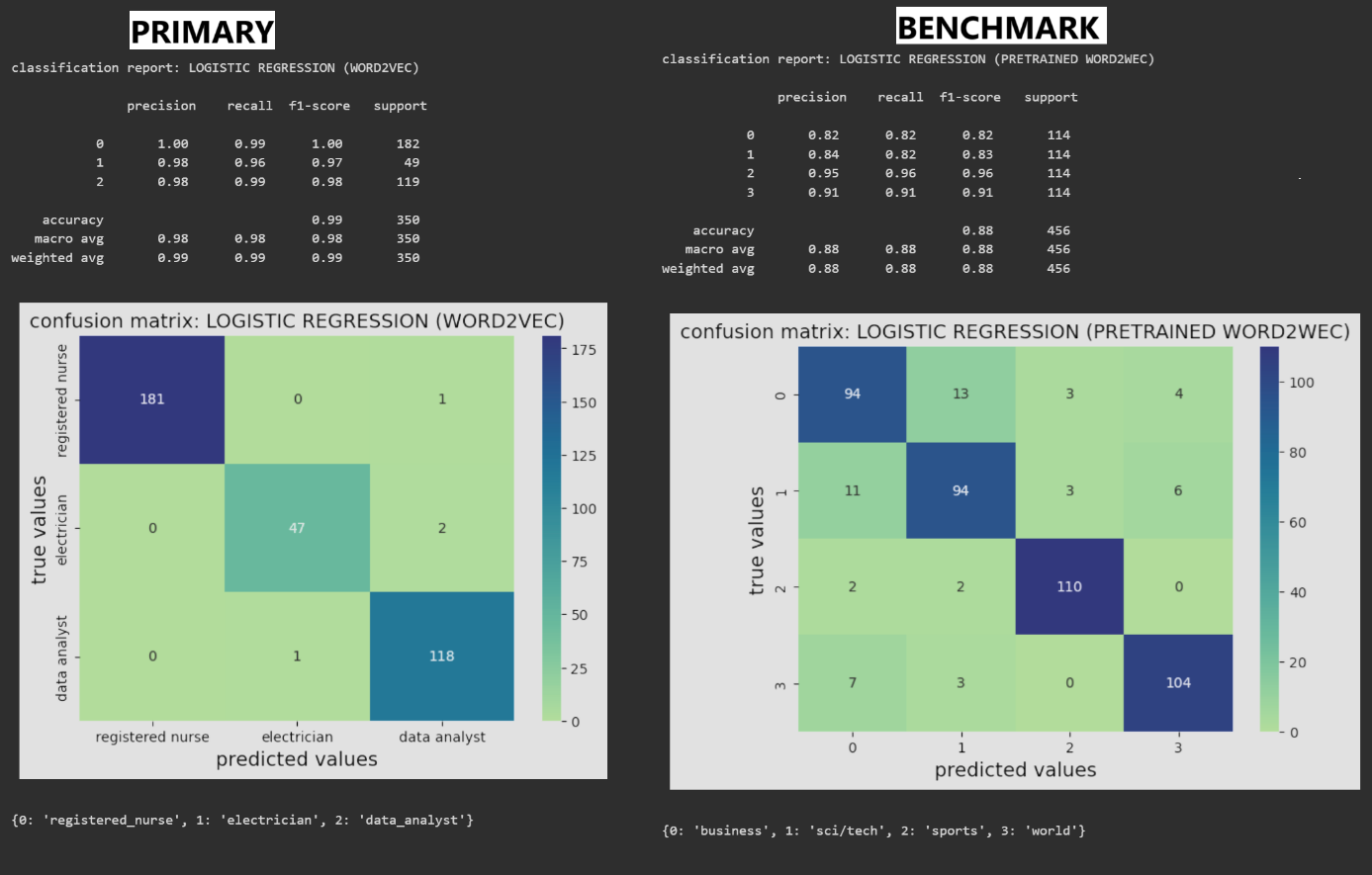
In the implementation phase, a Logistic Regression (LR) model was employed, focusing on the comparison of various embedding methods in the Classification Unit of the custom designed JRS. Detailed information about the unit’s architecture, data preparation, and model development processes is provided in the methodology section (Sub-section 3.3). This section concentrates on the outcomes and insights derived from applying these implementation strategies. The results of each embedding technique are analysed separately before integrating the final results.

### A screenshot of a graph Description automatically generated4.1.1 Evaluation of BERT Encoding

The entire process of fine-tuning the model and assessing its performance on the primary dataset took 45 minutes and 42 seconds. Following the adjustment of the bert-base-uncased model on the primary dataset and subsequent fine-tuning on the benchmark dataset over three epochs, the following outcomes were observed, as detailed in the referenced figure no.

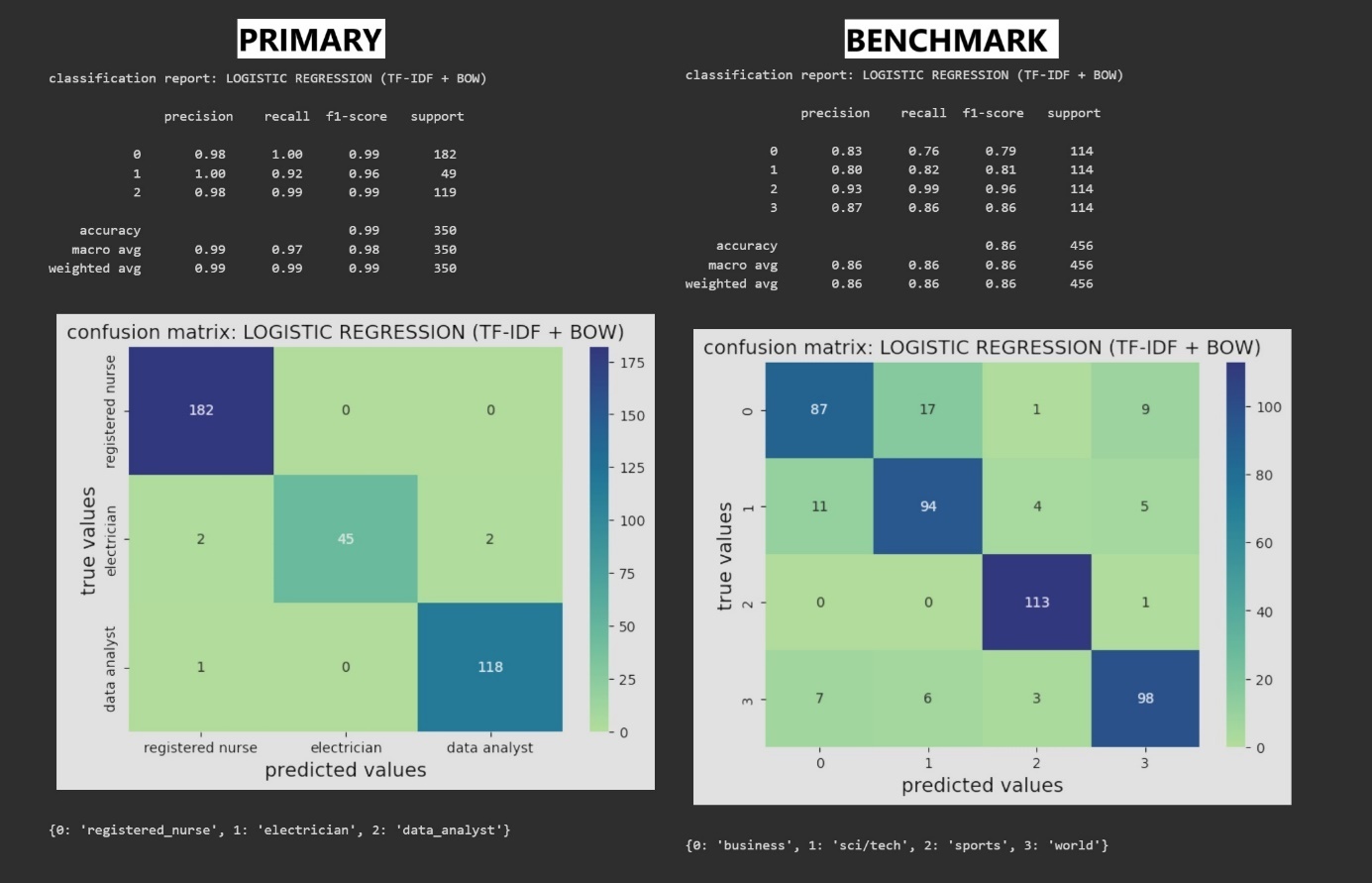
For the primary data set, the model achieves perfect performance across all metrics (loss, accuracy, F1 score, precision, and recall) from the first epoch, maintaining this level in subsequent epochs. This indicates that the model fits the primary data set exceptionally well from the start. On the benchmark data set, there's a notable improvement across epochs. In the first epoch, the model shows a loss of 0.430, with accuracy at 0.881, and very similar values for F1 score, precision, and recall, indicating good but not perfect performance. By the third epoch, loss decreases to 0.392, and there's a slight increase in accuracy to 0.897, along with corresponding improvements in F1 score, precision, and recall, suggesting that the model's ability to generalise improves with additional training.

### 4.1.2 Evaluation of Word2Vec Embedding

Regarding the Word2Vec embedding, the Classification Unit's process had been completed in merely 31 seconds for the entire cycle, which was probably due to the model being preloaded and pretrained. As depicted in **Figure**, the classification report for the primary dataset showed exceptionally high precision, recall, and F1 scores for all classes. Notably, the class for <registered nurse> had achieved perfect scores of 1.00 across all three metrics. Moreover, the classes for <electrician> and <data analyst> had also shown very high-performance metrics, with all scores exceeding 0.96. The overall accuracy of the model had reached 0.99, an extraordinarily high figure, suggesting that the model's performance on the primary dataset was exceedingly robust and comparable to that achieved with BERT embeddings. In reference to the benchmark dataset, the classification report indicated lower precision, recall, and F1 scores compared to the primary dataset, with scores ranging from 0.82 to 0.95. The overall accuracy for the benchmark dataset had been recorded at 0.88, which, although lower than the primary dataset's accuracy, still signifies a strong performance level.

### 4.1.3 Evaluation of TF-IDF Embedding

The Classification Unit’s cycle for the primary dataset using TF-IDF and BoW embeddings was executed in a mere 6 seconds, marking it as the quickest embedding method for processing nearly 1200 rows of job advertisement ads, in contrast to Bert and Word2Vec.The classification report displayed in the figure indicates outstanding precision, recall, and F1 scores for all classes for the primary dataset, similarly to BERT and Word2Vec. The class of <registered nurse> achieved perfect recall, very high precision, and an F1 score, leading to an average accuracy of 0.99 for the model. The <electrician> and <data analyst> classes also showed high metrics, all above 0.96, contributing to a macro and weighted average of 0.99 for both precision and F1 score and 0.97 for recall. There are very few misclassifications overall, suggesting that TF-IDF and BoW embeddings have provided a strong foundation for classification on the primary data. In contrast, the benchmarking report presents slightly lower results, with precision, recall, and F1 scores ranging from 0.76 to 0.99. The overall accuracy of the model on the benchmark dataset is 0.86, which, while not as high as the primary dataset's, still represents a relatively solid performance.



### 4.1.3 Experiment Conclusion

A comparative experiment of the three text representation methods—BERT (encoding), Word2Vec (embediing), and TF-IDF with BoW—has been conducted within the framework of the Classification Unit for the custom-developed JRS. When evaluating performance, all methods exhibited high effectiveness on the primary dataset, with a slight decrease observed on the benchmarking. Among them, the embedding performance of the BERT model achieved the most superior results overall compared to the other two techniques. In terms of efficiency and processing speed, TF-IDF distinguished itself with its exceptionally fast processing capability. In summary, each of the three models demonstrates suitable qualities to serve as the embedding technique for the Classification Unit in the suggested JRS.

## 4.2 Second Experiment Results (Cosine Calculation)

This section focuses on examining the cosine similarity scores derived from three distinct text representation methods to conclude and address the experiment's objective. Unlike the analysis conducted in prior experiment, the organization of this section varies. The analysis begins with assessing the cosine similarity scores for the job advertisements related to each participant in the experiment. Subsequently, it examines the outcomes of the statistical analyses performed. This is followed by the evaluation of the ranking model. In conclusion, it encapsulates the experimental results.

### 4.2.1 EDA

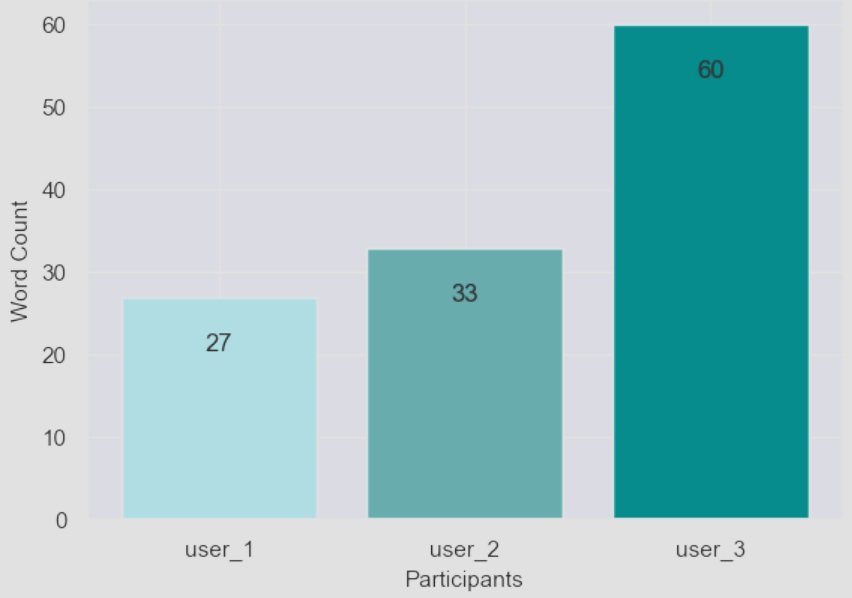
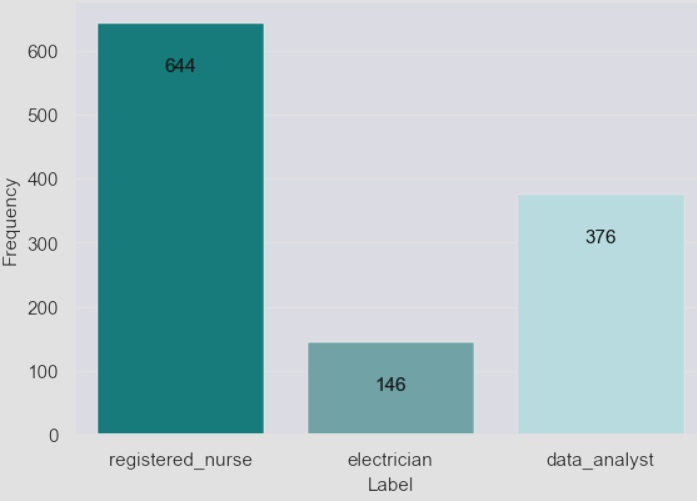
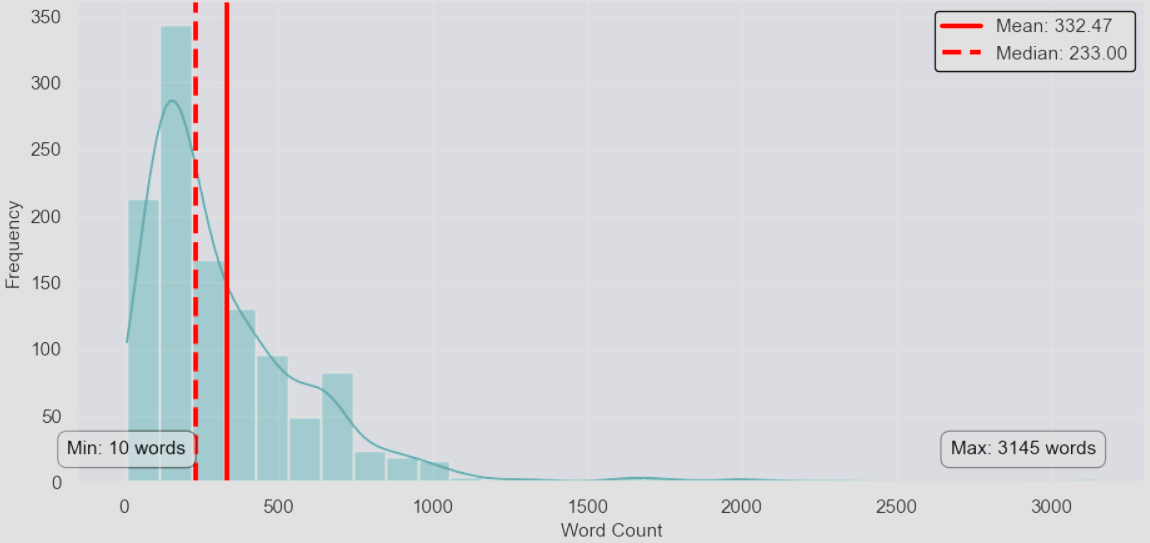
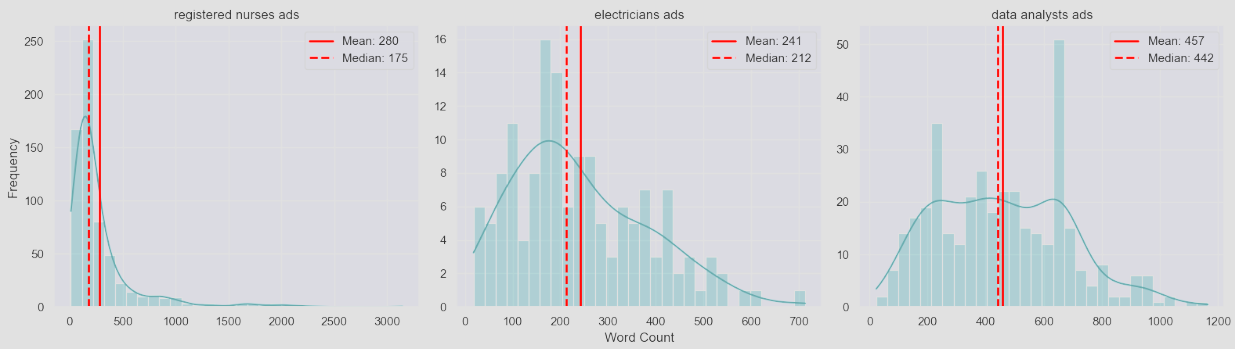
*Job seekers dataset:* The <job seekers> dataset contains information of three individuals participating in the experiment. The participants are based in the Dublin area, and their data was collected on various dates in December 2023. They shared their job search details through voice calls and Google Forms. Notably, each participant expressed their preferred search for job positions and provided detailed information regarding their education, skills, and work experience.

Figure 1 illustrates that the data utilized for computing the cosine similarity for each participant varies in word count after preprocessing, yet all are under 100 words. The most extensive entry is from participant number 3, comprising 60 words, and the briefest is from participant number 2, with 27 words.

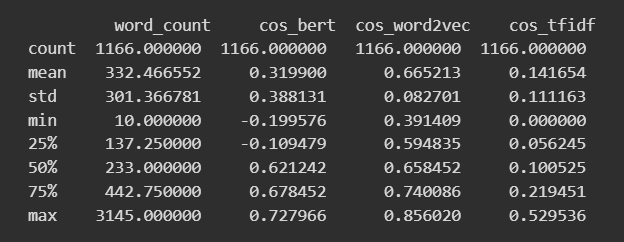
*Job seekers dataset:* The <job ads> dataset contains data from 1166 unique job advertisements, which were collected from Indeed.com on two separate occasions, January 10th and 20th. It consists of six columns, each providing details such as the job title, job ID, page URL, posting date, assigned position label, and a detailed job description. These columns encompass all the relevant information that was available on the website link. It's important to note that certain details like salary, contact information, organisation locations, office location, etc. have been excluded from the job descriptions to comply with ethical standards.

As it can be seen from the figure the total count of gathered job ads stands at 1,166, with a considerable majority focused on the registered nurse position. This suggests that, among the three professions collected, registered nurses were the most in-demand on Indeed.com during January. The role of electrician had the fewest listings, totaling 146, while data analyst positions were intermediate, featuring in 376 job advertisements over the same period.



The word count for each job ad varies from a minimum of 10 words to a maximum of 3,145 words. The average word count for all job ads is 332, with a median of 233 (refer to figure.). The lower median value indicates that more than half of the data points fall below the mean. The rightward skew of the mean suggests the influence of high-value outliers or the long tail on the dataset's average. The job ads for nursing positions are not only the most in-demand job among the three categories but also have the highest word counts, with some exceeding 3000 words. However, the mean word count for these positions is relatively low at 280, especially when compared to the mean word count for data analyst job positions. Conversely, electrician job ads prove to be the least popular among the three categories on the website, characterised by the lowest word counts and mean word counts (refer to figure.).

*Cosine score summary statistics regarding to each user:*

1. USER1 = NURSE (r**efer to figure**):

-\*word\_count\*: The statistics for word count remained consistent across the DFs for all three users, as each user's cosine similarity was calculated using the same set of job advertisements. The mean word count stands at approximately 332 words, suggesting that the average length of the text extracted from Indeed.com is somewhere around 330 words. Nonetheless, there is a wide variation in the length of job ads, with word counts ranging from as few as 10 to as many as 3145 words, and the standard deviation is about 301 words, suggesting a wide variation in text lengths, from very short to quite lengthy documents. The 25th percentile is around 137 words, the median (50th percentile) is 233 words, and the 75th percentile is about 443 words, confirming the broad spread in document lengths.

-\*cos\_bert\*: The average <cos\_bert> is 0.3199, indicating a moderate level of semantic similarity across the documents when analysed with BERT. All scores generally range from -0.1996 to 0.728, showing that some document pairs are seen as somewhat dissimilar (with negative values) while others are highly similar. The data exhibits a wide distribution (std of 0.3881), suggesting varied degrees of semantic similarity among document pairs as per BERT's analysis.

- \* cos\_word2vec\*A higher average cosine similarity of 0.67 for <cos\_word2vec> score indicates a generally higher level of similarity across documents using Word2Vec embeddings compared to BERT. The standard deviation is lower (0.08) than for BERT, suggesting less variability in similarity scores with Word2Vec. The minimum and percentile values suggest a range of similarities, with most documents being moderately to highly similar.

-\*cos\_tfidf\*: The average cosine similarity of 0.14 for < cos\_tfidf> is much lower than for the other two methods, indicating a lower level of textual similarity across documents when using TF-IDF + BoW vectors. A lower standard deviation (0.11) and the range of values suggest that, while there is variability, documents tend to be less similar to each other based on TF-IDF scores.

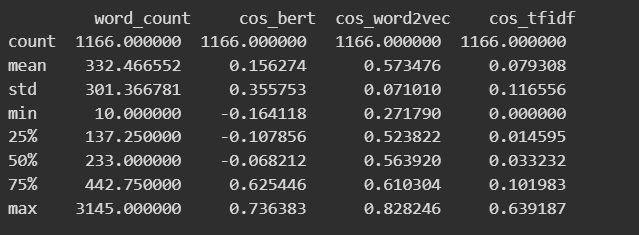
1. A black screen with white numbers

   Description automatically generatedUSER2 = ELECTRICIAN (r**efer to figure**)

-\*cos\_bert\*: The average cosine similarity is -0.025, suggesting a slight overall dissimilarity between documents when analyzed with BERT embeddings. This is a extreme contrast to the Nurse's data, which had a positive mean of 0.67, indicating more similarity. The standard deviation is lower (0.25) compared to the Nurse's, indicating less variability in the similarity scores. The range from -0.243 to 0.720 shows some documents are very dissimilar while others are quite similar, albeit the majority lean towards dissimilarity.

- \* cos\_word2vec\*: The average similarity score of 0.60 for <cos\_word2vec> suggests a moderate level of similarity between documents, lower than that observed for the Nurse. This indicates that the documents associated with the Electrician share fewer semantic similarities. A relatively low standard deviation (0.057) suggests that the similarity scores are more consistent across the dataset than those for the Nurse. The minimum and maximum values indicate a narrower range of similarity scores compared to the Nurse's data.

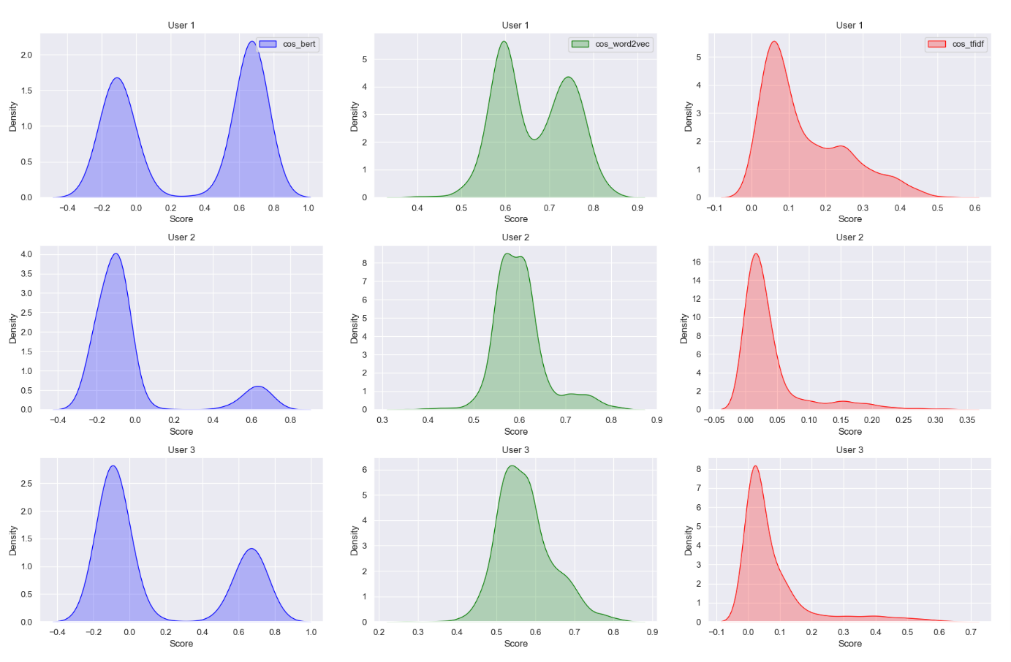
The average similarity of 0.039 score of <cos\_tfidf> is significantly lower than that of the Nurse, suggesting that the documents are generally quite dissimilar based on the commonality and importance of terms. The standard deviation and the range of values indicate a relatively low variability in similarity scores, with most documents being less similar to each other.

1. USER3 = DATA ANALYST (refer to figure)

-\*cos\_bert\*: The average cosine similarity of 0.156 indicates a moderate level of similarity between documents, with a positive tilt suggesting some degree of similarity in content or context when analyzed with BERT embeddings. A relatively high standard deviation (0.356) suggests a wide variability in similarity scores, indicating that some document pairs are found to be quite similar while others are more dissimilar. The range from -0.164 to 0.736 shows variability in document similarity, with the maximum score indicating very high similarity in at least some document comparisons.

-\*cos\_word2vec\*: An average similarity score of 0.573 suggests a moderate level of semantic similarity between documents, albeit lower than observed for the Nurse but higher than the Electrician. The standard deviation (0.071) indicates less variability in similarity scores than BERT, suggesting more consistent semantic relationships across the dataset. The range from 0.272 to 0.828 reflects a spectrum of semantic similarities, from relatively low to very high.

-\*cos\_tfidf\*: The average similarity of 0.079 is higher than that of the Electrician but lower than the Nurse, indicating a moderate level of dissimilarity based on term frequency and document uniqueness. The relatively high standard deviation (0.117) and the range up to 0.639 indicate that while many documents share few common terms, some pairs share a significant amount of unique terms, leading to higher similarity scores.

1. THE SUMMARY: Overall, the summary statistics for similarity scores across all three users show that the length of job advertisements can vary significantly, with some ads containing as few as 10 words and others extending up to 3000 words. The BERT model demonstrates the widest range in document similarity, from -0.243 to 0.736, suggesting that it is more capable of identifying both similarity and dissimilarity in job ads compared to the other models. The Word2Vec model, on average, shows the highest mean similarity score across users, implying that it may not be as effective in distinguishing dissimilarities between texts in job ads. This observation might extend to the TF-IDF + BoW model as well, given that their text vector representations did not produce any cosine similarity scores below 0, indicating a potential limitation in detecting dissimilarities. Additionally, both the TF-IDF + BoW and Word2Vec methods exhibit a lower standard deviation in similarity scores compared to BERT, suggesting they maintain a more consistent level of similarity or dissimilarity across the dataset.

*Distribution:* The density plots on the **Figure** shows that across all three users, the cosine similarity scores generated by the BERT model exhibit a bimodal distribution with two distinct peaks and spanning a broader range of intervals compared to the other two models. It suggests that this particular encoding method may possess the ability to capture a richer spectrum of contextual information, varying in levels of semantic similarity. Furthermore, both peaks observed for the BERT model in all three instances are situated on both the negative and positive sides, indicating that the provided text for scoring produces either negative or positive similarity scores. Conversely, the TF-IDF + BoW cosine score distribution demonstrates a right-skewed pattern across all three instances, suggesting the presence of texts with higher similarity scores as the distribution extends towards the right. In contrast, the Word2vec score distribution exhibits a mixture of one multimodal graph and two unimodal distributions. The secondary peak within the multimodal distribution appears less pronounced. With all peaks clustered between 0.6 and 0.8, it indicates that this method's similarity scoring predominantly generates high degrees of similarity in most cases. Furthermore, the TF-IDF + BoW and Word2Vec techniques producing only positive scores may indicate an inability to effectively differentiate between textual dissimilarities.

A screenshot of a computer

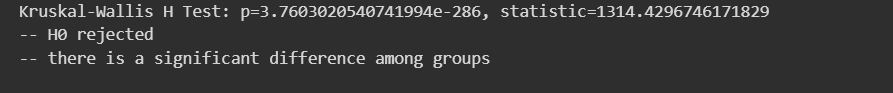
Description automatically generatedCorrelation: The heatmap matrix (**refer to figure**) on each user's data reveal that there is neither a high nor a moderate correlation between the word counts of a text and the cosine similarity scores for all users. This suggests that the length of the text does not significantly impact the calculation of cosine scores. However, there is a high correlation observed among the cosine scores generated by the three techniques. This implies that, while the distribution of scores may vary across different embedding techniques, there appears to be some similarity in the direction of a linear relationship between them.

### 4.2.2 Statistic Test

*Kruskal-Wallis test*

Upon observing that the dispersion of cosine similarity scores notably differs across various techniques (BERT, Word2Vec, TF-IDF+BoW), it's crucial to acknowledge that there remains a correlation between them despite these variances. Graphs above, suggests that the scores may not be the same, but they tend to follow similar patterns of linear correlation. Determining which method exhibits significant score variations is crucial for accurate comparisons among the models. As shown in **Figure 5,** each method displays distributional disparities, encompassing variations in skewness, modalities, and spread. However, to confirm whether these disparities in embedding methods hold statistical significance, conducting a test is necessary. Given the non-normal distribution of the data for each distinct methods, the appropriate test to employ in this scenario is the Kruskal-Wallis test.

The null hypothesis and alternative hypotheses for the Kruskal-Wallis test are as follows:

* H0: The median cosine similarity scores are the same across all three text embedding methods (BERT, Word2Vec, and TF-IDF).
* HA: At least one text embedding method results in a median cosine similarity score that is significantly different from the others.
* alpha = 0.05 (A common significance level value)

After computation, the *p* score is 3.7603020540741994e-286 and the *test statistics* value is 1314.4296746171829 (r**efer to figure**). Based on the aforementioned values, it is evident that there exists compelling statistical evidence to reject the null hypothesis. It can be concluded that a statistically significant difference exists in the median cosine similarity scores derived from comparing document pairs through various text embedding techniques (BERT, Word2Vec, and TF-IDF+BoW). It also suggests that the choice of embedding method has a significant impact on the cosine similarity scores, and therefore, different embedding methods may not be equivalent in terms of how they represent document similarities.

*Cliff's Delta test*

The outcomes of the Kruskal-Wallis test provided compelling statistical support for rejecting the null hypothesis, indicating a significant variance in cosine similarity scores among one of the groups. Nevertheless, this test doesn't quantify the extent or importance of these discrepancies. This is where the Cliff's Delta test proves valuable: it complements the Kruskal-Wallis test by not only pinpointing significantly distinct pairs of groups but also quantifying the magnitude of these differences. Therefore, the calculation of delta scores is crucial for reaching conclusions about the effectiveness of the embedding strategies.

The delta scores span from -1 to 1, with scores closer to 0 indicating minor effects and those nearer to -1 or 1 signifying substantial effects. Once these scores are computed, they can be interpreted using a benchmark guideline that categorizes them as follows:

* negligible (∣d∣ < 0.147): negligible effect size suggests that the difference between the two groups is so small that it is considered insignificant for practical purposes.
* small (0.147 ≤ ∣d∣ < 0.33): small effect size indicates a slight but noticeable difference between the groups, which might be meaningful depending on the context.
* medium (0.33 ≤ ∣d∣ < 0.474): medium effect size reflects a moderate difference that is likely to be of practical significance in many situations.
* large (∣d∣ ≥ 0.474): large effect size implies a substantial difference between the groups, suggesting a strong effect that is very likely to be important in most contexts.

And using this benchmark the results of the effect size on the dataset can be interpreted as follows (re**fer to figure**):

* BERT – TF-IDF+BoW (0.1057): This magnitude of Cliff's Delta indicates a negligible effect size; it suggests that the difference in distributions between BERT and TF-IDF+BoW cosine similarity scores is statistically detectable but small in magnitude. The positive value indicates that, on average, BERT tends to yield slightly higher cosine similarity scores than TF-IDF, but the difference is not substantial.
* BERT – Word2Vec (-0.4331): This delta indicates a medium effect size; it suggests that the difference in distributions between BERT and Word2Vec cosine similarity scores is not only statistically significant but also of a substantial magnitude. The presence of a negative sign in the value suggests that, in most cases, the cosine similarity scores obtained with Word2Vec are larger than those obtained with BERT.
* Word2Vec – TF-IDF+BoW (0.9998) This value is very close to 1, indicating a very large effect size. It suggests a substantial difference in the distributions of cosine similarity scores between Word2Vec and TF-IDF. The positive value signifies that Word2Vec scores are, on average, substantially higher than those of TF-IDF. Given the magnitude of this effect, it can be interpreted that in almost all comparisons, Word2Vec yields higher cosine similarity scores than TF-IDF.

By conducting Kruskal-Wallis hypothesis test without incorporating human evaluation, it can be determined that the cosine scores generated through various embedding techniques differ, despite showing some level of correlation. Moreover, upon applying Cliff's Delta test, it was discovered that a substantial discrepancy exists in cosine calculation score between Word2Vec and TF-IDF+BoW, with a moderate difference observed between BERT and Word2Vec models.

### 4.2.3 Experiment Conclusion

In conclusion, examining the distribution and conducting statistical tests reveal that BERT encoding, through cosine scores, can discern the similarities and distinctions among tests within JRS, producing both positive and negative scores. Moreover, while cosine scores from various embedding techniques exhibit correlations ranging from moderate to high, the varied results they generate imply that choosing an embedding method should depend on the specific needs and objectives.

## 4.3 Third Experiment Results (Human Evaluation)

This section delves into the analysis of rankings based on cosine similarity scores obtained from three different text representation techniques, aiming to fulfill and discuss the experiment's objective. Differing from previous analyses, this experiment incorporates human evaluation to gauge the efficacy of the various text representation methods. It begins with an examination of the ranking outcomes, followed by the results from human evaluation, and concludes with the insights gathered from these two assessments.

4.3.1 Evaluation of the rankings

From the information provided in previous sections, it is noted that three participants voluntarily joined the experiment. These participants consented to share their private information and assist in conducting the experiment focused on assessing person-job matching experience. Furthemore, these volunteers agreed to evaluate the recommendations through a ranking process and this subsection will examine the results from both the model and human evaluation for each participant.

*A black and white screen with numbers

Description automatically generatedUser1 = registered nurse*: The logic for the first incorrect label occurrence and the total number of incorrect label occurrences were used to determine how well the cosine ranking order worked for each method. As can be seen from the **Figure** , the Bert model was able to precisely rank the correct job ads above all incorrect job ads, with a correct ranking percentage of 100. For BERT, the first incorrect occurrence (an ad labelled differently from "registered\_nurse") appears at index 644, which implies that all the ads ranked before this index are correctly labelled as 'registered\_nurse', showing that BERT's embeddings for similarity measures are highly effective for this particular ranking task. Word2Vec's performance shows the first incorrectly labelled ad appearing much earlier in the list, at index 535. This indicates that while most of the top-ranked ads are correctly labelled, there's a slight drop in precision compared to BERT. The correct ranking percentage is 96%, reflecting this high but not perfect accuracy. It suggests that Word2Vec is effective but slightly less precise than BERT for ensuring that the most relevant ads appear at the top of the ranking. TFIDF + BOW shows the earliest occurrence of an incorrectly labelled ad, at index 433. This method has the lowest correct ranking percentage at 89%, indicating that it is less effective than both BERT and Word2Vec for this specific ranking task.

*User2 = electrician*: The results for user2 (**refer to figure**) indicates that the BERT model again outperforms the other techniques in ranking job ads correctly, with a nearly perfect correct ranking percentage of 99.315068%. The first instance of an incorrect job ad labelling using BERT is observed at index 144, suggesting that all preceding ads were ac curately identified. In contrast, the Word2Vec model exhibits its first incorrect label at a significantly A black and white numbers

Description automatically generatedearlier index, 71, indicating a decrease in accuracy when compared to the previous result with user1. Despite this, it still achieves a correct ranking percentage of 82.876712%. The TF-IDF + BoW method demonstrates a slightly bad performance than Word2Vec in terms of the index of the first incorrect occurrence, which stands at 66. However, it has a higher correct ranking percentage of 89.726027%.

A black and white numbers

Description automatically generated with medium confidence*User3 = data analyst:* For user3 (refer to figure) ranking BERT embedding again demonstrates superior performance, with the first incorrect occurrence index at 373, indicating that a large number of job ads were ranked correctly before encountering the first misclassification. This model also has the highest correct ranking percentage at 99.734043%. Contrary to previous results, Word2Vec had an incorrect first occurrence much earlier in the ranking at index 103, and the correct ranking percentage for Word2Vec went down to 77.393617%, the lowest among the three models, indicating a considerable drop in precision compared to the previous two results. Lastly, the first incorrect occurrence for TF-IDF + BoW is observed at the earliest index among the three, at 55. However, its correct ranking percentage stands at 86.702128%, which is higher than that of Word2Vec again.

Overall, the BERT model consistently outperformed Word2Vec and TF-IDF + BoW in ranking order of the job ads accurately across all evaluations for each user, with correct ranking order percentages exceeding 99%. Its superiority highlights BERT's effectiveness in semantic similarity tasks, demonstrating a significant advantage in precision. Word2Vec showed variability and a decrease in accuracy across labels, indicating less consistency in top-ranking relevancy. TF-IDF + BoW, while competitive, also fell short of BERT's performance, especially in early incorrect classifications. Overall, BERT stands out as the most reliable model for precise ranking tasks, underscoring the importance of choosing advanced embedding techniques for tasks requiring high semantic accuracy.

### 4.3.2 Human evaluation

Simply examining the outcomes of model evaluations does not provide sufficient insight to determine the most effective model for this kind of task. It's necessary to incorporate human judgement into the analysis to understand how these rankings reflect practical scenarios. Hence, to draw a definitive conclusion about which embedding technique performs best in real-world applications, human evaluation of the rankings is essential. Previously, we computed cosine scores for each model and assigned rankings from 1 to 3. Through stratified sampling, human evaluation samples were carefully selected from the primary dataset. For every model, 30 job ads were randomly chosen, ensuring an equal distribution across each ranking level. These selected shuffled sets were then forwarded to experiment participants without disclosing the ranking order, providing only job links and a space for rankings. Participants were instructed to visit the job listing websites, review the job details, and assign rankings on a scale of 1 to 3, adhering to a specified ranking logic:

* Rank 1: The job advertisement is not suitable and does not match my desired field. (Negative)
* Rank 2: The field of the job advertisement corresponds to my interests, but some aspects do not fully fit my professional background or may not entirely meet the company's expectations. (False Positive)
* Rank 3: The job advertisement is in line with my interests, and all the requirements and qualifications match what I am seeking. (True Positive)

A screenshot of a graph

Description automatically generated*User1 = registered nurse*: After a certain period, the rankings provided by humans were received back. And from the bar charts and tables detailing the evaluation of user1 (**refert to**), it's evident that the BERT model achieved the most matches, totaling 23 out of 30 rankings. In contrast, the TF-IDF + BoW method had the lowest matches, with 16 out of 30. Word2Vec performed marginally better than TF-IDF + BoW, securing 18 matches out of a total of 30. Both the BERT and Word2Vec models aligned with the participant's rankings on the job ad’s ranking 1. Moreover, the inconsistencies for jobs ranked as 2 and 3 in relation to user evaluations were computed. The findings reveal that job advertisements given a model ranking of 3 were never rated as rank 1 by the user across all models, which is good sign. This indicates that job ads deemed highly suitable by the models were not seen as unsuitable by the user. However, there were discrepancies observed with job ads assigned a model ranking of 2. Specifically, 7 job ads were identified as mismatches for TF-IDF, and 4 job ads ranked by Word2Vec were also seen as mismatches.

A screenshot of a graph

Description automatically generated*User2 = electrician*: For user2 the BERT encoding demonstrates the strongest correlation with human rankings (refer to figure), particularly for the least suitable job ads (rank 1). Word2Vec performs well at rank 1 but shows significant discrepancies at rank 2. TF-IDF + BoW, while having the most matches at rank 3, shows a moderate level of mismatch across all ranks. The data indicates that while all models have their strengths, BERT seems to be the most aligned with human judgment, particularly for low-priority recommendations. Furtherrmore, the inconsistencies for jobs ranked as 2 and 3 in relation to user evaluations were computed for user 2. The findings reveal that job advertisements given a model ranking of 3 were never rated as rank 1 by the user across all models again, which is a good sign. This indicates that job ads deemed highly suitable by the models were not seen as unsuitable by the user. However, there were discrepancies observed with job ads assigned a model ranking of 2. Specifically, 7 job ads were identified as mismatches for Wod2Vec this time, and 2 job ads ranked by TF-IDF + BoW were also seen as mismatches.

A screenshot of a graph

Description automatically generated*User3 = data analyst:* BERT again consistently aligns closely with human rankings, especially for the least suitable job ads (rank 1), and shows few mismatches across all ranks. Word2Vec, while having issues at rank 2, aligns perfectly with human rankings at rank 3, indicating its strength in identifying the most suitable job ads. TF-IDF + BoW demonstrates moderate performance in both matches and mismatches, with a reasonably good alignment in rank 1 matches but more mismatches at rank 2. Each model shows particular strengths at different ranking levels, with BERT being the most reliable overall and Word2Vec distinguishing itself at rank 3. For user3, discrepancies were calculated for job ads that the models ranked as 2 and 3, in comparison to the user's evaluations. The outcomes indicated that jobs ranked as 3 by the models were consistently not rated as 1 by the user, demonstrating alignment at this level of ranking. However, there were again inconsistencies with job ads that the models ranked as 2. In this instance, Word2Vec showed the most significant number of discrepancies with 8, and TF-IDF + BoW followed with 4, indicating a divergence in model and human evaluations for moderately suitable job ads.

It's apparent that human ranking do not always align with those made by models, though certain models exhibit fewer discrepancies with human rankings than others. Specifically, across all cases studied, the BERT model consistently showed the most accurate performance for the lowest-ranked jobs. However, for advertisements with high and middle rankings, it didn't surpass the other two methods. As it can be seen from the calculations above the jobs ranked as 3 by the models were consistently not ranked as 1 by users in all three instances but the jobs with rank 2 have considerable mismatches with human evaluation. This suggests that for a content-based job recommendation system, solely relying on positive rankings (rank 2, rank 3) for making recommendations might not be the best strategy, as even the BERT model can sometimes deviate from human evaluations. Therefore, the optimal approach would be to choose the highest ranked job ads from the list of recommendations, and all three models for this purpose can be accepted. The second key point is that jobs given a ranking of 1 by the models are not suitable for recommendations to users, as this rank indicates the lowest compatibility or highest difference, not only according to the models' evaluations but also reflecting a uniform accuracy across all users. Furthermore, jobs ranked as 1 by the models were never considered as rank 2 or 3 by human evaluators.

Observations indicated unresolved inquiries concerning human evaluation within the dataset, highlighting the necessity for testing to uncover these answers. This led to the necessity of conducting comparisons for statistical significance between models and human evaluations. Following this, Spearman's Rank Correlation and the Wilcoxon Signed-Rank test were applied for each user’s evaluated dataset. And the results showed that that the BERT model consistently aligned closely with human evaluation across all instances, demonstrating highly correlated results with a very low p-value. Similarly, for the Word2Vec model, no statistically significant differences were detected compared to human evaluation. However, for user3, the correlation exhibited moderate results, with a rho score of 0.69, suggesting a detectable but not substantial discrepancy. Regarding the TF-IDF + BoW model, statistical differences were detected for user 1, with a moderate correlation observed. Since the sample size for the job seekers population in this instance comprises only three users, it is not appropriate to extrapolate conclusions to the entire population. Further analysis with a larger number of experiment participants is necessary. The only definitive conclusion here is that model results can vary under different conditions and with different users.

### 4.3.3 Experiment Conclusion:

Based on the experiment results, the BERT model emerges as the most reliable embedding technique for a job recommendation system, closely mirroring human evaluations with significant statistical backing. The Word2Vec model, while not showing statistically significant differences, aligns well with human judgment, indicating its potential effectiveness. However, the moderate correlation for one user suggests the model's performance can vary among individuals. The TF-IDF + BoW model's inconsistent performance, highlighted by statistical differences for a user, underscores the variability in model effectiveness across different users.

# REFERENCES

Bansal, S., Srivastava, A., and Arora, A. (2017). Topic modeling driven content-based jobs recommendation engine for recruitment industry. Procedia computer science, 122, (pp. 865-872).

Guo, X., Jerbi, H., & O'Mahony, M. P. (2014). An analysis framework for content-based job recommendation. In 22nd International Conference on Case-Based Reasoning (ICCBR), Cork, Ireland, 29 September-01 October 2014.

Brahushi, G., & Ahmad, U. (2022). Empirical Evaluation of Word Representation Methods in the Context of Candidate-Job Recommender Systems. In 2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI) (pp. 183-187). IEEE.

Gugnani, A., & Misra, H. (2020). Implicit skills extraction using document embedding and its use in job recommendation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 08, pp. 13286-13293).

Kurdija, A. S., Afric, P., Sikic, L., Plejic, B., Silic, M., Delac, G., ... & Srbljic, S. (2020). Building vector representations for candidates and projects in a CV recommender system. In Artificial Intelligence and Mobile Services–AIMS 2020: 9th International Conference, Held as Part of the Services Conference Federation, SCF 2020, Honolulu, HI, USA, September 18-20, 2020, Proceedings 9 (pp. 17-29). Springer International Publishing.

Salinas, A., Shah, P., Huang, Y., McCormack, R., & Morstatter, F. (2023). The Unequal Opportunities of Large Language Models: Examining Demographic Biases in Job Recommendations by ChatGPT and LLaMa. In Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (pp. 1-15).

Nguyen, T., Vu, N., & Ly, B. (2022). An approach to constructing a graph data repository for course recommendation based on IT career goals in the context of big data. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 301-308). IEEE.

Panchasara, S., Gupta, R. K., & Sharma, A. (2023). AI Based Job Recommendation System using BERT. In 2023 7th International Conference on Computing, Communication, Control And Automation (ICCUBEA) (pp. 1-6). IEEE.

Cheng, J., Wang, Z., Wen, J. R., Yan, J., & Chen, Z. (2015). Contextual text understanding in distributional semantic space. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management (pp. 133-142).

Liu, L., & Özsu, M. T. (Eds.). (2009). Encyclopedia of database systems (Vol. 6). New York, NY, USA:: Springer.

Gao, M., Fu, Y., Chen, Y., & Jiang, F. (2012). User-Weight Model for Item-based Recommendation Systems. J. Softw., 7(9), 2133-2140.

Al-Otaibi, S. T., & Ykhlef, M. (2012). A survey of job recommender systems. International Journal of the Physical Sciences, 7(29), 5127-5142.

Vega, J. (1990). Semantic matching between job offers and job search requests. In COLING 1990 Volume 1: Papers presented to the 13th International Conference on Computational Linguistics.

Abel, F., Benczúr, A., Kohlsdorf, D., Larson, M. and Pálovics, R (2016) RecSys challenge 2016: Job recommendations. In Proceedings of the 10th ACM conference on recommender systems (pp. 425-426).

Ahmed, S., Hasan, M., Hoq, M.N. and Adnan, M.A. (2016) User interaction analysis to recommend suitable jobs in career-oriented social networking sites. In 2016 International Conference on Data and Software Engineering (ICoDSE) (pp. 1-6). IEEE.

Allmark, P., Boote, J., Chambers, E., Clarke, A., McDonnell, A., Thompson, A. and Tod, A.M. (2009) Ethical issues in the use of in-depth interviews: literature review and discussion. Research Ethics, 5(2), pp.48-54.

Amaar, A., Aljedaani, W., Rustam, F., Ullah, S., Rupapara, V. and Ludi, S. (2022) Detection of fake job postings by utilizing machine learning and natural language processing approaches. Neural Processing Letters, pp.1-29.

Antoun, C., Zhang, C., Conrad, F.G. and Schober, M.F. (2016) Comparisons of online recruitment strategies for convenience samples: Craigslist, Google AdWords, Facebook, and Amazon Mechanical Turk. Field methods, 28(3), pp.231-246.

Bailey, K. (2008) Methods of social research. Simon and Schuster.

Bartlett, M., Morreale, F. and Prabhakar, G. (2023) Analysing Privacy Policies and Terms of Use to understand algorithmic recommendations: the case studies of Tinder and Spotify, Journal of the Royal Society of New Zealand, 53(1), pp. 119-132.

Beel, J., Gipp, B., Langer, S. and Breitinger, C. (2016) Paper recommender systems: a literature survey. International Journal on Digital Libraries, 17, pp.305-338.

Bortnick, S.M. and Ports, M.H. (1992) Job search methods and results: Tracking the unemployed, 1991. Monthly Lab. Rev., 115, p.29.

Castells, P. and Moffat, A. (2022) Offline recommender system evaluation: Challenges and new directions. AI Magazine, 43(2), pp.225-238.

Davidson, J., Liebald, B., Liu, J., Nandy, P., Van Vleet, T., Gargi, U., Gupta, S., He, Y., Lambert, M., Livingston, B. and Sampath, D. (2010) The YouTube video recommendation system, Proceedings of the fourth ACM conference on Recommender systems, pp. 293-296.

De Ruijt, C. and Bhulai, S. (2021) Job recommender systems: A review. arXiv preprint arXiv:2111.13576.

Dong, Z., Wang, Z., Xu, J., Tang, R., and Wen, J. (2022) A Brief History of Recommender Systems, ArXiv, abs/2209.01860.

Etikan, I., Musa, S.A. and Alkassim, R.S. (2016) Comparison of convenience sampling and purposive sampling. American journal of theoretical and applied statistics, 5(1), pp.1-4.

Färber, F., Weitzel, T. and Keim, T. (2003) An automated recommendation approach to selection in personnel recruitment.

Gilotte, A., Calauzènes, C., Nedelec, T., Abraham, A. and Dollé, S. (2018) Offline a/b testing for recommender systems. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (pp. 198-206).

Gomez-Uribe, Carlos, A. and Hunt, N. (2016) The Netflix Recommender System: Algorithms, Business Value, and Innovation, ACM Transaction on Management Information Systems (TMIS), 6(4), pp. 1-19.

Google (2023), Advanced Machine Learning Course. Available at: https://developers.google.com/machine-learning/recommendation/content-based/summary (Accessed 30 October 2023)

Gugnani, A. and Misra, H. (2020) Implicit skills extraction using document embedding and its use in job recommendation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 08, pp. 13286-13293).

Guo, C., Lu, H., Shi, S., Hao, B., Liu, B., Zhang, M., Liu, Y. and Ma, S. (2017) How integration helps on cold-start recommendations. In Proceedings of the Recommender Systems Challenge 2017 (pp. 1-6).

Guo, G., Zhang, J., Sun, Z. and Yorke-Smith, N. (2015) Librec: A java library for recommender systems, UMAP workshops, Vol. 4, pp. 38-45.

Hardin, J., Hoerl, R., Horton, N.J., Nolan, D., Baumer, B., Hall-Holt, O., Murrell, P., Peng, R., Roback, P., Temple Lang, D. and Ward, M.D. (2015) Data science in statistics curricula: Preparing students to “think with data”. The American Statistician, 69(4), pp.343-353.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.S. (2017) Neural collaborative filtering, Proceedings of the 26th international conference on world wide web, pp.

Indeed (2023) Indeed General Terms of Service. Available at: https://www.indeed.com/legal (Accessed 30 October 2023)

Isinkaye, F.O., Folajimi, Y.O. and Ojokoh, B.A. (2015) Recommendation systems: Principles, methods, and evaluation. Egyptian informatics journal, 16(3), pp.261-273.

Janusz, A., Stawicki, S., Drewniak, M., Ciebiera, K., Ślęzak, D. and Stencel, K. (2018) How to match jobs and candidates-a recruitment support system based on feature engineering and advanced analytics. In Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations: 17th International Conference, IPMU 2018, Cádiz, Spain, June 11-15, 2018, Proceedings, Part II 17 (pp. 503-514). Springer International Publishing.

Kara, A., Daniş, F.S., Orman, G.K., Turhan, S.N. and Özlü, Ö.A. (2022) Job Recommendation Based on Extracted Skill Embeddings. In Proceedings of SAI Intelligent Systems Conference (pp. 497-507). Cham: Springer International Publishing.

Karypis, G. (2001) Evaluation of item-based top-N recommendation algorithms, Proceedings of the tenth international conference on information and knowledge management - CIKM’01. New York, New York, USA: ACM Press.

Kenthapadi, K., Le, B. and Venkataraman, G. (2017) Personalized job recommendation system at LinkedIn: Practical challenges and lessons learned, In Proceedings of the eleventh ACM conference on recommender systems, pp. 346-347.

Kerlinger, F.N. (1966) Foundations of behavioral research.

Konstan, J.A. and Riedl, J (2012) Recommender systems: from algorithms to user experience, User modelling and user-adapted interaction, 22(1–2), pp. 101–123.

Koren, Y. (2008) Factorization meets the neighbourhood: a multifaceted collaborative filtering model’, Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 426-434.

Krotov, V. and Silva, L. (2018) Legality and ethics of web scraping.

Lacic, E., Reiter-Haas, M., Kowald, D., Reddy Dareddy, M., Cho, J. and Lex, E. (2020) Using autoencoders for session-based job recommendations. User Modeling and User-Adapted Interaction, 30, pp.617-658.

Li Y., Ning G., and Zhang H. (2021) Research on Position Recommendation System Based on Convolutional Neural Network. Journal of Physics: Conference Series, Volume 2171, Issue 1, id.012065, 7 pp.

Liu, K., Shi, X., Kumar, A., Zhu, L. and Natarajan, P. (2016) Temporal learning and sequence modeling for a job recommender system. In Proceedings of the Recommender Systems Challenge (pp. 1-4).

Mashayekhi, Y., Li, N., Kang, B., Lijffijt, J. and De Bie, T. (2022) A challenge-based survey of e-recruitment recommendation systems. arXiv preprint arXiv:2209.05112.

Mauro, N., Ardissono, L., Petrone, G., Geninatti Cossatin, A. and Mattutino, C. (2021) Beyond traditional cultural heritage recommender systems: Suggesting Airbnb experiences to users, In Adjunct Proceedings of the 29th ACM Conference on User Modelling, Adaptation and Personalization, pp. 203-207.

Millecamp, M., Htun, N.N., Jin, Y. and Verbert, K. (2018) Controlling Spotify recommendations: effects of personal characteristics on music recommender user interfaces, Proceedings of the 26th Conference on user modelling, adaptation and personalization, pp. 101-109.

Mpela, M.D. and Zuva, T. (2020) A mobile proximity job employment recommender system. In 2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD) (pp. 1-6). IEEE.

Ni, Q. (2022) Deep Neural Network Model Construction for Digital Human Resource Management with Human-Job Matching. Computational Intelligence and Neuroscience.

Pan, C. and Li, W. (2010) Research paper recommendation with topic analysis, International Conference on Computer design and applications, Qinhuangdao, China, pp. 264-268.

Peska, L. and Vojtas, P. (2020) Off-line vs. On-line Evaluation of Recommender Systems in Small E-commerce. In Proceedings of the 31st ACM Conference on Hypertext and Social Media (pp. 291-300).

Resnick, D.B. (2020) What Is Ethics in Research & Why Is It Important? Available at: https://www.niehs.nih.gov/research/resources/bioethics/whatis/index.cfm (Accessed 29 October 2023)

Resnick, J.H. and Schwartz, T. (1973) Ethical standards as an independent variable in psychological research. American Psychologist, 28(2), p.134.

Reusens, M., Lemahieu, W., Baesens, B. and Sels, L. (2017) A note on explicit versus implicit information for job recommendation. Decision Support Systems, 98, pp.26-35.

Roy, D. and Dutta, M. (2022) A systematic review and research perspective on recommender systems. Journal of Big Data, 9(1), p.59.).

Sarwar, B., Karypis, G., Konstan, J. and Riedl, J.T. (2000) Application of dimensionality reduction in recommender system-a case study, Retrieved from the University of Minnesota Digital Conservancy.

Schinke, S.P. and Gilchrist, L. (1993) Ethics in research. Social Work, Research and Evaluation, pp.80-92.

Schmitt, T., Caillou, P. and Sebag, M., 2016, September. Matching jobs and resumes: a deep collaborative filtering task. In GCAI 2016-2nd Global Conference on Artificial Intelligence (Vol. 41).

Shani, G. and Gunawardana, A. (2011) Evaluating recommendation systems. Recommender systems handbook, pp.257-297.

Shapira, B., Rokach, L. and Freilikhman, S. (2013) Facebook single and cross domain data for recommendation systems, User Modelling and User-Adapted Interaction, 23, pp. 211-247.

Short, T. (2014) Research methodology: A step-by step guide for beginners (4th edition) [Book Review]. International Journal of Training Research, 12, 158.

Stevenson, C.L. (1944) Ethics and language.

Stratton, S.J. (2021) Population research: convenience sampling strategies. Prehospital and disaster Medicine, 36(4), pp.373-374.

Su, X. and Khoshgoftaar, T.M. (2009) A survey of collaborative filtering techniques. Advances in artificial intelligence.

Tran, M.L., Nguyen, A.T., Nguyen, Q.D. and Huynh, T. (2017) A comparison study for job recommendation. In 2017 International Conference on Information and Communications (ICIC) (pp. 199-204). IEEE.

Valverde-Rebaza, J.C., Puma, R., Bustios, P. and Silva, N.C. (2018) Job Recommendation Based on Job Seeker Skills: An Empirical Study. In Text2Story@ ECIR (pp. 47-51).

Vanetik, N., Kolesnev, A. and Kogan, G. (2023) NLP-based Screening for IT Job Vacancies: A Case Study.

Volkovs, M., Yu, G.W. and Poutanen, T. (2017) Content-based neighbor models for cold start in recommender systems. In Proceedings of the Recommender Systems Challenge 2017 (pp. 1-6).

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L.D.A., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J. and Kuhn, M. (2019) Welcome to the Tidyverse. Journal of open-source software, 4(43), p.1686.

Wu, S., Sun, F., Zhang, W., Xie, X. and Cui, B. (2022) Graph neural networks in recommender systems: a survey. ACM Computing Surveys, 55(5), pp.1-37.

Yang, C., Hou, Y., Song, Y., Zhang, T., Wen, J.R. and Zhao, W.X. (2022) Modeling two-way selection preference for person-job fit. In Proceedings of the 16th ACM Conference on Recommender Systems (pp. 102-112).

Yin, R.K. (2009) Case study research: Design and methods (Vol. 5). sage.

Yu, H., Liu, C. and Zhang, F. (2011) Reciprocal recommendation algorithm for the field of recruitment. Journal of Information & Computational Science, 8(16), pp.4061-4068.

Zheng, Z., Qiu, Z., Hu, X., Wu, L., Zhu, H. and Xiong, H. (2023) Generative job recommendations with large language model. arXiv preprint arXiv:2307.02157.

Bulgan, T 2024 Participate in my Research, Google Forms, viewed 20 February 2024, https://docs.google.com/forms/d/e/1FAIpQLSffzk7Kjof2CnAKvWQKS0cM-Cvwo8578HFVo9xYi7saHIlWUg/viewform?fbclid=IwAR3JEthDnDqmsfUZ\_iaAuFwxisjsJd6jlb85ZVBeNDF6VlubYCipnA1xU-0

Chapagain, A. (2019). Hands-On Web Scraping with Python: Perform advanced scraping operations using various Python libraries and tools such as Selenium, Regex, and others. Packt Publishing Ltd.

Apaza, H., Rubin de Celis Vidal, A. A., & Chire Saire, J. E. (2021). Job recommendation based on curriculum vitae using text mining. In Advances in Information and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 1 (pp. 1051-1059). Springer International Publishing.

Elsafty, A., Riedl, M., & Biemann, C. (2018). Document-based recommender system for job postings using dense representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers) (pp. 216-224).

Romadon, A. W., Lhaksmana, K. M., Kurniawan, I., & Richasdy, D. (2020, June). Analyzing TF-IDF and word embedding for implementing automation in job interview grading. In 2020 8th International Conference on Information and Communication Technology (ICoICT) (pp. 1-4). IEEE.

Lavi, D., Medentsiy, V., & Graus, D. (2021). consultantbert: Fine-tuned siamese sentence-bert for matching jobs and job seekers. arXiv preprint arXiv:2109.06501.

Neelima, A., & Mehrotra, S. (2023, February). A Comprehensive Review on Word Embedding Techniques. In 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS) (pp. 538-543). IEEE.

Saeed, A., Khan, H. U., Shankar, A., Imran, T., Khan, D., Kamran, M., & Khan, M. A. (2023). Topic Modeling based Text Classification Regarding Islamophobia using Word Embedding and Transformers Techniques. ACM Transactions on Asian and Low-Resource Language Information Processing.

Shrestha K. (2020) Comparative Analysis of TF-IDF and Word2vec Algorithm for Content-based Job Recommendation System.

Singh, P., Jain, B., & Sinha, K. (2023, July). Evaluating Bert and GPT-2 Models for Personalised Linkedin Post Recommendation. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.

Brahushi, G., & Ahmad, U. (2022). Empirical Evaluation of Word Representation Methods in the Context of Candidate-Job Recommender Systems. In 2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI) (pp. 183-187). IEEE.

Zhu, J., Patra, B. G., & Yaseen, A. (2021). Recommender system of scholarly papers using public datasets. AMIA summits on translational science proceedings, 2021, 672.

Zhao, F., Li, X., Gao, Y., Li, Y., Feng, Z., & Zhang, C. (2022). Multi-layer features ablation of BERT model and its application in stock trend prediction. Expert Systems with Applications, 207, 117958.