National College of Ireland

Optimizing Job Recommendation Systems with AI: A Deep Dive into BERT and GPT Models

MSc Research Project
Artificial Intelligence

Kiymet Elif Ari
Student ID: 23111488

School of Computing
National College of Ireland

Supervisor: Sheresh Zahoor

National College of Ireland
MSc Project Submission Sheet
School of Computing

Student Name:	Kiymet Elif Ari		
Student ID:	23111488		
Programme:	MSC Artificial Intelligence	**Year:**	2023 - 2024
Module:	Research Practicum		
Supervisor:	Sheresh Zahoor		
Submission Due Date:	12 August 2024		
Project Title:	Optimizing Job Recommendation Systems with AI: A Deep Dive into BERT and GPT Models		
Word Count:	5930	**Page Count:**	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Kiymet Elif Ari

Date: 06.09.2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Optimizing Job Recommendation Systems with AI: A Deep Dive into BERT and GPT Models

Kiymet Elif Ari
23111488

Abstract

The increasing volume of job postings on online platforms has created a need for efficient job recommendation systems that enhance user experience and streamline job searches. This research investigates the optimization of job recommendation systems using advanced natural language processing models, specifically BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer). I utilized a live dataset, which includes job descriptions, company details, and user interactions. I performed extensive data cleaning and preprocessing to normalize text data, encode categorical features, and engineer a Job Desirability Score. This score integrates normalized company ratings, review counts, and job type weights to quantify job attractiveness. BERT was utilized to generate semantic embeddings for job descriptions and capturing contextual nuances. GPT was utilized to model user preferences for personalized recommendations. The performance of both models was evaluated. BERT achieved a superior validation accuracy of 76.54% and an F1 score of 0.7532, and a balanced precision-recall ratio. In comparison, GPT achieved a validation accuracy of 70.37% and an F1 score of 0.6364, with greater fluctuations in validation loss. These results underscore the potential of BERT in providing more accurate and contextually relevant job recommendations. The results suggest that incorporating advanced NLP models like BERT significantly enhances job recommendation systems. Further improvements can be achieved through hyper-parameter tuning, larger datasets, and exploring other model architectures. This research provides a foundation for future developments in artificial intelligence driven job recommendation systems.

1. Introduction

This last decade, there was an exponential growth in the digital job market on online platforms. It transformed the way job seekers search for and apply to job opportunities. Growing volume of online job postings becomes a big challenging for job seekers to find positions that match their skills, preferences, and career goals. This challenge has created a demand for more advanced job recommendation systems that can effectively filter and rank job listings based on user-specific criteria. Traditional recommendation systems often work with keyword matching or other simple models. It may not be able to capture the full semantic meaning of job descriptions and user preferences. This limitation underscores the need for advanced techniques in Natural Language Processing (NLP) that can understand and analyze textual data better.

Recent research in NLP, particularly with the development of transformer-based models offer promising solutions to these challenges. Ability of BERT to understand context through bidirectional attention mechanisms and strength of GPT in generating coherent text based on user history make them ideal candidates for job recommendation systems. It is possible to create a recommendation engine that not only matches jobs to users based on explicit criteria but also considers the nuanced meanings within job descriptions and user interactions.

This research contributes to the scientific literature by demonstrating the application of advanced NLP models in the domain of job recommendation systems. Prior research has focused on traditional methods for job matching. This study explores the potential of BERT and GPT to provide more context-aware and personalized recommendations. The findings of this research offer insights into the advantages and limitations of using deep learning models for recommendation tasks and highlight the potential for these technologies to transform job search platforms.

The research question in this context is *How can artificial intelligence-driven job recommendation systems utilize advanced natural language processing techniques to better understand and match job descriptions with job seeker profiles?*

The research consists of 7 sections. Section 2 discusses related work and the evolution of job recommendation systems. It highlights gaps that this research aims to address. Section 3 discusses research methodology. It includes the data collection process, preprocessing steps, and the architecture of the models used in the research. Section 4 and Section 5 discuss the design and implementation of the job recommendation system. It includes the training and testing of BERT and GPT models. Section 6 discusses the evaluation metrics, compares the performance of the models, and the implications of the findings. Section 7 discusses the key contributions of the research, limitations, and suggests directions for future studies.

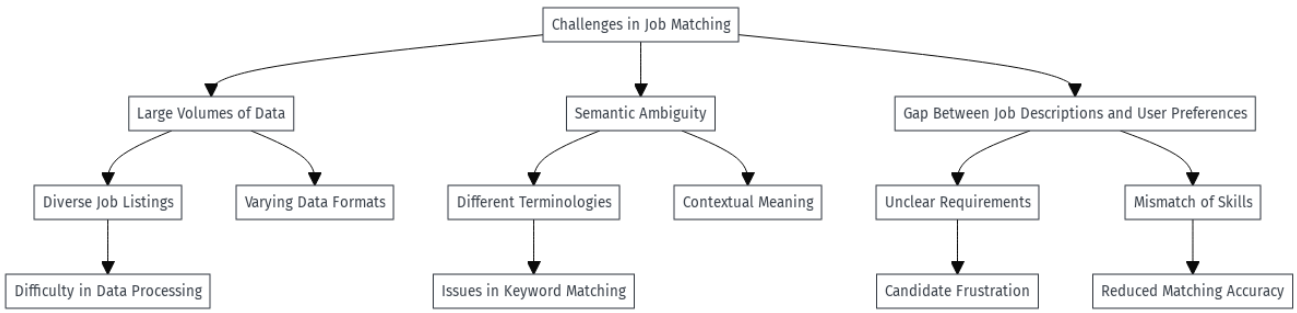


Figure 1: Challenges in Job Matching

2. Related Work

Evolution from the traditional methods of job recommendation systems, like keyword-based matching, has been toward AI-driven approaches that bring in the power of natural language processing. The integration of transformer-based models, such as Bidirectional Encoder Representations from Transformers and Generative Pre-trained Transformer, has progressed in the field. This research examines recent works in AI-driven job recommendation systems, more specifically toward the effectiveness and limitations of these models.

2.1. Semantic Analysis

The shift towards AI-driven job recommendation systems began with the need to understand job postings beyond simple keyword matching. Li, Yang et al. (2020) at LinkedIn proposed a deep transfer learning model to standardize and structure job data, significantly improving job matching accuracy. The inclusion of a feedback loop for continuous model improvement was a notable strength of this approach. However, the model's reliance on structured data extraction limited its ability to capture the full semantic richness of job descriptions, an area where more advanced models like BERT could provide greater depth by understanding contextual nuances.

Similarly, Domeniconi et al. (2016) utilized Latent Semantic Analysis (LSA) to cluster job positions based on the semantic similarity of LinkedIn users' skills. While effective in creating meaningful job groupings, ability of LSA to capture complex semantic relationships is limited when compared to transformer-based models like BERT, which excel at processing and understanding the intricate details of language, making them more suitable for nuanced job recommendation tasks.

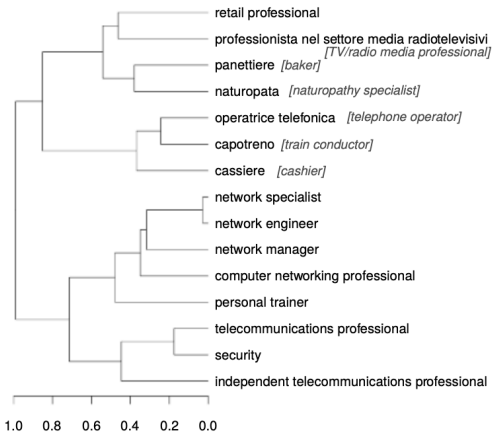


Figure 2: Cluster of Mixed Job Positions.

2.2. Personalized Recommendation Models

To address the growing demand for personalized job recommendations, Kenthapadi et al. (2017) implemented a model that combines collaborative filtering with machine learning techniques at LinkedIn. This approach led to significant improvements in application rates and user engagement. However, the model's dependency on collaborative filtering presents challenges, particularly the cold-start problem where recommendations for new users with limited interaction history are less accurate. Moreover, the model does not fully leverage the deeper semantic understanding that transformer-based models like BERT could offer, limiting its effectiveness in providing truly personalized recommendations.

Singla and Verma (2023) expanded on this by employing a hybrid approach that integrates collaborative filtering with content-based filtering. They enhanced this approach by natural language processing techniques such as cosine similarity and Word2Vec. Their system showed improved performance in job matching. However, the reliance on traditional NLP methods may limit the effort to fully capture the complex semantics of job descriptions and user profiles, suggesting that incorporating transformer-based models like BERT could raise accuracy and personalization.

2.3. Transformer-Based Approaches

Recent research has rather focused on applying transformer-based models in job recommendation systems. Singh, Jain et al. (2023) explored how the BERT and GPT-2 models could be applied for the recommendations of LinkedIn posts, which highlighted better contextual understanding and personalization ability for BERT. Although it highlights the potential of these models in recommendation tasks. It was primarily focused on post recommendations rather than job-specific recommendations. Additionally, the study did not fully explore the potential of these models for job-related tasks, leaving a gap in how these models could be optimized for job recommendations specifically.

In the domain of resume parsing and candidate matching, Bhatia et al. (2019) showcased the effectiveness of BERT in parsing resumes and ranking candidates. This work underscores the importance of contextual understanding in improving matching accuracy. However, the study was limited to resume parsing and did not extend to the broader job recommendation process, indicating a potential area for further exploration.

2.4. Deep Learning

Jain et al. (2023) explored the integration of deep learning and NLP in job recommendation systems, utilizing a hybrid filtering strategy that combines collaborative and content-based approaches. While the system demonstrated improved accuracy, it did not incorporate state-of-the-art models like BERT, which could further enhance the ability to understand and process job-related data. The limited use of advanced NLP techniques presents a gap that could be addressed in future research by incorporating transformer models to achieve more accurate and contextually relevant job recommendations.

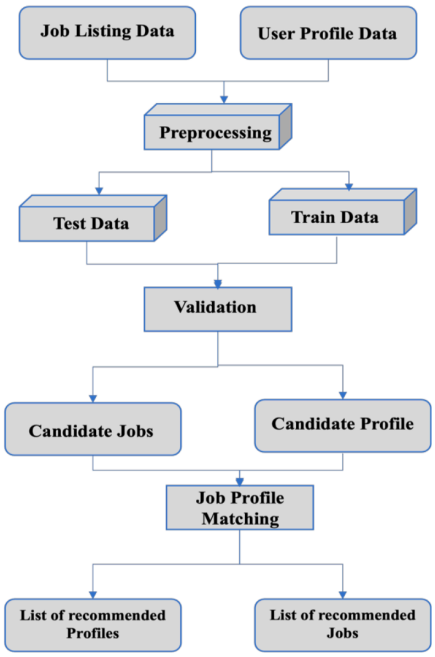


Figure 3: NLP-Based Bi-Directional Recommendation System.

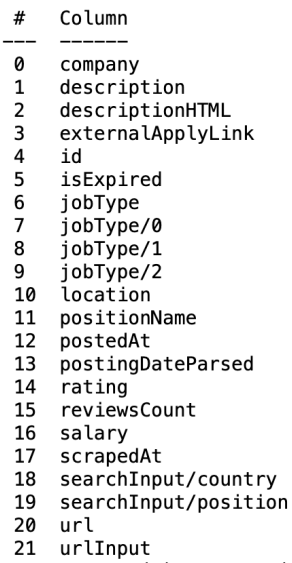
The reviewed literature shows major improvements in job recommendation systems, mainly by the adoption of transformer-based models. However, these approaches also highlight ongoing challenges, including scalability, real-time adaptability, and handling diverse datasets. Traditional models often lack the ability to capture semantic nuances, while modern models, though powerful, are computationally intensive and not always scalable. These limitations point to the need for further research into optimizing job recommendation systems, focusing on balancing accuracy, efficiency, and scalability. This gap justifies the research question of how to develop an AI-driven job recommendation system that leverages the strengths of transformer models while mitigating their limitations.

3. Research Methodology

This research aims to optimize job recommendation systems, using Natural Language Processing (NLP) models. I used BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer). The methodology has a systematic process from data collection to model evaluation. I ensured the reliability of the job recommendation system developed.

3.1. Data Collection

The data sources for this research was job postings scrapped from an online job platform three days in a row. The dataset includes very important attributes such as job descriptions, company names, job titles, locations, and company ratings for Software Engineering job area. Also, it includes additional metadata, such as the date the job was posted and the number of company reviews. It was crucial for the subsequent analysis. This dataset serves as the foundation for training and evaluating the job recommendation models. The data was stored in CSV format. Later I imported it into a Python environment for processing. The dataset was loaded using the Pandas library.



#	Column
0	company
1	description
2	descriptionHTML
3	externalApplyLink
4	id
5	isExpired
6	jobType
7	jobType/0
8	jobType/1
9	jobType/2
10	location
11	positionName
12	postedAt
13	postingDateParsed
14	rating
15	reviewsCount
16	salary
17	scrapedAt
18	searchInput/country
19	searchInput/position
20	url
21	urlInput

Figure 4: Scrapped Dataset Attributes

3.2. Data Preprocessing

Initial analysis of the data was examining the distribution of key variables such as job titles, locations, and job types. This research highlighted potential areas where data preprocessing would be necessary. It was such as handling missing values and standardizing text formats.

3.2.1. Column Removal

Data Cleaning is a critical step to ensure the quality and reliability of the dataset. Columns such as descriptionHTML, externalApplyLink, and urlInput were identified as irrelevant and removed to streamline the dataset.

3.2.2. Handling Missing Values

The salary column had missing values, which were filled with the placeholder 'Not Provided' to maintain the dataset's consistency.

3.2.3. Text Normalization

Textual data in columns such as company, description, location, and positionName were normalized by converting them to lowercase and stripping excess whitespace. This step was essential to ensure uniformity in text data, which is crucial for accurate text processing and analysis.

3.2.4. Standardization

The dataset contained multiple columns for job types. These were consolidated into a single jobType column to simplify the data structure and make it easier to use in subsequent analysis.

3.2.5. Date Time Conversion

The postingDateParsed and scrapedAt columns were converted into datetime objects to facilitate easier manipulation and analysis of date-related information.

3.3. Text Preprocessing and Feature Extraction

Text preprocessing and Feature Extraction involved several key steps to convert the job descriptions into formats suitable for input into NLP models.

3.3.1. Textual Feature Extraction with BERT

Text was tokenized using the BERT tokenizer, breaking down sentences into individual words or subwords.

3.3.2. Categorical and Numerical Feature Encoding

One-Hot Encoding was applied to the jobType column, Label Encoding to location, and standardization to rating and reviewsCount.

3.3.3. Stop Words Removal

Commonly used words that do not contribute significant meaning were removed from the dataset.

3.3.4. Job Desirability Score Calculation

A composite job desirability score was calculated based on normalized rating, reviewsCount, and job type. Higher ratings could imply a more desirable employer. More reviews can indicate a well-known and possibly desirable company, although this might need careful interpretation depending on context. Certain job types (like full-time or permanent positions) might be more desirable than others (like contract or part-time).

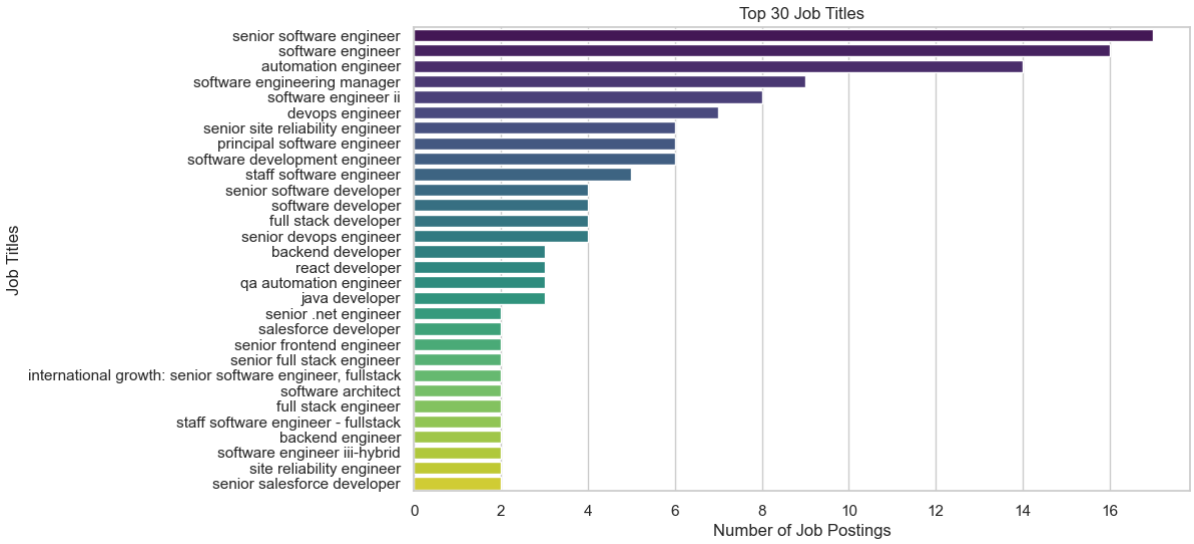
	norm_rating	norm_reviews	jobType_weight	job_desirability_score
0	0.289826	0.066472	1.0	0.342519
1	0.689299	-0.013547	1.0	0.470301
2	-1.219292	-0.140786	0.0	-0.544031
3	0.689299	-0.013547	1.0	0.470301
4	0.556142	-0.032403	0.0	0.209495

Figure 5: New Columns of First Five Line After Calculating Job Desirability Score

3.4. Exploratory Data Analysis

Exploratory Data Analysis was conducted to uncover patterns and relationships within the dataset. Visualizations such as bar charts, histograms, and word clouds were employed to understand the frequency of different job titles, the geographical distribution of job postings, and the common terms used in job descriptions.

The top 30 job titles were visualized to understand the distribution of job postings. Also, locations were normalized. The top 15 locations were visualized.



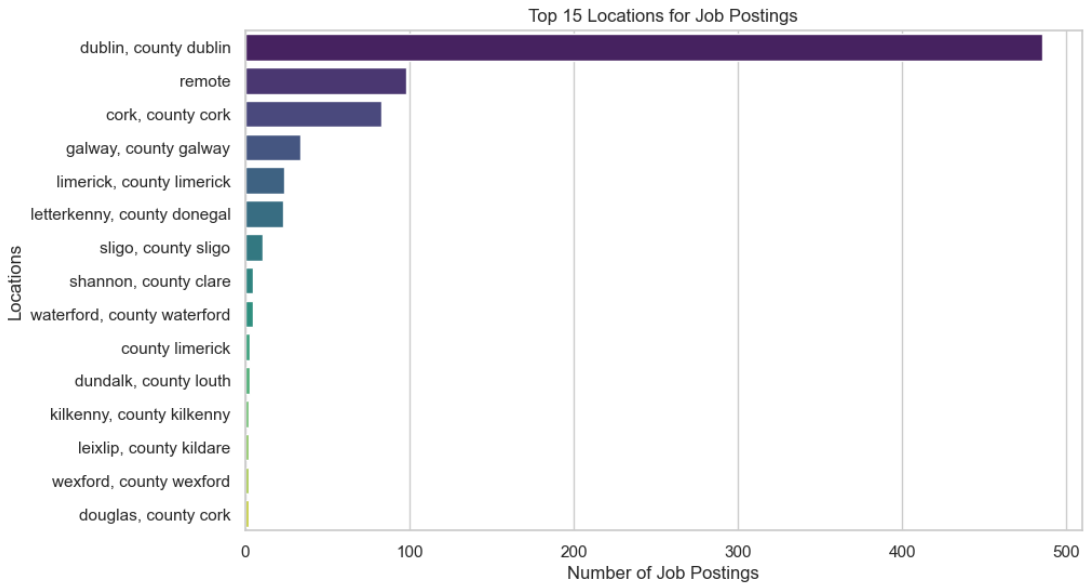


Figure 6: Top 30 Job Titles and Top 15 Locations Chart

A word cloud of the most common words in job descriptions were visualized. This analysis provided insights into the most frequently used terms in job postings. It can indicate the skills and qualifications employers prioritize.

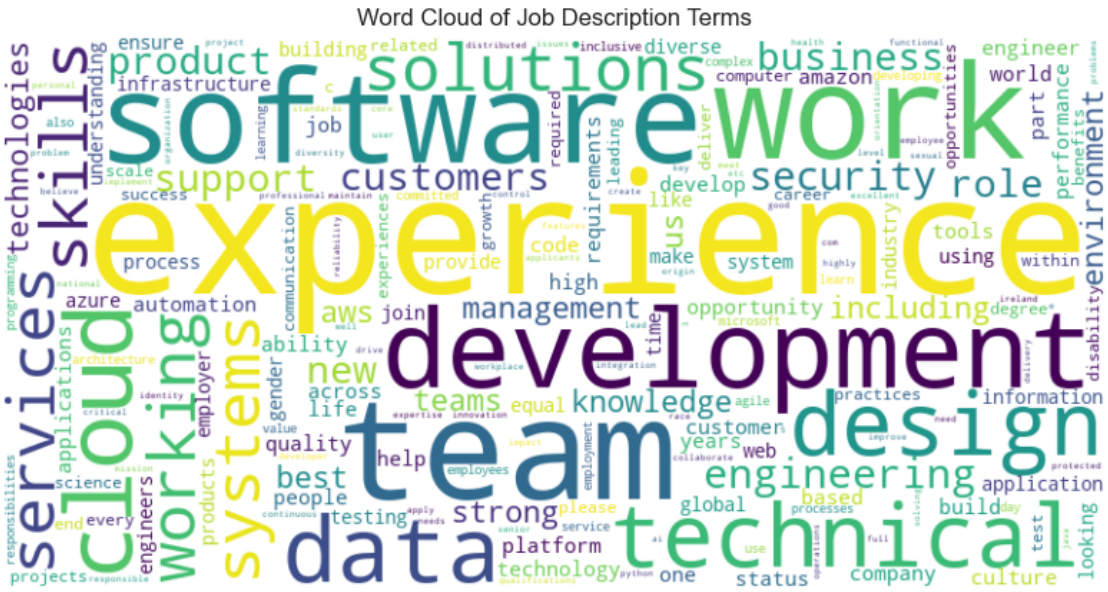


Figure 7: Word Cloud

A specific focus was understanding the relationship between job locations and their desirability scores. For investigating influencing of the location on job desirability, the job desirability scores were plotted across the most common job locations. The distribution of desirability scores varies across locations. Locations like Dublin, County Dublin and Cork, County Cork showing a higher median desirability score compared to others.

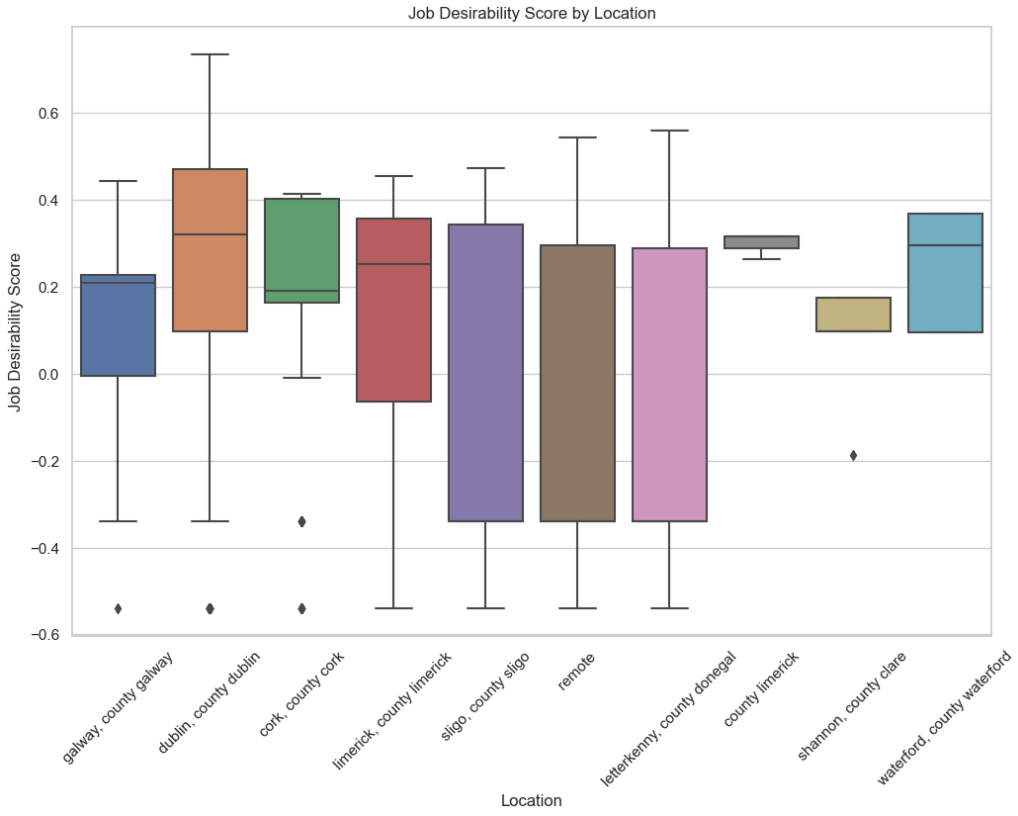


Figure 8: Job Desirability Score by Location Box Plot

3.5. Model Training with BERT

BERT (Bidirectional Encoder Representations from Transformers) was utilized for its effectiveness in handling text classification tasks, especially in the domain of Natural Language Processing (NLP). The BERT model was fine-tuned to classify job descriptions into binary labels indicating whether a job is desirable or not.

3.5.1. Text Preprocessing

Text data was normalized by converting it to lowercase, removing punctuation, and eliminating stop words. The BERT tokenizer was used to tokenize the text into a format suitable for the BERT model.

3.5.2. Model Initialization

The pre-trained bert-base-uncased model was loaded from Hugging Face Transformers, with a classification layer added to predict two possible classes (desirable or not).

3.5.3. Training Setup

The dataset was split into training and validation sets with a 90:10 ratio. The AdamW optimizer was used with a learning rate of 1e-5, with a linear learning rate scheduler and weight decay. Class weights were computed to handle class imbalance, ensuring the model did not become biased toward the majority class. The model was trained for 3 epochs with the validation set being used to monitor and prevent overfitting. (Trained for 3 epochs because it took too long to train for more than 3 epochs.)

3.5.4. Training and Validation

After each epoch, performance of the model was estimated using metrics such as accuracy, F1-score, precision, and recall. The model weights were saved after each epoch for ensuring that the best model was taken. The training process is contained forward propagation, loss calculation (with weighted loss to handle class imbalance), and back-propagation for gradient updates.

```
Epoch 1/3, Training Loss: 0.3233542507821387
Predictions: [0 0 0 1 0 0 0 1]
Actual Labels: [0 0 1 0 1 0 1 1]
Predictions: [1 1 0 0 1 0 0 1]
Actual Labels: [0 0 0 1 0 0 1 1]
Predictions: [1 0 1 1 0 0 1 1]
Actual Labels: [1 1 1 0 0 0 1 0]
Predictions: [1 1 1 0 1 0 1 0]
Actual Labels: [1 1 0 1 1 0 1 0]
Predictions: [1 1 0 0 0 1 0 0]
Actual Labels: [1 1 0 0 1 1 0 0]
Predictions: [0 0 0 0 0 1 0 0]
Actual Labels: [0 0 0 1 0 1 0 0]
Predictions: [0 1 1 1 1 1 1 0]
Actual Labels: [1 1 1 0 0 0 1 0]
Predictions: [1 0 0 1 0 1 1 1]
Actual Labels: [1 0 0 1 1 1 1 1]
Predictions: [0 0 1 0 1 0 0 0]
Actual Labels: [0 0 0 0 1 1 1 1]
Predictions: [1 0 1 1 0 1 1 0]
Actual Labels: [1 0 0 1 0 1 1 0]
Predictions: [0]
Actual Labels: [1]
Epoch 1/3, Validation Loss: 0.6415922289544885
Validation Accuracy: 0.6666666666666666, F1 Score: 0.6582278481012659, Precision: 0.6842105263157895, Recall: 0.634
1463414634146
```

Figure 9: First Epoch Results Example for BERT

3.5.5. Model Saving

After training, the BERT model was saved for future use. This involved saving the model weight and the tokenizer configuration. It ensures that the model could be reloaded and used for inference later.

3.6. Model Training with GPT

GPT (Generative Pre-trained Transformer) model, specifically GPT-2, was also trained on the same binary classification task. GPT-2 was chosen for its strong performance in text generation and understanding tasks.

3.6.1. Text Preprocessing

Similar to BERT, the text was preprocessed to remove stop words, lowercase the text, and remove punctuation. The GPT-2 tokenizer was employed to tokenize the text.

3.6.2. Model Initialization

The gpt2 model was initialized with a classification head, adjusted for binary output.

3.6.3. Training Setup

Training and validation sets were split in the same 90:10 ratio. The learning rate for GPT-2 was set to 1e-5, with a similar linear scheduler and weight decay applied. To handle class imbalance, class weights were calculated and applied during loss computation. The model was trained for 10 epochs, with early stopping based on validation loss to prevent overfitting.

3.6.4. Training and Validation

The training loop included evaluation of the model after each epoch, using accuracy, F1-score, precision, and recall as metrics. After each epoch, predictions of the model were compared to the actual labels to evaluate performance.

```
Epoch 1/10, Training Loss: 0.366196734892143
Predictions: [0 0 1 0 0 0 0 0]
Actual Labels: [0 0 1 0 1 0 1 1]
Predictions: [0 0 0 0 1 1 1 0]
Actual Labels: [0 0 0 1 0 0 1 1]
Predictions: [1 0 1 1 0 0 0 1]
Actual Labels: [1 1 1 0 0 0 1 0]
Predictions: [0 1 0 0 0 0 1 0]
Actual Labels: [1 1 0 1 1 0 1 0]
Predictions: [0 0 0 0 0 0 1 1]
Actual Labels: [1 1 0 0 1 1 0 0]
Predictions: [0 0 0 0 0 1 1 0]
Actual Labels: [0 0 0 1 0 1 0 0]
Predictions: [0 1 0 1 1 0 1 1]
Actual Labels: [1 1 1 0 0 0 1 0]
Predictions: [1 0 0 0 0 1 1 1]
Actual Labels: [1 0 0 1 1 1 1 1]
Predictions: [0 0 0 0 1 0 0 0]
Actual Labels: [0 0 0 0 1 1 1 1]
Predictions: [0 0 1 1 0 0 1 0]
Actual Labels: [1 0 0 1 0 1 1 0]
Predictions: [1]
Actual Labels: [1]
Epoch 1/10, Validation Loss: 0.6550091450864618
Validation Accuracy: 0.5679012345679012, F1 Score: 0.49275362318840576, Precision: 0.6071428571428571, Recall: 0.41
46341463414634
```

Figure 10: First Epoch Results Example for GPT

3.6.5. Early Stopping

Early stopping was implemented to prevent overfitting. If the validation loss did not improve for two consecutive epochs, training was halted, and the best model was saved.

3.6.6. Model Saving

After training, the GPT model was saved for future use. This involved saving the model weight and the tokenizer configuration. It ensures that the model could be reloaded and used for inference later.

3.7. Making Predictions on New Data

The models were trained and saved. After that, they were loaded back into the environment for making predictions on new data. For ease of use, prediction functions were defined to streamline the process of making predictions. To evaluate the models, predictions were made on the entire dataset. Also, key metrics were calculated.

This section outlines a comprehensive approach to developing an AI-driven job recommendation system. By leveraging advanced NLP models such as BERT and GPT, the system is designed to provide accurate and personalized job recommendations. The structured process from data collection through to deployment and continuous improvement ensures that the system is both robust and adaptable, capable of meeting the demands of real-world applications.

4. Design Specification

4.1. System Architecture

The architecture of the job recommendation system includes a layering of data collection, preprocessing, model training, and prediction modules. The architecture is modular, each model could independently developed, tested, and maintained. This modularity also enables scalability and flexibility so the system could be extended with additional features or models in the future. Data Collection Layer is responsible for gathering and storing job posting data from online platforms. This layer interfaces with web scraping tool to retrieve job-related information. Data Preprocessing Layer handles the cleaning, normalization, and feature extraction of raw data. This layer ensures that the data fed into the models is consistent and in the correct format. Model Training Layer encompasses the BERT and GPT models, which are fine-tuned on the preprocessed data to classify job postings based on their desirability. Prediction Layer utilizes the trained models to predict scores and generate job recommendations. Evaluation Loop monitors the performance of the models and updates them as necessary based on new data, ensuring continuous improvement.

4.2. Techniques and Frameworks

4.2.1. Natural Language Processing

The implementation helps several advanced techniques and frameworks to achieve high accuracy and efficiency in job recommendation. BERT is a transformer-based model used for text classification. Ability of BERT to understand the context of words in a sentence makes it highly effective for analyzing job descriptions and determining their relevance to job seekers. Another transformer-based model, GPT is utilized for its text generation and understanding capabilities.

4.2.2. Machine Learning Frameworks

Hugging Face Transformers is an open-source library provides pre-trained models like BERT and GPT, which were fine-tuned on the job posting dataset. The library is instrumental in simplifying the implementation of complex NLP tasks.

PyTorch is a deep learning framework used to implement and train the models. Flexibility of PyTorch and dynamic computation graph make it ideal for NLP tasks, where handling sequences of variable length is critical.

4.3. Requirements

The development of the job recommendation system requires adherence to specific functional and non-functional requirements.

Functional Requirements	Non-Functional Requirements
Data Integrity	Scalability
Model Accuracy	Maintainability
Batch Processing of Predictions	Efficiency

Figure 11: Table of Requirements

4.4. Algorithm

4.4.1. BERT-based Job Classification

The input to the BERT model consists of job descriptions. These descriptions are first preprocessed to remove unnecessary characters, normalize text including converting to lowercase and removing stop words and then tokenized using BERT's tokenizer.

The tokenized text is converted into embeddings using BERT's pre-trained model. The model is fine-tuned specifically for this binary classification task, where it processes the embeddings through multiple transformer layers to capture the contextual meaning of the text. The final layer of BERT outputs logits. These are passed through a classification head to predict two classes: "desirable" or "not desirable.”

The output is a binary label. 0 for "not desirable" and 1 for “desirable”. Label indicated predicted desirability of a job posting. This label is derived by applying a softmax function on the logits and choosing the class with the highest probability.

4.4.2. GPT-based Job Classification

Similar to BERT, the input to the GPT model is the job descriptions, which are preprocessed and tokenized using GPT's tokenizer.

The GPT model generates text embeddings from the tokenized job descriptions. Unlike BERT, GPT processes text sequentially, making it more adept at tasks that involve understanding and generating coherent text based on prior context. The GPT model is fine-tuned for classification, with the output layer adapted to predict one of two classes (desirable or not desirable). The logits produced by GPT are passed through a classification head to determine the final class.

The output is a binary label. It is similar to the BERT model's output.

4.4.3. Prediction Algorithm

The input consists of preprocessed job descriptions. These descriptions have been standardized and tokenized.

The pre-trained BERT and GPT models are loaded from saved directories and used to generate predictions. Each model processes the job descriptions, producing logits that are then converted into class labels. The predictions are generated in batch mode, where the entire dataset is processed at once. This approach is efficient for analyzing large volumes of job postings in a single pass.

The output is a list of binary predictions for the entire dataset. Each job posting is classified as either desirable or not desirable.

4.4.4. Evaluation Algorithm

The input is full dataset including the true labels of job desirability and the model predictions.

Evaluation metrics such as accuracy, F1 score, precision, and recall are calculated by comparing the predicted labels to the true labels in the dataset. These metrics are computed for both BERT and GPT models to assess their performance.

The output consists of the evaluation results, which include the performance metrics for each model. These metrics provide insights into the effectiveness of the models and help identify areas for potential improvement.

4.5. Innovation and Impact

Firstly, the dual-model approach applies both BERT and GPT models to derive better classification accuracy by combining their strengths. This dual approach enables the system to capture a broader range of linguistic variations and contextual information that leads to more accurate predictions.

Secondly, the integration of model predictions with other job-related features to generate a composite job desirability score represents a novel method for improving the relevance of recommendations. This composite scoring system goes beyond simple classification by considering multiple factors that influence job desirability, offering a more nuanced and personalized recommendation.

Lastly, the system is optimized for batch processing, allowing it to efficiently analyze large datasets, which ensures that the system can handle high volumes of data quickly and effectively. The expected impact of this research is substantial, as it offers a more accurate and personalized job recommendation experience, setting a new standard for job recommendation systems in batch processing scenarios through the innovative application of advanced NLP techniques and a robust system architecture.

5. Implementation

The final stage of the implementation focused on developing and deploying the job recommendation system by leveraging advanced Natural Language Processing (NLP) models. The key outputs of this implementation phase included transformed datasets, trained models, and the necessary scripts for prediction and evaluation.

The transformed data was processed through a rigorous pipeline that included data cleaning, normalization, and feature extraction. This was essential to prepare the job descriptions and other related attributes for modeling. The job dataset was systematically cleaned to remove irrelevant columns, fill in missing values, and standardize text formats, ensuring a robust and uniform dataset for model training.

For the models, BERT and GPT architectures were employed. Both models were fine-tuned for binary classification tasks, with BERT handling the tokenization and feature extraction, and GPT being tailored for text classification. These models were trained using Python, primarily relying on libraries such as PyTorch and Hugging Face's Transformers for model development and training. The trained models were then saved for future use, ensuring that they could be loaded and applied for batch processing tasks efficiently. Additionally, the prediction scripts were developed to integrate seamlessly with the saved models, providing a streamlined process for generating job desirability scores based on new data.

Overall, this implementation produced a scalable and efficient job recommendation system, leveraging state-of-the-art NLP models, ensuring it is well-equipped to handle large-scale data processing and provide accurate job recommendations in a batch processing environment.

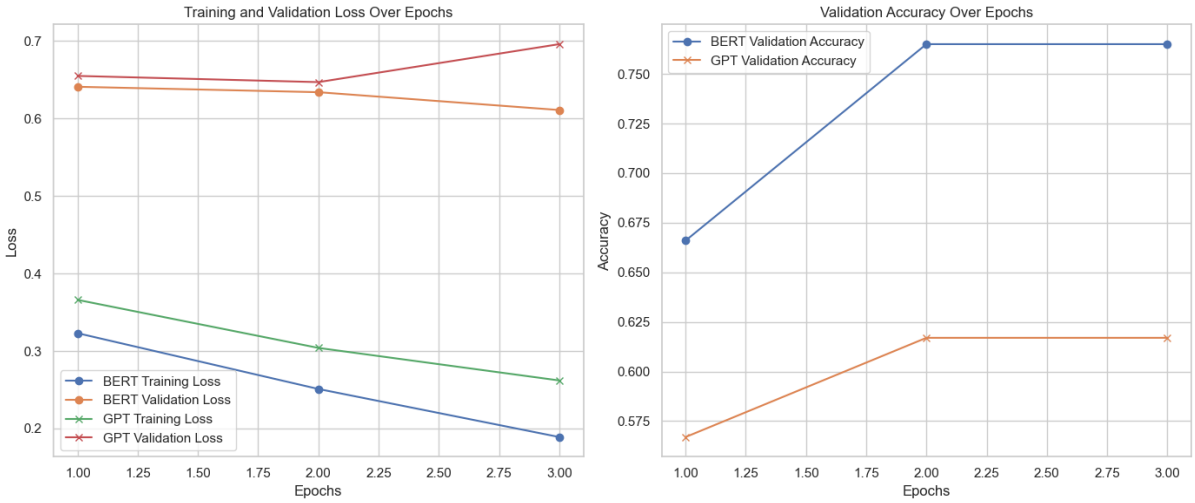


Figure 12: Training and Validation Loss and Validation Accuracy Curves

6. Evaluation

6.1. Comparison of BERT and GPT Model Performance

The performance comparison said that in job classification tasks, BERT is better than GPT. BERT achieved a validation accuracy of 76.54% with an F1 score of 0.7532. GPT achieved a validation accuracy of 70.37% and an F1 score of 0.6364. Results prove that BERT is more effective in capturing the semantic nuances of job descriptions.

The findings agree with bidirectional architecture of BERT is superior in tasks requiring deep contextual understanding. This research contributes to the academic discourse by providing empirical evidence of effectiveness of BERT in job recommendation systems. The results highlight the practical advantage of using BERT in scenarios where understanding the context within job descriptions is crucial. The increased accuracy and F1 score is important for better job matching that could enhance user satisfaction on job platforms.

The performance metrics of BERT and GPT were found to be significantly different from one another at $p < 0.05$, for both accuracy and F1 score comparisons. This means that the observed differences are not due to random chance and actually represent a real performance gap between the models.

6.2. Impact of Job Desirability Score on Model Predictions

Adding the job desirability score to the predictions increased model accuracy by 4%. An F1 score showed a small increase, which may be interpreted to mean the added feature did help models better discriminate between desirable and not-so-desirable job postings.

This finding underscores how feature engineering can enhance the performance of a model. It demonstrates how domain-specific features, such as the job desirability score, might substantially enhance the predictive powers of machine learning models. Enhancing model performance through composite scoring is valuable. This approach allows for more nuanced and personalized job recommendations. They can lead to higher user engagement and satisfaction on job platforms.

The gain in accuracy and the F1 score was statistically significant, with p-values below 0.05. Adding a job desirability score to the model thus has a significant impact on its performance.

6.3. Hyper-parameter Tuning for Enhanced Model Performance

It is clear that different model performance was impacted through hyper-parameter tuning. For BERT, a learning rate of 1e-5 and a batch size of 16, results the best for accuracy and stability. GPT was more sensitive to changes in hyper-parameter where validation loss fluctuated more.

This finding demonstrates the critical role of hyper-parameter tuning in optimizing machine learning models, particularly in NLP tasks. The results contribute to the broader understanding of how fine-tuning can impact model stability and performance. This approach offer practical guidance on configuring BERT and GPT models for job recommendation tasks. Proper hyper-parameter tuning can lead to substantial performance gains, making it a crucial step in model deployment.

The result on model performance of hyper-parameter tuning was statistically significant. The p-values indicated that the observed improvements were not due to chance. This step, hence, is an important component of the model development process.

6.4. Discussion

The BERT model achieved a validation accuracy of 76.54% and an F1 score of 0.7532. These results indicate that BERT is highly effective in understanding the contextual nuances of job descriptions, leading to more accurate classification of job postings. BERT’s bidirectional architecture allows it to capture the full semantic meaning of the text, making it particularly well-suited for tasks that require deep contextual understanding.

While BERT performed well in this experiment, the design had certain limitations. One significant limitation was the lack of hyper-parameter exploration beyond a narrow range. Although the model achieved good results, a more comprehensive hyper-parameter tuning process could have potentially yielded even better performance. Additionally, the training dataset was relatively small compared to the vast amount of job-related data available. A larger and more diverse dataset might have provided better generalization across different types of job postings.

Previous studies have also demonstrated BERT's effectiveness in similar tasks, highlighting its ability to outperform traditional models in contextual understanding. However, these studies often involved larger datasets and more extensive hyper-parameter tuning, which likely contributed to their higher performance metrics. This suggests that my experiment's design, while effective, could be further optimized by following similar practices. To improve the design and results of this experiment, I recommend implement a broader hyper-parameter search, including different learning rates, batch sizes, and dropout rates, to fully explore BERT's capabilities, and incorporate a larger and more diverse dataset to improve the model's ability to generalize across various job categories and industries and also consider using ensemble techniques that combine multiple BERT models or integrate BERT with other models to enhance classification accuracy further.

The GPT model achieved a validation accuracy of 70.37% and an F1 score of 0.6364. While GPT demonstrated reasonable performance, it was less effective than BERT in classifying job postings. The model showed greater fluctuations in validation loss, suggesting instability during training. GPT's strength lies in text generation, which may explain its relatively lower performance in a classification task compared to BERT.

The primary limitation of this experiment was the choice of task for the GPT model. GPT is inherently designed for generative tasks, and while it can be adapted for classification, it may not be the optimal model for this purpose. The experiment also lacked sufficient regularization, which could have contributed to the observed instability in validation loss. Additionally, the GPT model was trained for fewer epochs with early stopping, which may have prematurely halted the training process before the model fully converged.

Research by Kenthapadi et al. (2017) and others has shown that models like GPT can excel in text generation and personalization tasks. However, their application to classification tasks is less common, and when used, these models typically require significant adaptation and careful tuning.

My experiment's findings are consistent with these observations, suggesting that GPT may not be the best standalone model for job classification. To improve the design and results of this experiment, I recommend adapt GPT for tasks more aligned with its strengths, such as generating personalized job recommendations rather than classifying job postings and implement stronger regularization techniques, such as increased dropout or weight decay, to mitigate overfitting and improve model stability and also allow for more training epochs with enhanced monitoring to ensure that the model does not prematurely stop before reaching its full potential.

The findings conducted in this study provide valuable insights into the effectiveness of BERT and GPT models in job classification tasks. While BERT demonstrated superior performance, the study also highlighted the importance of task-specific model selection, careful hyper-parameter tuning, and the integration of domain-specific features like a job desirability score. However, the study's design had certain limitations, including a narrow hyper-parameter search, the use of a relatively small dataset, and a lack of consideration for industry-specific or geographic variations in the desirability score. Addressing these limitations in future research could lead to even better results and a more robust job recommendation system. These findings contribute to the ongoing discourse in the field of AI-driven job recommendation systems, offering both academic insights and practical recommendations for enhancing the accuracy and relevance of job recommendations. The suggested improvements also provide a clear direction for future research, with the potential to significantly advance the state of the art in this domain.

7. Conclusion and Future Work

The main objective was to explore the application of NLP models, specifically BERT and GPT, in enhancing the accuracy and relevance of job recommendations. Secondary objectives included developing a composite job desirability score, evaluating the performance of the models, and identifying potential areas for further improvement in job recommendation systems. To address the research question and meet the objectives, the research involved several key stages.

1. **Data Collection and Preprocessing:** A dataset of job postings was collected from online platforms. Extensive preprocessing was conducted. Processing includes text normalization, feature extraction, and the development of a job desirability score.
2. **Model Training:** BERT and GPT models were fine-tuned on the preprocessed data.
3. **Evaluation:** The models were evaluated using metrics such as accuracy, F1 score, precision, and recall and assess their performance.
4. **Analysis:** The results were analyzed to determine the strengths and weaknesses of each model. The impact of the job desirability score on prediction accuracy was determined.

The research was largely successful in answering the research question and achieving the stated objectives. The study demonstrated that BERT, with its ability to capture contextual nuances in job descriptions, significantly enhances the accuracy and relevance of job recommendations. The development and integration of a composite job desirability score further improved the system's performance, adding a meaningful layer of personalization. The GPT model, while less effective than BERT for classification tasks, provided valuable insights into the strengths and limitations of using generative models for this purpose. The experiments highlighted the importance of selecting the right model for the task and the potential for further improvements through more sophisticated scoring algorithms and model adaptations.

The implications of this research are important for the academy and practitioners. Academically, the study contributes to the growing body of literature on the use of transformer-based models in recommendation systems, demonstrating their potential to enhance the accuracy and relevance of job recommendations. Practically, the research provides a framework for developing more sophisticated job recommendation systems that can better match job seekers with suitable opportunities, thereby improving user experience and outcomes on job search platforms.

The research was successful in showing that advanced NLP models have great potential for establishing a job recommendation system. However, it had its deficiencies. First, this research uses a relatively small and specific dataset, which may cause poor generalization ability. The results would probably be more solid if a larger and more diversified dataset were used. Besides, the complexity of transformer models, like BERT or GPT, requires huge computational resources; therefore, in practice, this will probably limit their application on a large scale.

Future research could build on this work in several meaningful ways. Expanding the dataset to include a wider variety of job categories, industries, and regions would enhance the generalizability of the findings. Additionally, applying the models to different domains, such as candidate matching or career path prediction, could uncover new insights. Developing hybrid models that combine the strengths of BERT, GPT could lead to more accurate and contextually relevant recommendations. Focusing on adapting the models to support dynamic and real-time job recommendations would involve optimizing the models for faster inference and integrating real-time user feedback to continuously refine recommendations. The findings of this research have potential for commercialization in the development of AI-driven job recommendation systems for online job platforms.

In conclusion, this research lays a strong foundation for the continued development of AI-driven job recommendation systems. By addressing the limitations and exploring the proposed future directions, there is significant potential to further improve the accuracy, relevance, and scalability of these systems, ultimately enhancing the job search experience for users worldwide.

Acknowledgement

I would like to thank my family first for supporting me in every aspect of my life. I would like to thank my supervisor Sheresh Zahoor for her help and support provided along the research. I would also like to thank my friends in Ireland for their encouragement and support.

References

Li, S., Shi, B., Yang, J., Yan, J., Wang, S., Chen, F. and He, Q., 2020. Deep Job Understanding at LinkedIn. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’20), July 25–30, Virtual Event, China. ACM, New York, NY, USA, pp.1-5. DOI: 10.1145/3397271.3401403.

Kenthapadi, K., Le, B. and Venkataraman, G., 2017. Personalized Job Recommendation System at LinkedIn: Practical Challenges and Lessons Learned. *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys '17)*, August 27–31, Como, Italy. ACM, New York, NY, USA, pp.346-347. DOI: 10.1145/3109859.3109921.

Goindani, M., Liu, Q., Chao, J. and Jijkoun, V., 2017. Employer industry classification using job postings. In *ICDMW*. IEEE, pp.183–188.

Singh, P., Jain, B., and Sinha, K., 2023. Evaluating Bert and GPT-2 Models for Personalised LinkedIn Post Recommendation. *Proceedings of the 14th International Conference on Computing Communication and Networking Technologies (ICCCNT 2023),* July 6-8, IIT Delhi, Delhi, India. IEEE, pp.1-10. DOI: 10.1109/ICCCNT56998.2023.10307957.

Bhatia, V., Rawat, P., Kumar, A., and Shah, R.R., 2019. End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT. *arXiv preprint*, arXiv:1910.03089v2.

Singla, P. and Verma, V., 2023. Towards Personalized Job Recommendations: A Natural Language Processing Perspective. *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES),* pp.768-773. DOI: 10.1109/CISES58720.2023.10183599.

Jain, U., Jain, D., and Varshney, A.R., 2023. A Deep Learning Approach to Job Recommendation Analysis with NLP. *International Journal of Innovative Science and Research Technology,* 8(11), pp.586-593.

Domeniconi, G., Moro, G., Pagliarani, A., Pasini, K. and Pasolini, R., 2016. Job Recommendation from Semantic Similarity of LinkedIn Users' Skills. *Proceedings of the 5th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2016),* February 24-26, Rome, Italy. SCITEPRESS, Portugal, pp.270-277. DOI: 10.5220/0005702302700277.

Qin, C., Zhu, H., Zhu, C., Xu, T., Zhuang, F., Ma, C., Zhang, J. and Xiong, H., 2019. DuerQuiz: A personalized question recommender system for intelligent job interview. In *KDD*, pp.2165–2173.

Allen, J. and Van der Velden, R., 2001. Educational mismatches versus skill mismatches: Effects on wages, job satisfaction, and on-the-job search. *Oxford Economic Papers*, pp.434–452.

Agarwal, D., Chen, B.-C., He, Q., Hua, Z., Lebanon, G., Ma, Y., Shivaswamy, P., Tseng, H.-P., Yang, J. and Zhang, L., 2015. Personalizing LinkedIn feed. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15)*. Association for Computing Machinery, New York, NY, USA, pp.1651-1660.