

Job-Applicant Matchmaking System using Natural Language Processing

By

Wooi Zhuang Ru

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2023

REPORT STATUS DECLARATION FORM

Title: Job-Applicant Matchmaking System using Natural Language Processing

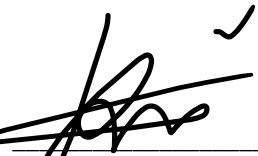
Academic Session: MAY 2023

I WOOI ZHUANG RU

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.



(Author's signature)

Address:

4A, JLN TKL5,
TAMAN KOTA LAKSAMANA
75200 MELAKA

Date: 15/09/2023

Verified by,



(Supervisor's signature)

Aun Yichiet

Supervisor's name

Date: 15 Sep 2023

Universiti Tunku Abdul Rahman			
Form Title : Sample of Submission Sheet for FYP/Dissertation/Thesis			
Form Number: FM-IAD-004	Rev No.: 0	Effective Date: 21 JUNE 2011	Page No.: 1 of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN

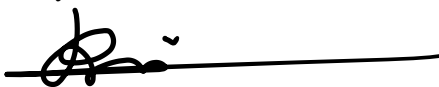
Date: 15/09/2023

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Wooi Zhuang Ru (ID No: 1901406) has completed this final year project entitled “Job-Applicant Matchmaking System using Natural Language Processing” under the supervision of Dr. Aun Yichiet (Supervisor) from the Department of Computer and Communication Technology, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,




Wooi Zhuang Ru

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled “**Job-Applicant Matchmaking System using Natural Language Processing**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____ 

Name : Wooi Zhuang Ru

Date : 13/09/2023

ACKNOWLEDGEMENTS

I would like to express my appreciation and thanks to my supervisor, Dr Aun Yichiet who, through this, has given me an opportunity to engage in exploring Natural Language Processing. I have learnt extensively about Machine Learning and its ins and outs thanks to you.

I would also like to take the time to thank my parents, who have always supported me throughout the course and encouraged me to challenge myself upon adversity.

ABSTRACT

This Job-Applicant Matchmaking System using Natural Language Processing project is for academic purpose. This project aims to provide students with the concept, and implementation process of a matching system catered to both job seekers and employers. Besides just matching job requirements with applicant qualifications, the system also provides personalized job recommendations to job seekers based on their skills, experience, and job preferences, which exposes job seekers to job opportunities that align with their career goals while increasing their overall hire rate. This system uses natural language processing (NLP) techniques to analyse job descriptions and candidate resumes, and machine learning algorithms to recommend the most suitable candidate to a job opening, and vice versa. This process ensures that the employer receives a pool of candidates that meet their job requirements and preferred skills, reducing the need for interviews with unfit candidates. The system is built using Python as the primary language, with the backend consisting of web-scraping, NLP, and data visualization/dashboard libraries such as Selenium, BeautifulSoup, SpaCy, Scikit-Learn, Natural Language Toolkit (NLTK), Gensim, and Streamlit. The system is currently tested with real-world data scraped from well-known job opening hosting sites and shows promising results. The system significantly reduces the time and effort required for recruiters to find the right candidate for a job opening, inversely the job seekers would be able to apply for jobs they are the most well-equipped for.

TABLE OF CONTENTS

TITLE PAGE	i
REPORT STATUS DECLARATION FORM	ii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x
CHAPTER 1 INTRODUCTION	1
1.1 Problem Statement and Motivation	1
1.2 Project Objectives	2
1.3 Project Scope and Direction	2
1.4 Impact, Significance & Contributions	3
1.5 Background Information	3
1.6 Report Organization	4
CHAPTER 2 LITERATURE REVIEW	5
2.1 Review of Technologies	5
2.1.1 Hardware Platform	5
2.1.2 Firmware/OS	5
2.1.3 Database	6
2.1.4 Programming Language	6
2.1.5 Algorithms	6
2.1.6 Summary of Review of Technologies	7
2.2 Review of Existing Systems/Applications	7
2.2.1 Brand Celebrity Matching Model using NLP	7
2.2.2 Automated Resume Screening	10

CHAPTER 3 SYSTEM METHODOLOGY/APPROACH	12
3.1 System Design	12
3.1.1 System Design Diagram	12
3.1.2 Use Case Diagram and Description	13
3.1.3 Activity Diagram	14
CHAPTER 4 SYSTEM DESIGN	15
4.1 System	15
4.1.1 Overview	15
4.1.2 Components	15
4.2 System Components Specifications	17
4.3 System Components Interaction Operations	17
CHAPTER 5 SYSTEM IMPLEMENTATION	18
5.1 Hardware Setup	18
5.2 Software	18
5.3 Settings and Configuration	19
5.4 Word2Vec Model	21
5.5 BERT Model based on Word Embeddings	22
5.6 Cosine Similarity	23
5.7 System Operation	26
5.8 Implementation Issues and Challenges	26
5.9 Concluding Remark	
CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION	27
6.1 System Testing and Performance Metrics	27
6.2 Test Setup and Result	28
6.3 Project Challenges	28
6.4 Objective Evaluation	29

CHAPTER 7 CONCLUSION AND RECOMMENDATIONS	30
7.1 Conclusion	30
7.2 Recommendations	30
REFERENCES	32
APPENDIX	A-1
Weekly Report	A-1
Poster	A-7
PLAGIARISM CHECK RESULT	
CHECK LISTS	

LIST OF FIGURES

Figure Number	Title	Page
Figure 2.2.1	Program Flow of Brand Celebrity Model Matching System	7
Figure 2.2.2	System Architecture for Brand Celebrity Model Matching System	8
Figure 2.2.3	Equation Breakdown of Scaled Dot-Product Attention	9
Figure 2.2.4	Results of the System	9
Figure 2.2.5	System Architecture of Resume Screening System	10
Figure 2.2.6	Cosine Similarity Formula	11
Figure 2.2.7	Results of the System	11
Figure 3.1	System Design Diagram	12
Figure 3.2	Use Case Diagram	13
Figure 3.3	Activity Diagram	14
Figure 5.1	Specifications of Laptop	18
Figure 5.2	Screenshot of LinkedIn's job section	20
Figure 5.3	Printed results from BERT	21
Figure 5.4	Top Correlated Words	22
Figure 5.5	The input section of our application which allows users to add in their csv and OpenAI key	23
Figure 5.6	Communication directly with your datasets.	24
Figure 5.7	Easily view job applications on the web application	25

LIST OF ABBREVIATIONS

<i>NLP</i>	Natural Language Processing
<i>PCA</i>	Principal Component Analysis
<i>GPT</i>	Generative Pre-training Transformer
<i>LLM</i>	Large-Language Models
<i>BERT</i>	Bidirectional Encoder Representations from Transformers

CHAPTER 1

Introduction

1.1 Problem Statement and Motivation

As a soon-to-be graduate and job seeker myself, I have found the stress and anxiety job seeking entails, going through it somewhat during internships. As job seekers often face challenges in finding suitable job, employers on the other hand struggle to identify the most suitable candidates from a large pool of applicants. The current recruitment process is manual, inefficient, and subject to biases, as it may rely on subjective factors such as personal connections rather than objective qualifications and skills, not giving everyone an equal chance and most resulting in people unfitting for a job and missed opportunities on both ends. Even large job hosting companies utilize NLP to some capacity, especially for their searching algorithm as seen in GlassDoor (Schollmeyer, 2023)

1.2 Project Objectives

The main motivation behind researching job-application matchmaking using natural language processing is to counter the inefficiencies and biases present in recruitment processes currently. The advancement of natural language processing techniques and machine learning algorithms, it enables us to create a smarter and more efficient system that is objective and fair for both sides, serving as an aide to hiring decisions. Besides that, by improving the current process of job recruitment, it benefits both sides (job seeker and the hiring company) by reducing the time and effort needed to identify suitable candidates for job vacancies while increasing the chances of finding a good fit between job requirements and candidate skills. Our main objectives are to gain a reasonably high accuracy while making it feasible to be used in real-time in real world scenarios.

I hope to contribute to the development of smarter and more efficient job-matching systems that optimize and promote fairness and diversity in the workplace. To do this, I aim to leverage LLMs, namely BERT and GPT-3.5, to accelerate and improve upon the pre-existing job matching systems currently available.

1.3 Project Scope and Direction

This project aims to develop a Python program that encompasses the collection, synthesizing, pre-processing and prediction of jobs and resumes using GPT-3.5 and BERT. Then it analyses the job requirements and qualifications using natural language processing techniques, it will then match them with the relevant resumes from a resume database. Beyond this, the scope of the project will also encompass analysis on the data on a dashboard. To ensure the data does not have any abbreviations or phrases that are overlooked, we are focusing on entry level jobs in the Information Technology field in Malaysia. Besides that, the data is synthesized using ground truths of a pre-existing labeled dataset, to counter the absence of a readily available dataset that fits our scope. This project has a mixture of prompt engineering, work with Large Language Models and Natural Language Processing.

1.4 Impact, Significance & Contributions

The data itself acts as a contribution as we use web-scraping to get previously not easily attainable data that is unique to its date and time, since job postings will be deleted once they have recruited a candidate.

Besides this, the NLP approach enables the system to learn from past job matches and manual corrections to predictions, inherently improving its accuracy over time. Furthermore, the use of web scraping techniques ensures the data reflects real-world situations and use-cases that can be further explored. These provide invaluable resources for future reference and starting points for new projects, providing a foundation for further research and development in natural language processing-based recruitment systems.

1.5 Background Information

The field of natural language processing (NLP) has been a rising hot topic in recent years across various industries. NLP started off as the intersection between artificial intelligence and linguistics, but the NLP today has matured and now consists of using computational methods to analyse and understand human language, which includes the capability to recognize patterns and structures in language data (Nadkarni et. al., 2011). These can be extended to producing human-like responses and sentiment analysis.

In job recruitment, NLP techniques are commonly used to automatically screen job resumes, to reduce intervention needed from human personnel. This is normally done through parsing resumes using a semantic search to get context and find keywords in an unstructured language to obtain desired information (Sinha et. al, 2021). This approach has significantly improved the efficiency of the recruitment process. Besides this, integration of NLP has also led to more accurate and efficient matching of job candidates since companies are able to quickly skim through hundreds and thousands of applicants based on specific skills and qualifications.

1.6 Report Organization

Chapter 1 details the project's overarching mission, objective, and initial ideation phase. Chapter 2 describes the initial work and research done to ensure the logic and the concept of the project is sound, we did this by having an analysis of the technologies used and reviewing preexisting systems. For Chapter 3, there is system methodology where we talk about how the system will work and its use cases. Chapter 4 describes the system design and its inner workings. Chapter 5 details the actual system implementation and its work in real time. Chapter 6 is where we evaluate our system and their performance in comparison with each other. Chapter 7 is our concluding remarks and a post-mortem overview of the entire project.

CHAPTER 2

Literature Review

2.1 Review of the Technologies

2.1.1 Hardware Platform

The hardware used is a robust Lenovo Legion 5 with a 16GB RAM and dedicated GPU. This is so that we can handle user requests smoothly and conduct complex calculations at a relatively fast rate. At scale, this project would have to have a serverless setup or a powerful server to handle requests from multiple clients.

2.1.2 Firmware/OS

This system runs using the Windows operating system, leveraged by using Windows compatible applications and tools like:

Jupyter Lab: It enables code execution, data visualization, and documentation within a easy to use interface. It is commonly used for LLM studies and development.

VSCoDe (Visual Studio Code): For easy to edit codebases for frontend, VSCoDe is used to develop and see real-time changes in Streamlit effectively.

Streamlit: As the core framework for our web-based interface, Streamlit integrates seamlessly with the Windows OS. It allows us to create interactive web applications with minimal code, enhancing the accessibility of our system.

2.1.3 Database

For databases, we have 2 formats to store them in, which both get converted into a Pandas Dataframe later in the prediction stage.

- **CSVs (Comma-Separated Values):** CSV files help to store our job listings, resumes, and related data. CSVs were chosen because they offer simplicity and ease of use, making them suitable for initial data input and storage. It is also effective for client-side input as it is widely used and accepted in the corporate world
- **ChromaDB:** Helps to host the embedding database that stores word embeddings generated from job listings and resumes. The presence of word embeddings enhances our model's capability to perform context-aware matching via either efficiently storing or indexing embeddings for retrieval.

2.1.4 Programming Language

The Python programming language was chosen due to its popularity and suitability in data science and NLP domains, since it already has a multitude of tools that cater to that industry at its disposal. Frameworks and libraries like Transformers (NLP tasks), Scikit-Learn (Machine Learning) and Streamlit (Web development) made it the best language to employ in this project.

2.1.5 Algorithms

Our project uses multiple algorithms, namely BERT (Bidirectional Encoder Representations from Transformers) by Google and GPT 3.5 (Generative Pre-trained Transformer 3.5) by OpenAI. These were chosen because of their robust prediction and semantics understanding ability while remaining economical and time efficient. BERT excels in contextualized word embeddings and semantics, which are key towards a semantics and acronym heavy domain like job matching (where skills have individual names and meanings unique to their industry), whereas GPT-3.5 has been the economical industry leader for the past year, also exceling for its text generation ability and understanding of natural language.

2.1.6 Summary of Review of Technologies

In summary, our matching system is built upon a robust technological foundation, that currently operates in a server-based environment, displaying its frontend using Streamlit. The choice of Windows OS supports the development tools used, including Jupyter Lab, VSCode, and Streamlit. Our data is saved as CSVs and ChromaDB, ensuring widespread usage capabilities while maintaining its efficient data storage and retrieval. Using the Python programming language, we use BERT and GPT-3.5 algorithms empower the system to perform advanced NLP-based matching.

2.2 Review of Existing Systems/Applications

2.2.1 Brand Celebrity Matching Model based on Natural Language Processing

In this piece of literature, they proposed a celebrity brand matching algorithm normally used for endorsement, which combines text summarization with word vector matching through Word2Vec and Crawler Module (Yang et. al., 2022). The algorithm takes in 2 inputs in the form of a Celebrity database and a Brand database, which are piped into the Crawler module followed by the text summary module.

The data is scraped from descriptive entries of the Baidu Encyclopedia alongside news about the celebrities as supplementary descriptors.

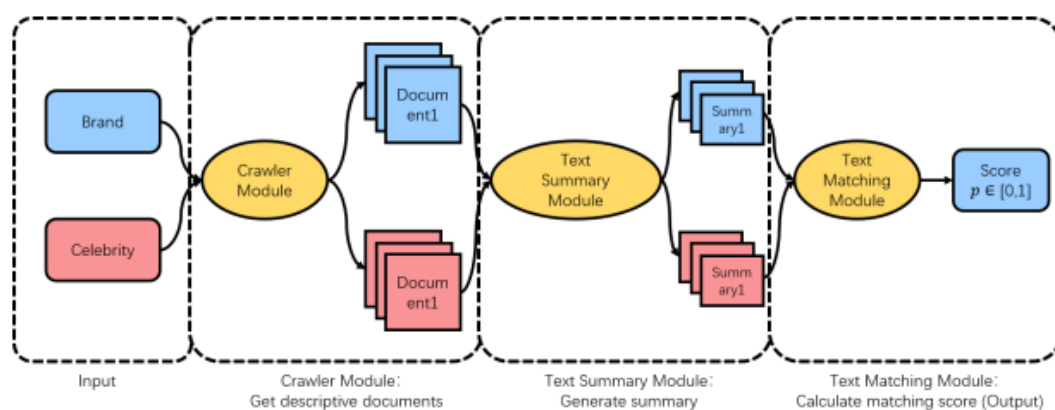


Figure 2.2.1 – Program Flow of Brand Celebrity Model Matching System

The text crawler module summarizes long documents to condensed semantics as a method to reduce noise. Their assumption provided, is that only one descriptive document would crawl for each celebrity or brand, hence every celebrity and brand is assumed to be unique.

For the text summary module, the paper talks about using word2vec model to create a word vector matrix. In this instance, they used seq2seq which is the most common Generative text summarization method—the encode-decoder structure. This is done by encoding an input sequence into a hidden layer and the decoder then generates a final output sequence according to the vector.

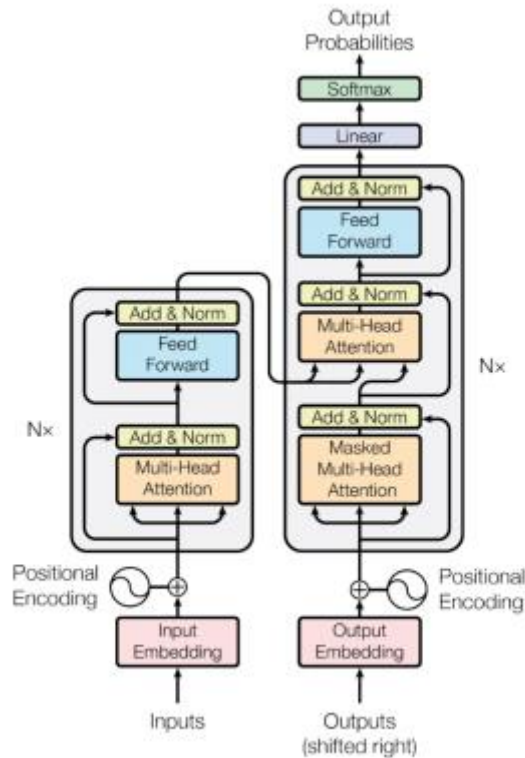


Figure 2.2.2 System Architecture for Brand Celebrity Model Matching System

The paper inputs a special character [CLS] as the decoding symbol and uses $y_0 = E[\text{CLS}]$, y_i as each output character. According to the training output, the error of the whole neural network can be calculated as the cross-entropy error. They also make note of adopting the greedy strategy, taking the maximum output probability's character out as the current character until the output terminator has reached its maximum output.

$$att_{y_i} = ScaledDot - ProductAttention$$

$$(y_i, (y_i, (y_i) = softmax(\frac{\sqrt{d_{y_i}}}{y_i y_i^T}) \quad (13)$$

Then, the input of the encoder is introduced to calculate an interactive multiplicative attention, as shown in equation (14).

$$att_{y_{i+1}} = Scaled Dot-Product Attention$$

$$(E, y_{0:i}, y_{0:i}) = softmax\left(\frac{y_{0:i}^T}{\sqrt{d_E}}\right) y_{0:i} \quad (14)$$

Figure 2.2.3 – Equation Breakdown of Scaled Dot-Product Attention

In the matching module stage, they are now left with the summary $S_i = \{s_1, s_2, \dots, s_n\}$, where $S_i = \{i = 1, 2, \dots, 3\}$ is the word in the summary. They then splice these abstracts and match them using word2vec.

Based on the Matchpyramid model, they create a dot product where its similarity is the similarity between the words in Celebrity and its descriptive summary in Brand. They then stack multiple CNN layers along with pooling layers to form a generalized matching feature map which is flattened and inputted into a multi-layer perception (MLP) to get a final prediction result, the equation: $o = W_2\sigma(W_1z + b_1) + b_2$.

Overall, the analysis found that the recall rate is lower but still acceptable, but the accuracy & precision are remarkably high. They also scored much higher than other models comparatively.

Model	Precision	Recall	F1 score	Accuracy
Random Guess 0	0	0	0	0.945
Random Guess 1	0.055	1	0.105	0.055
Text CNN	0.925	0.511	0.530	0.926
Our Model	0.990	0.811	0.892	0.989

Figure 2.2.4 – Results of the System

The proposed algorithm in this literature is very adaptive which is a strength, they use a combination of Matchpyramid model and multiple CNN layers to create a highly generalized matching feature map, which contributes to the high accuracy and

precision of the algorithm. However, the assumption that every celebrity and brand is unique may not always hold true, which could potentially lead to inaccurate matches. Furthermore, the training process for the algorithm may be time-consuming and computationally expensive, which may limit its scalability.

2.2.2 Automated Resume Screening

In this piece of literature (Daryani, et. al., 2020), they are proposing an automated resume screening system which consists of 2 phases, information extraction and classification, it should be noted that this does not aim to cater to job seekers as it pertains more to easing headhunter's jobs in reading over mountains of resumes to see if anyone fits certain criteria.

Like literature 1, this literature has an architecture split into 2 phases, which essentially are Pre-processing and Predicting, these two also share the similarity of using word2vec or word vectorization as their form of similarity weightage.

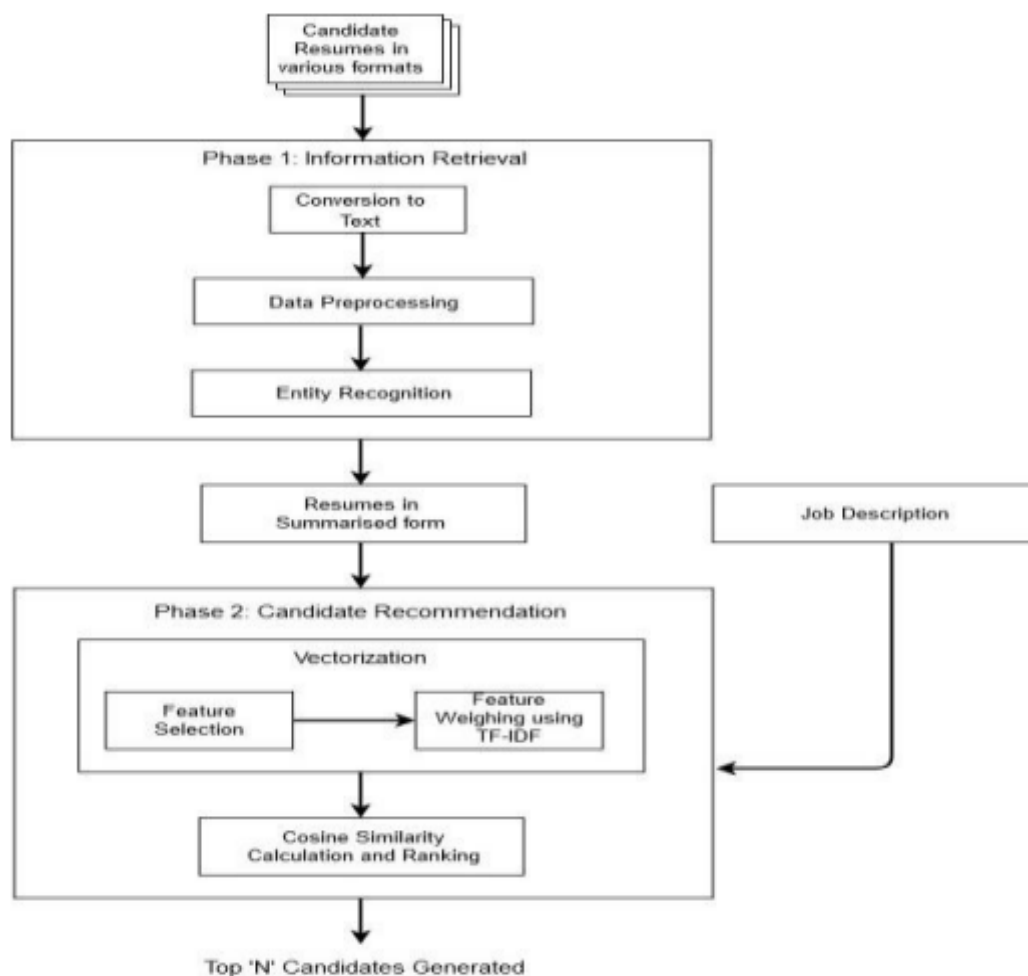


Figure 2.2.5 – System Architecture of Resume Screening System

Key points to note are after tokenization, the extracted information is converted into a summarized version in the form of a JSON so that it can be easily used for further processing tasks in the next phase of this resume screening system.

With all the insignificant and redundant information removed, the task of the screening officials is simplified, and they can better analyse each resume with better efficiency. Their system uses vectorization and similarity calculation to get the cosine similarity measurement between multiple resumes based on the job query to determine the fit for each job. Cosine similarity can be derived through this formula:

$$\text{cosine_similarity}(R_a, J_b) = \frac{\sum_{i=1}^n (w_i^a \times w_i^b)}{\sqrt{\sum_{i=1}^n (w_i^a)^2} \times \sqrt{\sum_{i=1}^n (w_i^b)^2}}$$

Figure 2.2.6 – Cosine Similarity Formula

Based on this, the results are as follows:

Table 1: Resultant ranked list of candidates prioritized by similarity score			
Candidate (Resumes)	Number	Cosine Similarity Score	Rank for the Job
Candidate 2		0.6802823482591744	1 st
Candidate 4		0.6514716047844277	2 nd
Candidate 3		0.49850131321205904	3 rd
Candidate 1		0.4907052756267933	4 th

Figure 2.2.7 – Results of the System

The proposed system has a structured approach to screening and satisfies its niche well as it only caters to company hiring agents, over time company specific algorithms will get better as they learn more about who and why the human selectors chose them, better fitting the company culture as the company uses the system.

For weaknesses, the system relies heavily on cosine similarity to measure the fit between resumes and job queries, which does not always capture all relevant factors that are important for a job. Besides this, the system would still need manual intervention after screening.

CHAPTER 3

System Methodology/Approach

The processes of the project were categorized into different phases in the development, which were project pre-development, data pre-processing, model training architecture building and data training, and prediction on test dataset.

3.1.1 System Design Diagram

Our system consists of 2 data sources as inputs, job applications and resume.

After that, we use the keyword extraction tools on both summarized datasets to get keywords from the summarized descriptions, using the summa library. We then proceed with the text matching using the keywords from both keyword datasets as the dictionary through word2vec, before that the keywords are lemmatized, and the stop words are removed. Then we put the vector embedding into ChromaDB which enriches our LLMs (GPT3.5 and BERT) to get the top applicants that fit that role the best.

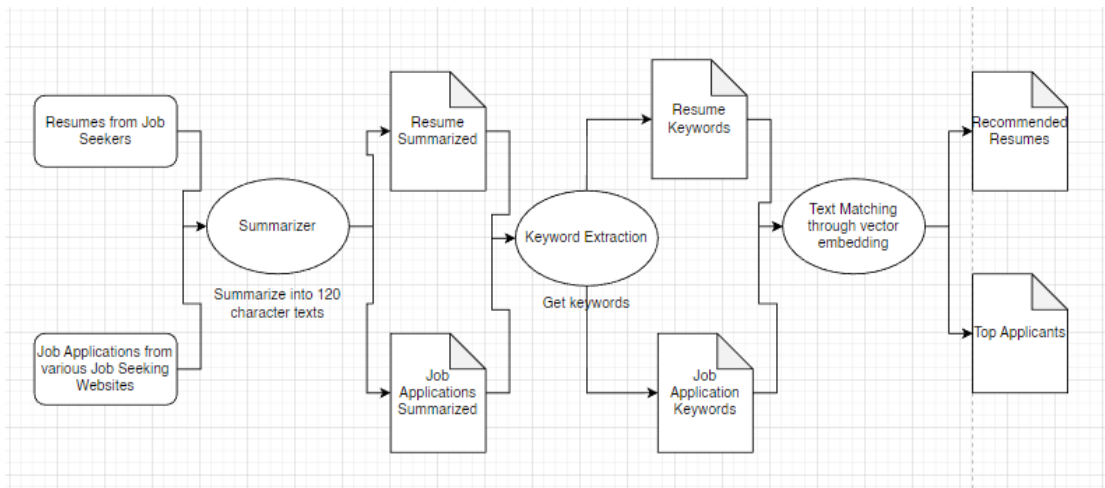


Figure 3.1 System Design Diagram

The usage would now be different depending on the input of the user, if they choose a job application as the input value, we will return the most capable applicants in the Job Seekers dataset. This takes in job applications from a user-inputted CSV.

3.1.2 Use-Case Diagram and Description

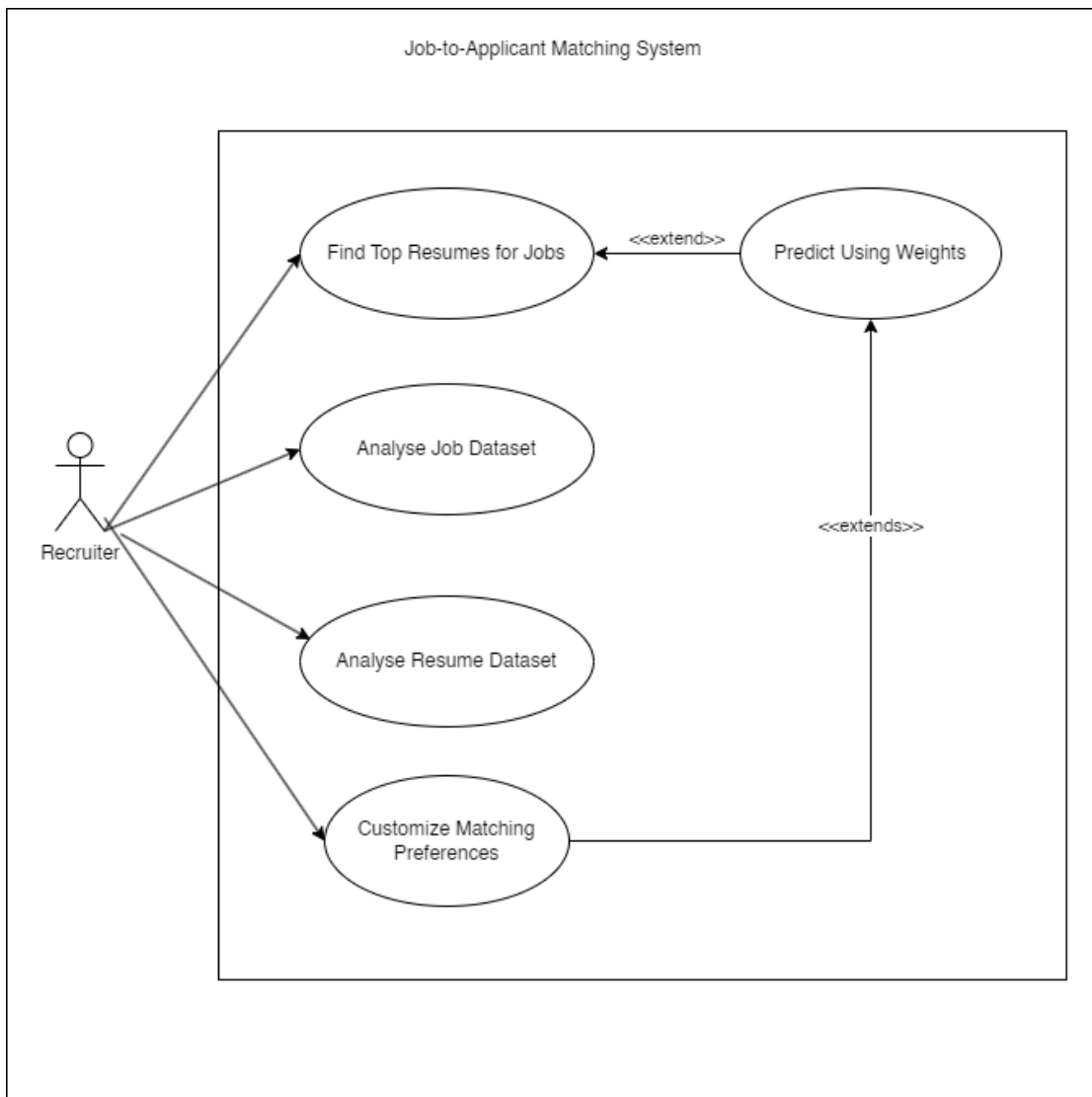


Figure 3.2 Use-Case Diagram

Our target users are recruiters, in this system you can match resumes to jobs via using customizable matching preferences like prior matching history and weightage of GPA. In the diagram above, it illustrates the use cases of this system in the real-world, where they can analyze and predict resumes in real time.

3.1.3 Activity Diagram

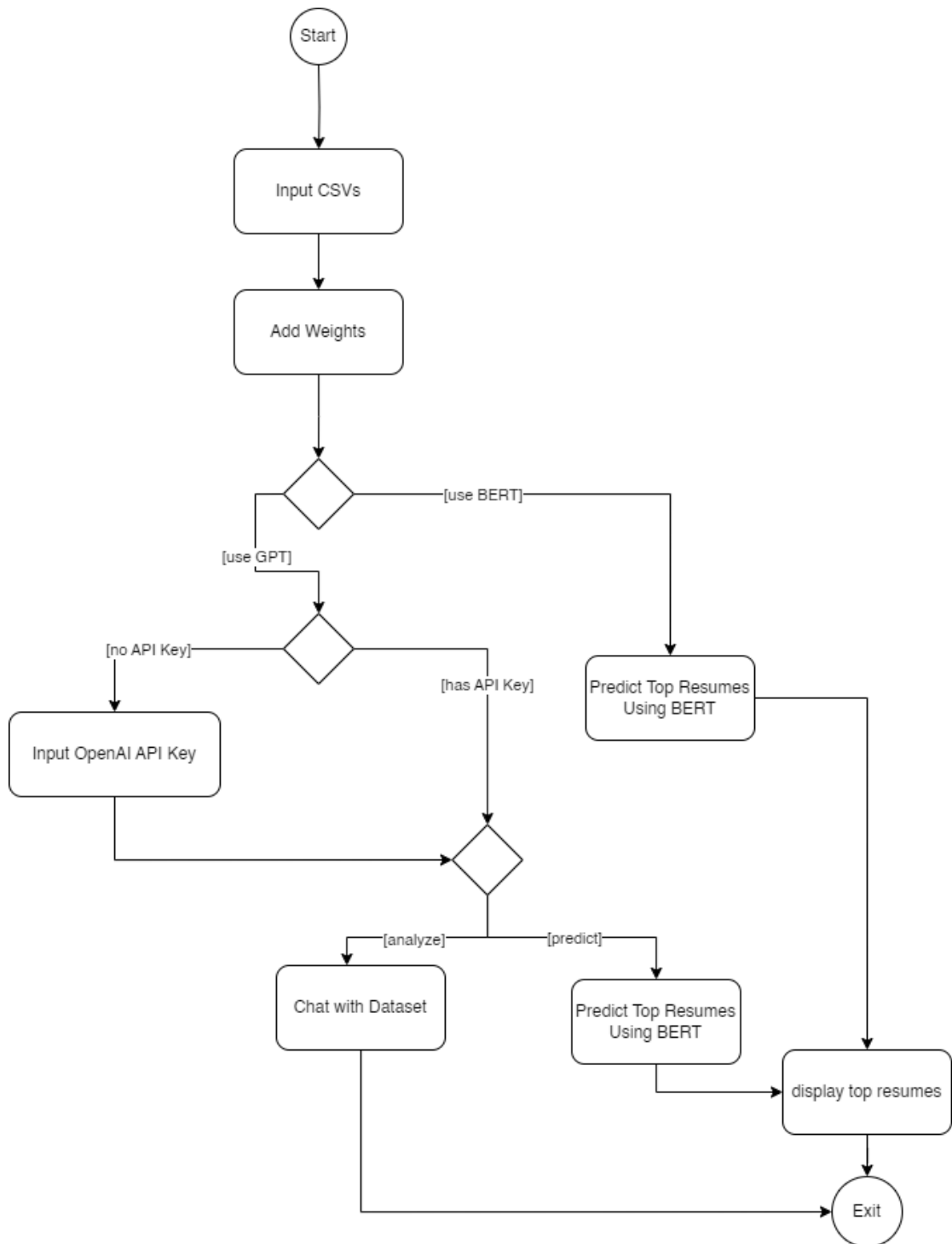


Figure 3.2 Activity Diagram

CHAPTER 4

System Design

In this chapter, we will delve into the system design for the job-to-applicant matching system using Natural Language Processing (NLP) techniques, specifically comparing the use of BERT Large Language Model (LLM) and GPT-3.5. This chapter provides an in-depth look at how the project is developed, including system block diagrams, system component specifications, circuits and components design, and system component interaction operations. The aim is to provide all the necessary information for someone to understand and potentially replicate the system.

4.1 System

4.1.1 Overview

The system block diagram provides a top-down view of the key components and their interactions within the job-to-applicant matching system. It illustrates how the system processes job listings and resumes to generate matching scores.

4.1.2 Components

The main components of the system block diagram include:

- Job Listings Dataset: a CSV file that is inputted by the user.
- Resumes Dataset: a CSV file that is inputted by the user.
- NLP Model (BERT LLM/GPT-3.5): Pretrained LLM models with
- Embedding Database (ChromaDB): a feature-rich embedding that stores embeddings to search by nearest neighbors.
- Prior Match History: a labelled CSV file with top matches for jobs, serving as a preference guide for the NLP models when matching.
- Match Weights: Skews the significance of matching history and CGPA's role when matching. The higher the value, the more intensely these weights impact the resultant list.

4.2 System Components Specifications

The resume dataset is provided by the user, must consist of a CSV with columns:

CHAPTER 5

- Full Name
- Email
- Summary: Summarized text for us to extract their key information
- Extracurricular Activity Name
- Position in Extracurricular Activity
- Responsibilities: Responsibilities within that position
- Achievements
- Soft Skills
- Hard Skills
- Degree Name
- University/Institution Name
- Graduation Date
- CGPA

For synthesized data, we based it off the original dataset we found and directly converted them into a pandas dataframe. Our Job CSV/Dataframe has these columns:

- Index
- Job Title
- Company Name
- Job Location
- Job Type
- Summary: Summarized text using our Word2Vec to extract key points.
- Responsibilities: Responsibilities the applicant will have if they got the role.
- Tech Stacks: Specific technologies or software tools that are relevant to the job
- Educational Qualifications: The preferred degree of choice by the employer
- Required Skills and Competencies: Skills the applicant should have when applying
- What We're Looking For: The original blurb they had when hiring/writing the job posting.

4.3 System Components Interaction Operations

The component's interaction operations consist of the following steps:

Data Ingestion: Before any other operation can be done, users are required to provide 2 datasets (Resume and Jobs) that have the predefined columns. These datasets are then ported to preprocessing before proceeding to NLP to ensure data integrity and prepare them for subsequent processing stages.

NLP Processing: This process takes place when users choose which models, they would like to interact via the tabs which includes tokenization and embedding generation. In the case of using GPT-3.5 they would have to provide an openAI key to have access to GPT functionalities

Embedding Storage: Embeddings are generated from job listing and resume datasets and stored on our embedding database (ChromaDB). This provides additional context to LLMs if needed, these also serve as the core component to chat with your datasets.

Match History: The system consults the Prior Match History, which contains historical data of top job-resume matches. This historical data informs the matching process by providing insights into past successful matches.

Match Generation: Matches are generated with the inputted information, consisting of the 2 CSV's embedded vectors, users weights and an optional match history. This generation often takes 10 – 15 seconds for GPT and around 3 – 5 seconds for BERT. This is mainly due to GPT's limitations in bandwidth when used at scale.

Match Output: The output is in the form of a ranked list with a default threshold of a match by 85%, this can be altered based on the user's provided weight and threshold in the alterations section.

CHAPTER 5

System Implementation

5.1 Hardware Setup

The hardware involved in this project is a computer, with specifications enough to run long hours of web-scraping. This device will also be used in programming and running prediction models, alongside hosting the dashboard locally.

Description	Specifications
Model	Lenovo Legion 5 15ARH7
Processor	AMD Ryzen 7 6800H
Operating System	Windows 11
Graphics	NVIDIA GeForce RTX 3050 Ti
Memory	16GB RAM
Storage	1TB SSD

Figure 5.1 Specifications of laptop

5.2 Software

This software is built using Python as the backend for the web-scraping, natural learning processes, and data visualization/dashboard.

The version of Python the project is built upon is 3.11. In this project we also made use of technologies like ...

- Bert-Base-Uncased Pretrained Model
- Scikit Learn
- WordCloud
- Transformers
- JupyterLab
- GPT 3.5 Turbo
- Streamlit
- Matplotlib

5.3 Settings and Configuration

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 5

To get adequate data for our project due to the unavailability of preexisting labeled datasets, we opted to use synthetic data that has grounded truths in the form of a small preestablished dataset found online. From that, we created a prompt where GPT would synthesize a set of job descriptions, for each job description, they would have 3 generated resumes. These resumes are:

- 100% (The best resume possible that would fit the job description perfectly)
- 50% (Middle or not so good in contrast with the best resume but far better than the 0% resume)
- 0% (Completely unrelated and unfit resume that is still within the technological industry)

In total we constructed 20 paired job descriptions and 60 generated resumes with the prior preestablished dataset we found online. This would serve as our “Matching History” also known as our finetuning dataset. This amount was set due to the restrictions of GPT’s model charges and in the interest of time.

Following that, we further synthesized unrelated jobs and resumes as our new data, this is also variable to user’s custom jobs and resumes. This totaled to 100 resumes and 30 jobs that are not labelled. These were based on our scraping work from LinkedIn we did in the prior work from FYP1, where we have collected data totaling to 593 unique entries, with ‘Web Development’ yielding 90 entries, ‘Software Engineering’ yielding 251 entries, ‘Programmer’ yielding 87 entries and ‘Computer Science’ yielding 165 entries. These jobs that were generated were limited to Malaysian technology companies hiring entry level jobs. This was done to reflect real world data.

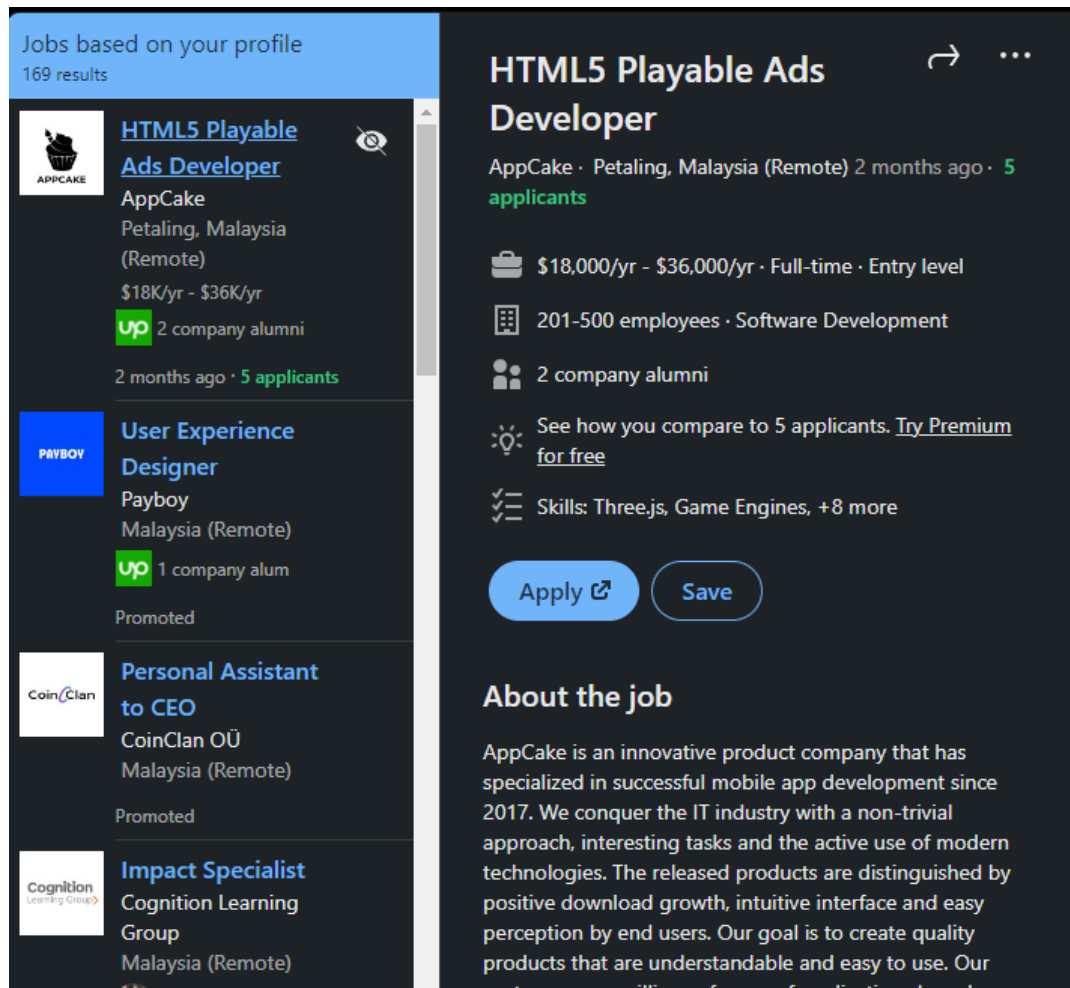


Figure 5.2 Screenshot of LinkedIn’s Job Section

The data is then pre-processed further in Python in the form of a data frame, we pre-process the raw text into lowercase, and remove unwanted punctuation and special characters—taking into consideration the special exceptions like ‘C#’ or ‘Notepad++’. After that, we lemmatize the words and remove common stop words before performing tokenization. In this process, Streamlit is used as a prototype frontend to have easier access to graphs and toggles.

CHAPTER 5

We also implemented a summary column for jobs and resumes to get key information that would be valid for the prediction process. This also reduces the overall token count which would be a big bottleneck if this project was implemented at scale. The figure below shows the information that the BERT and GPT-3.5 models consider when matching resumes with jobs. This allows us to gain a full view of key priorities and achievements of an individual without sacrificing performance.

```
Top Resumes for Job (Index 11): Game Developer (Unity)

Resume 32 (Combined Score: 0.9028)

Full Name: Megan Chong
Email: meganchong@generated.com
Summary: A computer science enthusiast with a flair for game development. Proficient in Unity and Unreal Engine.
Extracurricular Activity Name: Game Development Club
Position in Extracurricular Activity: Game Developer
Responsibilities: Developed game prototypes
Achievements: Developed playable game demos
Soft Skills: Game Development
Hard Skills: Unity, Unreal Engine
Degree Name: Bachelor's in Game Development
University/Institution Name: University of Malaya
Graduation Date: April 2023
CGPA: 3.45
```

Figure 5.3 Printed Results from BERT

5.4 Word2Vec Model

For finding correlation between words and getting job recommendations through words, our Word2Vec model is used in principal component analysis (PCA) which enables NLP to better process large sparse matrices (R. Drikvandi and O. Lawal, 2023), resulting in faster calculations and principle components, this allows our model to be more scalable as we obtain more data.

PCA is applied and using correlation matrix, we compute the eigenvalues and eigenvectors into a square array where we will sort them in descending order. The maximum variance can then be explained through taking the first 2 components of the projecting. To better illustrate the correlation between words, in the Streamlit app there is also a finder where you can see top words that have high correlation to your words, followed by ‘recommended jobs’.



Figure 5.4 Top Correlated Words

From here, the projecting is projected into a new dimension with an x and y axis to form a scatterplot, where the closer something is to another determines the correlation between them and vice versa. This is based off (Metsalu & Vilo, 2015)'s visualizations using ClustVis with PCA and heatmaps.

5.5 BERT Model based on Word Embeddings

Using Word2Vec, we ported the word embeddings into the BERT and GPT-3.5 models using ChromaDB. This integration allows for rapid semantic understanding in real-time, which will in turn result in improved matching accuracy, and the ability to handle domain-specific language effectively. This is especially crucial in the technology space as many acronyms and semantics are present (eg. "React and Notepad++") It also serves as a good checkpoint to monitor the system's data while maintaining efficient retrieval mechanisms. Using our BERT tokenizer, we convert the textual data into BERT's preferred format, which includes the standard truncation, padding and setting maximum token length. After getting our BERT embeddings, we use mean pooling to condense the embeddings, allowing them to be more manageable before doing similarity calculations.

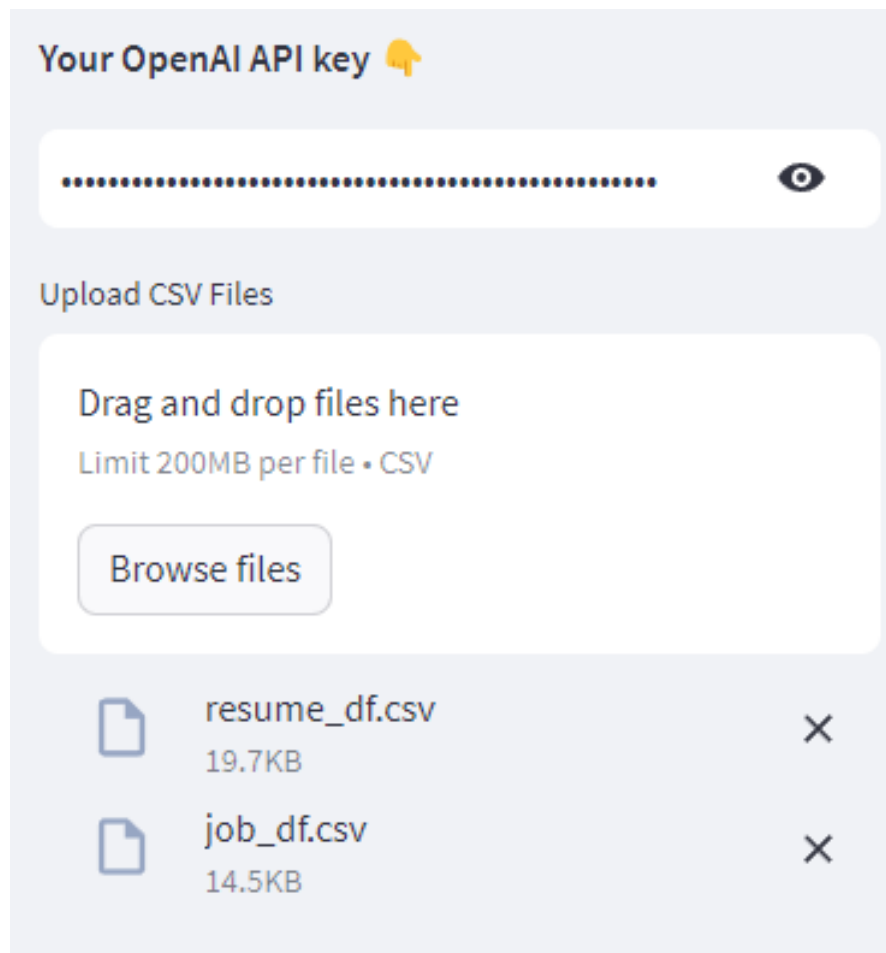
5.6 Cosine Similarity

For BERT, we use Scikit-Learn's cosine similarity function to calculate cosine similarity. This results in a number that scores higher based on how much the job

embeddings match with resume embeddings. This is done both for the finetuning dataset and the actual dataset used in real-time. GPT-3.5 also makes use of the BERT embeddings as it can also understand it, the embeddings made are saved into ChromaDB for easy access at anytime during the prediction process.

5.7 System Operation

To start off with using our system, users must first add in the relevant CSVs (resumes and jobs). After that, if the users would like to make use of OpenAI, they must first add in their own unique API key. This is done to ensure security of development use API key, and is standard practice among GPT based applications that are not paid.



The screenshot displays the user interface for adding an OpenAI API key and uploading CSV files. At the top, there is a section titled "Your OpenAI API key" with a yellow hand icon. Below this is a text input field containing a series of dots, with an eye icon on the right side to toggle visibility. Underneath is a section titled "Upload CSV Files" with a white background. It contains the text "Drag and drop files here" and "Limit 200MB per file • CSV". A "Browse files" button is located below this text. At the bottom, two files are listed: "resume_df.csv" (19.7KB) and "job_df.csv" (14.5KB), each with a file icon and a close (X) button.

Figure 5.5 The input section of our application which allows users to add in their csv and OpenAI key

If opting for use of OpenAI and subsequently GPT-3.5, you can openly communicate with your dataset, gaining a more important insight into your document that would be tedious to read through manually otherwise.

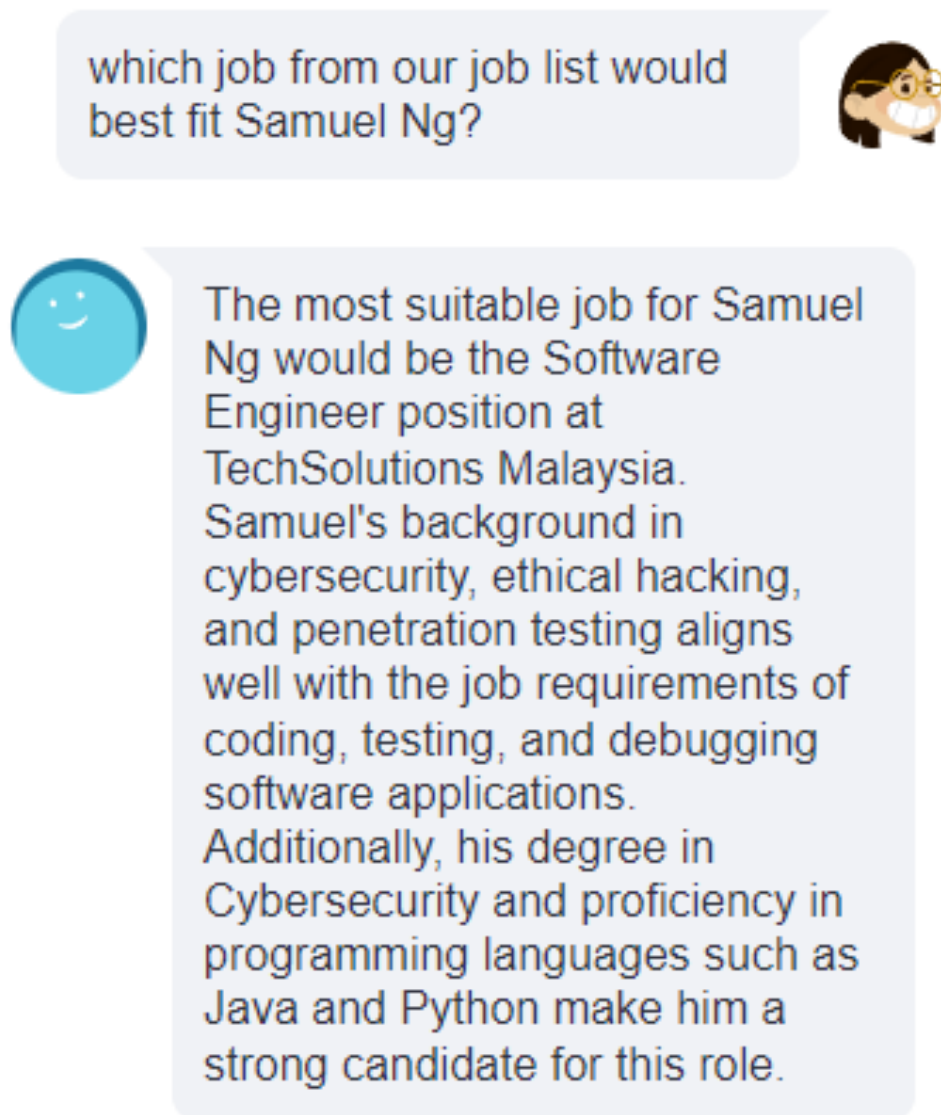


Figure 5.6 Communication directly with your datasets.

Regardless of which model the users are using, they can easily understand and predict any job they have from the Job CSV, they can also alter the weightage CGPA plays when predicting top candidates, toggle it higher to prioritize high CGPA holders rather than users that have preexisting skills more fit for the role. Users also have the option to add in a match history, which serves as a reference point for predictions, our embeddings will make sense of the matches and cater predictions to their preferences based on the historical data.

Select	Index	Job Title	Con
<input type="checkbox"/>	0	Blockchain Developer	Cryp
<input type="checkbox"/>	1	iOS App Developer	Mob
<input type="checkbox"/>	2	Android App Developer	App
<input type="checkbox"/>	3	Machine Learning Engineer	AI Te
<input type="checkbox"/>	4	Cloud Solutions Architect	Clo
<input type="checkbox"/>	5	UI/UX Designer	Desi
<input type="checkbox"/>	6	Data Scientist	Data
<input checked="" type="checkbox"/>	7	Cybersecurity Analyst	Sec
<input type="checkbox"/>	8	Front-end Developer	Dev
<input type="checkbox"/>	9	Software Quality Assurance	Tecl

Cybersecurity Analyst

from SecureNet Solutions | Industry: Cybersecurity

SecureNet Solutions is hiring a Cybersecurity Analyst to protect against cyber threats and vulnerabilities.

Responsibilities: Monitor and respond to security incidents

Tech Stacks: Firewalls, IDS/IPS, SIEM, & related competences (Cybersecurity, threat detection, risk assessment)

SecureNet Solutions is looking for a Cybersecurity Analyst to strengthen our security posture and protect against threats. Candidate should possess a Bachelor's degree in Cybersecurity or related field.

Figure 5.7 Easily view job applications on the web application

5.8 Implementation Issues and Challenges

At scale we would require a high rate of API calls to GPT-3.5 if proceeding in real world scenarios, this is due to the limited bandwidth allotted to each user from OpenAI. This issue would not be applicable for BERT or any other LLMs ran locally but would require a high-performance machine to keep up with demand.

Besides this, prompt engineering was also a persistent issue, as words for the GPT-3.5 prompt was vital to get consistent and accurate matches consistently, through experimentation, it is found that using the few-shot method is the best way to give contextual templates for GPT to respond to, which reduces hallucinations and tangential information.

5.9 Concluding Remark

The development of the matching system has not been without its challenges, ranging from API call limitations to prompt engineering complexities, but it has effectively strengthened my understanding of LLMs and their inner workings.

We have overcome those challenges by adopting a multi-faceted approach which combine local processing with cloud-based resources, and optimizing prompts for GPT-3.5, besides that we found that implementing efficient data preprocessing is the key to making a system that not only addresses these challenges but also leverages them as opportunities for growth.

CHAPTER 6

System Evaluation and Discussion

6.1 System Testing and Performance Metrics

To assess the performance and functionality of this application in real life situations, this app is testing via creating a synthetic ground truth dataset by manually matching a subset of job listings with resumes. Using our custom synthetic dataset, we can observe the accuracy of matches in when taking in prior job match history. Human evaluations were also conducted to counteract the ambiguity that interacting with LLMs bring, since there is no definite accuracy or related metrics.

As for accuracy of the matches, a test to get Correct Matches (CM) and Total Processing Time (TPT) was done in tandem to see the best blend of correctness and efficiency. Besides that, we also considered the Matching Efficiency Ratio (MER) which measures the efficiency of matching process in terms of time and accuracy to understand the feasibility of this project working at scale in real time. The calculation was done using our unlabeled dataset, and only took into consideration the top matches as the other matches were quite variable when involving weights and other metrics. In essence, the higher the MER value, the more effective the system is to generate accurate job-resume matches in real-time.

Our tests were done using the labeled dataset as our ‘Match History’ and CGPA as the default 0.1.

6.2 Testing Setup and Result

By randomly selecting 20 out of our job's dataset, we opted to match them based on the prior history. This prior history is not weighted or skewed. The result of our experiment is as follows:

	BERT	GPT-3.5
Correct Matches (out of 20)	15	19
Total Processing Time (seconds)	15.2	23.8
Match Efficiency Ratio	98.68	79.83

BERT achieves a higher MER overall, however the GPT-3.5 is handicapped due to its limitations in API call rate, it would work significantly better at a higher rate which would increase MER. It should be worth noting that GPT-3.5 although slower, it has a exceedingly high accuracy in contrast with BERT. BERT achieved 15/20 whereas GPT 3.5 obtained 19/20. This is consistent with more advanced testing as well, BERT's results varied wildly, whereas GPT-3.5's stayed consistent throughout various tests with varying weights.

6.3 Project Challenges

To obtain a comparable scenario between BERT and GPT 3.5, the ambiguity of the inner workings of GPT was apparent. It did not have readily available metrics like BERT, and it was also an external API which made it difficult to run longstanding tests in.

Prompt engineering also played a crucial role in obtaining high match percentages when it comes to GPT-3.5. The prompt's wording was important to note when attributing GPT's high performance in contrast with BERT, through getting the correct prompt, GPT 3.5 performs reliably, producing similar results each time.

Another issue encountered during this project is the readily availableness of datasets, since labelled datasets were not accessible by the general public as they are often

reserved as confidential in hiring sectors, we had to improvise by obtaining a small dataset and expanding it using LLMs in order to get a substantial simulation of real-world use-cases. This brings up many issues such as data inconsistencies and biases due to the synthesizing process, there is no doubt that having real world data would only benefit this model as it would have accurate and more volume of data to justify its predictions (both GPT3.5 and BERT).

Since we are accessing sensitive information like job seeker's resumes and personal information, it is critical to implement robust security measures to preserve the confidentiality of the data. Another aspect to this is ensuring the data comes from a respectable source with informed consent from the participants. Besides this, for NLP ambiguity, since we live in a multi-ethnic country, our slangs and abbreviations might be hard to extract, this also applies to industry specific abbreviations that could be misconstrued or overlooked (A. Yadav et. al., 2021). Because of the novelty of this idea, it is important to not overlook industry specific keywords, as such this project's scope will be focused on 'Computer Science' and 'Information Technology' entry roles in Malaysia to minimize ambiguity (Ginter et. al. 2004).

6.4 Objectives Evaluation

To reflect on our objectives, we have gotten an acceptably high accuracy that works in real-time, this addresses our objectives that we had initially when we set out to work on this project. With adequate and real-life data, this system proves to be a model worth considering for further development.

CHAPTER 7

Conclusion and Recommendation

7.1 Conclusion

Throughout this project, we have effectively developed a system that leverages powerful NLP LLM models, like BERT and GPT-3.5, to provide accurate and efficient job recommendations. Though we had issues with sourcing data and ended up starting with a small pre-established labeled dataset, we have innovatively expanded that dataset 2x via data synthesis using the core dataset as the basis for this expansion. Using our synthesized additional data we are able to create a "Matching History", which helps us simulate real world scenarios like having custom weightages in the hiring process.

Our system uses a combination of data preprocessing techniques, word embeddings from Word2Vec, and cosine similarity as our embeddings for the LLM models, these help us enrich the LLMs with better understanding of the semantics and result in more accurate predictions. After thorough testing and evaluations, the results of BERT and GPT-3.5 offer strong performance, despite excelling in different avenues. Namely, BERT demonstrated higher Matching Efficiency Ratios (MER), indicating its efficiency in generating accurate matches in real-time. In contrast, GPT-3.5 showed higher accuracy and consistency in its predictions. Hence, it can be concluded that GPT-3.5 would be the best option to go for, if projects are not limited on funds and are able to accept slower prediction times compared to BERT.

In this project the limitations of the GPT-3.5 API's rates, heavily impacted its overall performance. We also saw prompt engineering playing a crucial role for GPT-3.5 to consistently produce accurate results. To overcome these adversities. Despite the challenges, this project has effectively deepened my understanding of LLMs as a whole.

7.2 Recommendations

Improvements that can be added to this project are as follows:

Bachelor of Computer Science (Honours)
Faculty of Information and Communication Technology (Kampar Campus), UTAR

CHAPTER 7

Increased API Access: Open-source locally ran alternatives or higher end API access to OpenAI could be a potential improvement and recommendation, as from our results, the limitations of the lower end rate staggered development significantly.

Diversify Data Sources: Different industries other than Malaysian Technology companies are worth exploring, as new industries come with their individual semantics and quirks. While in our project we focused on the "Computer Science" and "Information Technology" sectors in Malaysia, an expansion could make the system more versatile and applicable more recruiters and job seekers.

REFERENCES

- [1] A. K. Sinha, et. al., "Resume screening using Natural Language Processing and Machine Learning: A Systematic Review," *Machine Learning and Information Processing*, pp. 207–214, 2021. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-33-4859-2_21
- [2] A. Metsalu, J. Vilo, "ClustVis: a web tool for visualizing clustering of multivariate data using Principal Component Analysis and heatmap," in *Nucleic Acids Research*, vol. 43, no. W1, pp. W566-W570, July 2015. doi: 10.1093/nar/gkv468.
- [3] A. Yadav, et. al., "A comprehensive review on resolving ambiguities in natural language processing," *AI Open*, vol. 2, pp. 85-92, 2021, ISSN 2666-6510, [Online]. Available: <https://doi.org/10.1016/j.aiopen.2021.05.001>.
- [4] B. García, et al., "Automated driver management for Selenium WebDriver," *Empir. Software Eng.*, vol. 26, p. 107, 2021. doi: 10.1007/s10664-021-09975-3.
- [5] C. Daryani, G. S. Chhabra, H. Patel, I. K. Chhabra, and R. Patel, "An automated resume screening system using natural language processing and similarity," *ETHICS AND INFORMATION TECHNOLOGY*, 2020. [Online]. Available: https://web.archive.org/web/20201218183327id_/https://www.intelcomp-design.com/paper/2etit2020/2etit2020-99-103.pdf
- [6] F. Ginter et. al., "Analysis of link grammar on biomedical dependency corpus targeted at protein-protein interactions." *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*. 2004. [Online]. Available: <https://aclanthology.org/W04-1203.pdf>
- [7] J. Schollmeyer, "Using NLP to detect tradeoffs in employee reviews," *SpringerLink*, 2023. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-17288-5_19.
- [8] H. Yang, et. al., "Brand celebrity matching model based on Natural Language Processing," *arXiv.org*, 18 August 2022. [Online]. Available: <https://arxiv.org/abs/2208.08887>.
- [9] P. Nadkarni, et. al., "Natural language processing: an introduction" in *Journal of the American Medical Informatics Association, Volume 18, Issue 5*, September 2011. [Online]. Available: <https://doi.org/10.1136/amiajnl-2011-000464>
- [10] R. Drikvandi and O. Lawal, "Sparse Principal Component Analysis for Natural Language Processing," *Ann. Data. Sci.*, vol. 10, no. 1, pp. 25-41, Feb. 2023, [Online]. Available: <https://doi.org/10.1007/s40745-020-00277-x>.

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 2
Student Name & ID: Wooi Zhuang Ru 1901406	
Supervisor: Dr. Aun Yichiet	
Project Title: Job-Applicant Matchmaking System using Natural Language Processing	

<p>1. WORK DONE [Please write the details of the work done in the last fortnight.]</p> <p>Established project goals and initial scope. Started data collection, obtaining job descriptions and related resumes. Created API access.</p>
<p>2. WORK TO BE DONE</p> <p>Finalize project objectives and scope. Begin data collection and synthesis. Explore how to increase API access with OpenAI.</p>
<p>3. PROBLEMS ENCOUNTERED</p> <p>-</p>
<p>4. SELF EVALUATION OF THE PROGRESS</p> <p>I have been working on my understanding of OpenAI and LLM models</p>



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 4
Student Name & ID: Wooi Zhuang Ru 1901406	
Supervisor: Dr. Aun Yichiet	
Project Title: Job-Applicant Matchmaking System using Natural Language Processing	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Improved data expansion
Successfully added BERT and GPT-3.5 for word embeddings.

2. WORK TO BE DONE

**Use BERT and GPT-3.5 models for word embeddings.
Develop cosine similarity calculations.**

3. PROBLEMS ENCOUNTERED

OpenAI rate limits have been an issue when it comes to development

4. SELF EVALUATION OF THE PROGRESS

I have had issues with adding embeddings to GPT and BERT manually



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 6
Student Name & ID: Wooi Zhuang Ru 1901406	
Supervisor: Dr. Aun Yichiet	
Project Title: Job-Applicant Matchmaking System using Natural Language Processing	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Added in ChromaDB

Implement data privacy and security measures.

Enhanced data security measures.

2. WORK TO BE DONE

Began algorithm fine-tuning, experimenting with weights and metrics.

Continued frontend development.

Conducted initial synthetic data tests.

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

I hope to get the dashboard up in the next update



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 8
Student Name & ID: Wooi Zhuang Ru 1901406	
Supervisor: Dr. Aun Yichiet	
Project Title: Job-Applicant Matchmaking System using Natural Language Processing	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

I have made the word2vec model and gotten a 87% precision rate, where I have manually added flags about which job resumes are the most adequate for an input

I have also done the literature review for both literatures

2. WORK TO BE DONE

Start working on the dashboard that displays and predicts data in real time
Start compiling the FYP2 report

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

I have been ahead in literature review but I have to continue to work on the development side



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 10
Student Name & ID: Wooi Zhuang Ru 1901406	
Supervisor: Dr. Aun Yichiet	
Project Title: Job-Applicant Matchmaking System using Natural Language Processing	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Monitor system performance
Finished Dashboard

2. WORK TO BE DONE

Investigate options for scaling the system.

3. PROBLEMS ENCOUNTERED

Streamlit does not handle absence of OpenAI API key well.

4. SELF EVALUATION OF THE PROGRESS

The project has absence of data clarity/analysis, I should add that



Supervisor's signature



Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 3, 3	Study week no.: 12
Student Name & ID: Wooi Zhuang Ru 1901406	
Supervisor: Dr. Aun Yichiet	
Project Title: Job-Applicant Matchmaking System using Natural Language Processing	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Added analysis and a chat function for the frontend, where you can chat with the app

2. WORK TO BE DONE

Finalize report for submission and prepare for presentation.

3. PROBLEMS ENCOUNTERED

Configuring report and adding information has been hard

4. SELF EVALUATION OF THE PROGRESS

I want to put my best foot forward in the presentation for FYP2




Supervisor's signature



Student's signature

POSTER



BY: Wool ZHUANG RU

COURSE: BACHELOR'S OF COMPUTER SCIENCE

INSTITUTION: UTAR

Job-Applicant Matchmaking System using Natural Language Processing

Objective
To develop a job-applicant matchmaking system using natural language processing that matches job seekers with suitable job openings based on their skills and qualifications.

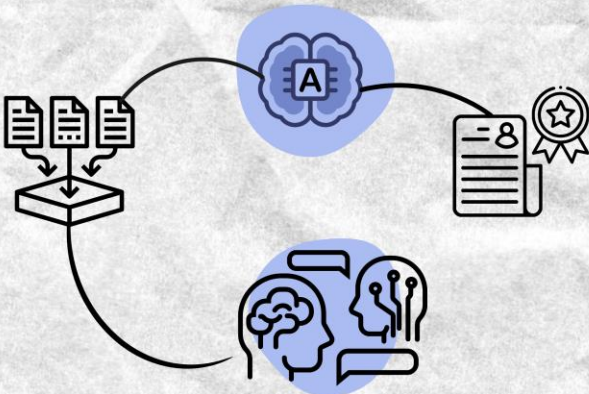
Skills

- Natural Language Processing
- Data Generation
- Machine Learning Algorithms
- Python Programming
- Visualizations and Data Presentation



Talentide



ALLows You To:

- COMMUNICATE WITH YOUR DATA DIRECTLY VIA CHAT
- GAIN INSIGHT AND INFERENCEs FROM YOUR DATA
- CATER MATCHES TO HIRING HISTORY AND COMPANY PREFERENCEs
- PREDICT YOUR JOB MATCHES IN REALTIME





Technologies Used

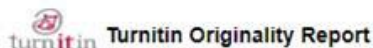



POWERED BY

PLAGIARISM CHECK RESULT



FYP2 by Zhuang Ru Wooi
From FYP Reports (ScanFYP2023)

Processed on 15-Sep-2023 13:04 +08
ID: 2166665504
Word Count: 5699

Similarity Index		Similarity by Source	
6%		Internet Sources:	5%
		Publications:	0%
		Student Papers:	3%

sources:

- 1 1% match (Internet from 28-May-2023)
<https://www.arxiv-vanity.com/papers/2208.08887/>

- 2 1% match (Internet from 18-Jul-2021)
https://fict.utar.edu.my/documents/FYP/IIPSPW_template/IIPSPW_Report_Template_CS.docx

- 3 1% match (Internet from 20-Apr-2023)
https://web.archive.org/web/20201218183327id_/https://www.intelcomp-design.com/paper/2etit2020/2etit2020-99-103.pdf

- 4 < 1% match (student papers from 28-Apr-2023)
[Submitted to Universiti Tunku Abdul Rahman on 2023-04-28](#)

- 5 < 1% match (student papers from 25-Apr-2023)
[Submitted to Universiti Tunku Abdul Rahman on 2023-04-25](#)

- 6 < 1% match (student papers from 27-Apr-2023)
[Submitted to Universiti Tunku Abdul Rahman on 2023-04-27](#)

- 7 < 1% match (student papers from 20-Apr-2022)
[Submitted to Universiti Tunku Abdul Rahman on 2022-04-20](#)

- 8 < 1% match (student papers from 21-Apr-2022)
[Submitted to Universiti Tunku Abdul Rahman on 2022-04-21](#)

- 9 < 1% match (student papers from 19-Apr-2023)
[Submitted to Universiti Tunku Abdul Rahman on 2023-04-19](#)

PLAGIARISM CHECK RESULT

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date:	Page No.: 1 of 1



FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Wooi Zhuang Ru
ID Number(s)	19ACB01406
Programme / Course	Bachelor's of Computer Science (Honours)
Title of Final Year Project	Job-Applicant Matchmaking System using Natural Language Processing

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)
Overall similarity index <u>6</u> % Similarity by source Internet Sources: <u>5</u> % Publications: <u>0</u> % Student Papers: <u>3</u> %	
Number of individual sources listed of more than 3% similarity: <u>0</u>	
Parameters of originality required, and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Signature of Co-Supervisor

Name: Aun Yichiet

Name: _____

Date: 15 Sep 2023

Date: _____



UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	1901406
Student Name	Wooi Zhuang Ru
Supervisor Name	Dr. Aun Yichiet

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Title Page
✓	Signed Report Status Declaration Form
✓	Signed FYP Thesis Submission Form
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Weekly Log
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
✓	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)

Date: 15/09/2023