RESUME RANKING FOR A JOB DESCRIPTION DERIVING SIMILARITY OF REPRESENTATIONAL EMBEDDING USING SENTENCE BERT

ABHISHEK THITE
MS Artificial Intelligence and Machine Learning

Final Thesis Report

MARCH 2022

DEDICATION

To my parents,

for their unconditional love and support and for always inspiring me to be the best version of myself.

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor, university mentor and student mentor for support and guidance in this program

ABSTRACT

With large number of job applicants for a single job these days, its often difficult for hiring managers to manually screen resumes and shortlist most relevant candidates. Screening resumes is often repetitive task due to new inflows and having automated resume ranking system is much needed. This subject has been studied widely; however, we are supporting our methodology by use of pretrained Bidirectional Encoder Representations from Transformers (BERT) for representational embedding of job descriptions and resumes. We further applied cosine similarity on BERT embeddings to rank resumes for given job description. This study produced ROUGE-1 recall score of 0.3172 between ranked resumes and input job description. Score was better than equivalent representation-based methods like BOW model with TF-IDF weighting or word2vec. The experimental procedure shown that BERT based models worked better with new unknown data and produced contextual outcomes aligned to a business goal. Also, models were computationally efficient with representation-based approach compared to interaction-based approaches. The study concluded that BERT representational embedding provides promising settings for document ranking tasks.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BERT..	Bidirectional Encoder Representations from Transformers
BOW..	Bag of Words
CBF..	Collaborative Filtering
CF..	Content Filtering
CLS…	Classification token in BERT (Special significance)
CNN..	Convolution Neural Network
CSS..	Cascading Style Sheet
HR….	Hiring Manager or Human Resource or Hit Rate
HTML..	Hyper Text Markup Language
IDF..	Inter Document Frequency
IT…...	Information Technology
JD…	Job Description
JRS…	Job Recommender System
KB..	Knowledge Based
KNN..	K-Nearest Neighbor
LDA..	Latent Dirichlet Analysis
LSTM…	Long Short Term Memory Architecture
ML..	Machine Learning
NDCG..	Normalised Discounted Cumulative Gain
NER…	Named Entity Recognition
NLP..	Natural Language Processing
NN…	Neural Network
OOP..	Object Oriented Programming
POS..	Part of Speech
RNN..	Recurrent Neural Network
ROUGE..	Recall-Oriented Understudy for Gisting Evaluation
SEP…	Separator token in BERT (Special significance)
STS..	Semantic Textual Similarity
SVC..	Support Vector Classifier
SVM..	Support Vector Machine
TF…	Term Frequency

CHAPTER 1

INTRODUCTION

Advent of internet in late 1980s has reformed operating model of many organizations. Fast growing adaption of technology has caused matching growth of amount of data available online. There was always question on how to effectively manage data with advancement in IT.
Recruitment is one of the most important aspect of development of organization which require effective management of data. Hiring right talent on time has significant effect on organizational performance (Ekwoaba et al., 2015).

There are two stakeholders involved in recruitment process – recruiter and job seeker. Recruitment has two main phases: attraction phase and selection phase (Färber et al., n.d.). Attraction phase involves branding and marketing, posting accurate job description over company site, job forums like linked-in, indeed, monster etc. Job seeker reviews various job descriptions and generally apply for relevant vacancies. Selection phase typically starts with screening of resumes and other submitted documents. This is called pre-screening or preselection phase. The final selection is then carried out with non-filtered candidates from pre-screening phase.

Selection is process of making choice of most suitable applicants from pool of applicants to fill relevant job vacancy (Opatha, 2009). From personal experience, I posted job for front end UI developer in India which attracted ~ 300 applications over a week. Glassdoor study states that every corporate job attracts ~ 250 resumes.

Most companies hire talent reviewing applicants applied to job post on own company web site. Recruitment platforms like linked in provides facility for recruiter to search talent via skills however solutions are often proprietary and not available publicly for use while screening resumes on company site.

Task of manually screening resumes to shortlist for interviews is cumbersome and often repetitive due to continuous inflow. So, it’s imperative to have automated resume screening solutions to shortlist relevant resumes which are matching jobs and focus on shortlisted applicants for manual screening and interviews.

Solutions for matching resumes to jobs are like job recommender systems (JRS) i.e., finding jobs for job seeker resume with some considerations. Researchers have explored various solutions involving NLP techniques like lexical analysis, syntactic analysis, semantic analysis, traditional ML classification, clustering algorithms. Recently researchers are exploring deep learning, attention-based transformers with the aim of improving accuracy, efficiency, scalability of solution. Relevance of this study is to explore Sentence BERT (Reimers and Gurevych, 2019) to create representational embedding of resumes and JD and perform screening based on similarity of embedding. BERT (Devlin et al., 2018) has outperformed in many STS tasks. This research has explored representational embedding power of BERT for resume ranking task.

1.1 Problem Statement

Researchers have used various machine learning approaches for resume screening. We have categorized approaches mainly in two parts – Type 1(standard NLP processing with traditional ML) and Type 2(deep learning-based approaches). In late 2000s, type 1 approaches were dominating research area. Latest studies are more driven by type 2 approaches with recent breakthrough of attention-based transformers in many NLP tasks.

Type 1 approaches include content-based filtering, collaborative filtering, hybrid filtering, knowledge-based recommendation models (Wei et al., 2007). Content based filtering considers semantic nature of terms between resume and JDs during screening (Singh et al., 2010) (Vishruth et al., 2020) (Roy et al., 2020). Lack of common terminology between job seeker and hiring manager is often problem with this approach. Collaborative filtering is based on behavioral data i.e., it considers user-item (job seeker and job) interaction to enhance recommendation (Ahmed et al., 2017). Such approaches generally face cold start problems due to lack of interaction data in initial phase. Hybrid models often combines multiple approaches to take advantages and overcome shortcomings of individual approaches (Liu et al., 2016). In knowledge-based models, one often uses domain knowledge for effective matching (Li et al., 2016).

Type 2 approaches were further classified as interaction based and representation-based models (Lin et al., 2020). In interaction based models, resumes and JD representations interact during NN training process to produce similarity matrix that captures term interaction which further undergoes analysis to calculate relevance score (Maheshwary and Misra, 2018). In representation based models, resumes and JD representations are learnt separately (Ramanath et al., 2018) and compared with simple matrix like cosine similarity.

While we have done detailed literature review in chapter 1, we have covered glimpse of few papers in this section to set context of this research. (Vishruth et al., 2020) used TF-IDF and K nearest clustering algorithm to rank resumes. Such approaches are simpler but inherently lacks semantics in TF-IDF/BOW based models. (Zhang and al Hasan, 2017) created joint embedding representation of resumes/JDs in shared skill and job vector space. (Maheshwary and Misra, 2018) trained Siamese network on labelled data of JD and matching resumes over contrastive loss function. Twin networks are trained with the aim to reduce semantic distance between resumes and relevant JDs and maximize the semantic distance between resumes and irrelevant JDs(Maheshwary and Misra, 2018). Such approach requires NN passes for every combination of JD/available resume to come up with ranking.

(Bhatia et al., 2019) deals with problem of nonstandard resume formats and lack of ground truth for resume ranking against JD. They used BERT (Devlin et al., 2018) for resume standardization by training on Linked- in resume format (standard resume format) and then detected important portions on non Linked-in format. For resume ranking, they trained BERT sentence classification on resumes' prior work experience with positive samples belonging to same candidate and negative sample belonging to different candidates. Eventually trained network is used passing JD and candidate work experience as input to classify match or not. Approach requires diverse resume dataset for training and relies heavily on experience section in resume and do not take skills mentioned in resume into consideration.

Resumes/JDs often have different terms used to represent similar skills. To address this problem, (Van-Duyet et al., 2017) trained NN to derive generalized related skills for given skill. (Gugnani and Misra, n.d.) derived implicit skills with custom model like linear SVC and used for matching resumes. They used affinity score to match job skills vs candidate skills. (Mahdi et al., 2021) used sentence BERT for embedding and Latent Dirichlet Allocation (LDA) for topic modelling to extract skill keywords from job descriptions.

This research has experimented sentence BERT (Reimers and Gurevych, 2019) for representational embedding of JD and resumes and applied cosine similarity on embedding representation to rank resumes for given JD. This approach covered semantics relationship with pretrained BERT and observed to be computationally efficient as there was no need for NN passes for every JD-resume combination. This approach fits in type 2 category of model with representation-based approach explained earlier in section.

1.2 Aim and Objectives

The main aim of this research was to propose use of Sentence BERT (Reimers and Gurevych, 2019) for representational embedding of resumes and JDs and process these embedding via cosine similarity for resume ranking task for given JD.

The research objectives were formulated based on the aim of this study as follows:

- To apply sentence BERT representational embedding on JD/resumes.
- To derive resume ranking with cosine similarity using JD vs resumes embedding.
- To evaluate performance of model using ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, n.d.) between JD and ranked resumes.
- To compare performance of BERT based model with baseline bag of word model with TF-IDF weighing.

1.3 Research Questions

Whether BERT representational embedding is effective tool for downstream resume ranking task for given JD?

1.4 Scope of the Study

Study demonstrated use of BERT as representational embedding for resume ranking task. It explored various ways to handle input sequence length problem in BERT. Due to lack of public datasets for JD and matching resumes, study has performed limited BERT fine tuning in this research and further improvements were kept as future work. Study performed comparative analysis of results with baseline BOW model with TF-IDF weighting and have not focused on achieving highest accuracy on BERT based model for resume ranking task.

1.5 Significance of the Study

Most research papers for resume screening in recruitment have explored traditional content-based filtering, collaborative filtering, hybrid, knowledge-based models and recently, interaction based deep learning models as explained in earlier sections. This study has further enhanced existing body of research by exploring representational embedding approach using BERT for resume ranking task. Representational embedding approach observed to be more computationally efficient for clustering type of tasks during prediction phase. This is because approach don't need NN passes for each JD and resume combination at prediction phase and can simply perform similarity calculation on prestored embeddings. Previous representational embedding approaches for resume scanning explored custom embeddings and have not evaluated BERT (used as representation). Pre-trained models like BERT have revolutionized many NLP tasks since inception.

Another significance of study came from the fact that there are hardly any publicly available recruitment datasets containing JD and respective applied resumes. This may be due to sensitive PII information in resumes which companies often don't release. Previous papers have mostly used accuracy and F1 scores for evaluation over own specific datasets using ground truth of applied resumes. This study uniquely explored use of ROUGE as evaluation matrix for ranking tasks when dataset with JD and respective applied resumes is not available. Such evaluation matrix can extend to multiple domains due to loose coupling.

1.6 Structure of the Study

The structure of thesis is as follows. Chapter 1 presents background of research. Problem statement is discussed in section 1.1. The study aims and objectives are discussed in section 1.2. Section 1.3 states research question. Scope of study is explained in section 1.4. Section 1.5 describes significance of study.

Chapter 2 describes Literature Review. We will start with discussion on need of ML solution for recruitment and characteristics of required solution, under motivation of study in Section 2.2. In Section 2.3, we will cover recruitment process to understand overall context of domain. In Section 2.4, we will discuss various challenges involved in recruitment which are considered while designing ML solutions. Overall objective is to make solution more mature in terms of accuracy, efficiency, scalability. In Section 2.5, we will categorize various literature papers published in domain along with pros, cons of each category. This section will help visualize overall adoption of techniques by researchers and lay groundwork to determine place of current study in overall context. Section 2.6 gives single snapshot of research papers over the years. Section 2.7 summarizes literature review and gaps. We will discuss various datasets and evaluation matrix used by researchers and boil down to hypothesis for approach and evaluation matrix for current study. Section 2.8 gives closing note for this chapter and set path for next chapter.

Chapter 3 describes Research Methodology. We will start with data selection criteria, data attributes and challenges in data selection in section 3.2. In section 3.3, we will show high level picture on overall research approach and give brief explanation. In section 3.4, we will cover pre-processing step and discuss sequence length problem of BERT and ways to handle it for our dataset. In section 3.5, we will discuss parameters for choosing actual embedding model comparing with available options. Section 3.6 covers quick details around selected embedding model from literature to have understanding on selected model. Section 3.7 covers downstream processing on embedding and selection criteria for current research. Section 3.8 covers details on base model used for comparison. Section 3.9 covers evaluation matrix that will be used in research along with reason. Section 3.10 discusses go to architecture for this task and impediment to cover as part of current thesis. We will conclude chapter with summary in section 3.11.

Chapter 4 explains analysis and various experiments conducted in research. We will start with EDA on resumes and job datasets in section 4.2. In Section 4.3, we will summarize various experiments conducted for this research. In Section 4.4, we will discuss resume ranking using sentence BERT and various experiments related to handling sequence length problem in BERT. In section 4.5, we will explain sentence BERT fine tuning experiment using recruitment domain data along with measures to handle sequence length problem. In section 4.6, we will discuss experiment of data cleaning on pre-trained model like BERT. In section 4.7 and 4.8, we will discuss experiments related to baseline models of TF-IDF weighting and word2vec for comparison with sentence BERT.

Chapter 5 discusses results obtained by conducting experiments in chapter 4. We will start with recap of usage of ROUGE matrix for current research and present evaluation of resume ranking with sentence BERT in section 5.2. In section 5.3, we will compare ROUGE matrix for various experiments conducted in last chapter. In section 5.4, we will interpret results of experiments with logic behind outcomes. In section 5.5, we will present impact of data cleaning on BERT embedding and downstream task output.

Chapter 6 provides closing remarks of this study and provides direction for future work. We will start with summarizing research approach, experiments conducted, results obtained in section 6.2. We will discuss contribution to knowledge in the field of JRS or resume ranking tasks in section 6.3. In section 6.4, we will propose future recommendations based on limitation of this study and literature review performed leading to this study.

CHAPTER 2

LITERATURE REVIEW

Advent of internet in late 1980s has reformed operating models of many organisations. During Industry 3.0, there was always question on how to use IT technology to improve operating models of organisations. Recruitment was no exception. In 1990, system was developed for newspaper to solve problem of ever-increasing advertisements in printed paper. System used to compare job offers in advertisement data base with skills entered by requester on their terminal (Minitel) (Semantic matching between job offers and job search requests, n.d.) and used to recommend matching jobs.

After 30 years since this initial contribution, concept of matching resumes to jobs or vice-versa is vaguely similar though there have been significant variations in terms of using different approaches, algorithms and techniques applied to improve efficiency, scalability, accuracy of solution. In this chapter, we will discuss various approaches in detail.

2.1 Introduction

In this chapter, we will start with discussion on need of ML solution for recruitment and characteristics of required solution under motivation of study in Section 2.2. In Section 2.3, we will cover recruitment process to understand overall context of domain. In Section 2.4, we will discuss various challenges involved in recruitment which are considered while designing ML solutions. Overall objective is to make solution more mature in terms of accuracy, efficiency, scalability. In Section 2.5, we will categorize various literature papers published in domain along with pros, cons of each category. This section will help visualize overall adoption of techniques by researchers and lay groundwork to determine place of current study in overall context. Section 2.6 gives single snapshot of research papers over the years. Section 2.7 summarizes literature review and gaps. We will discuss various datasets and evaluation matrix used by researchers and boil down to hypothesis for approach and evaluation matrix for current study. Section 2.8 gives closing note for this chapter and set path for next chapter.

2.2 Motivation for Study

Fast growing adaption of technology have caused matching growth of amount of data available online. It has also raised requirement for ability of users to manage this data efficiently. Main job of information filtering solutions is to filter unwanted data based on user query (Hanani et al., 2001).

Recommender systems are designed addressing this management of information overload and originates from cognitive science, approximation theory, information retrieval, forecasting theories, management science, consumer choice modelling in marketing (Adomavicius and Tuzhilin, 2005). Recommender systems basically match user profile and/or preferences with item specification and perform recommendation. Recommendation systems are implemented in various industries e.g., retail, media, banking, e-commerce and even in recruitment.

In recruitment industry, recommendation is often used in handling candidates' application, pre-screening of candidates. However underlying criteria for matching are often hard to measure. Most system perform binary match where resumes and job description are matched based on common terms between two. A few challenges with approach are – a) skills mention in different form or synonyms (e.g., cplusplus vs c++; programming vs scripting) b) There could be skills not specified in resumes or JDs but can easily be determined by domain knowledge (e.g., Java being object-oriented language (OOP), person knowing Java also knows OOP concepts) c) A skill could be out of dictionary skill or from new domain for which system may not have skills (Gugnani and Misra, n.d.).

For such cases, binary match system would not provide best person-job fit. So (Gugnani and Misra, n.d.) derived implicit skills with custom model training using large training dataset and word2vec. However, such approaches still need evaluation across domains, varied datasets.
Another consideration is JD often look for multiple skills in context for ideal match e.g., web developer often needs to have HTML and JavaScript/CSS skills together for match. Word2vec embedding is applied on individual words so often not suitable when longer sequence in context to be matched. So, there is need to explore more matured embeddings which will take context into account.

Another motivation for study comes from fact that most companies hire talent reviewing applicants applied to job post on own company web site. Recruitment platforms like linked-in provides facility to recruiter to search talent via skills however solutions are often proprietary and not available publicly for use while screening resumes on company site.

So, it makes sense to explore solution, which is robust to words mismatch, works for longer sequence in context, open source and pre-trained for easy adoption.

2.3 The Recruitment Process

Recruitment is one of the most important aspect of development of organization. Hiring right talent on time has significant effect on organizational performance (Ekwoaba et al., 2015).
There are two stakeholders involved in recruitment process – recruiter and job seeker. Recruiter creates job description mentioning expected skills required for role. Job seeker creates resumes mentioning own expertise and experience along with other educational background (Färber et al., n.d.). The IT support for recruitment ranges from attracting talent, selecting talent to retaining talent (Armstrong et al., 2011a). Degree of process integration represents complexity of using e-recruitment solutions (Malinowski et al., 2005).

Recruitment has two main phases: attraction phase and selection phase (Färber et al., n.d.). Attraction phase involves branding and marketing, posting accurate job description. Selection phase typically starts with screening of resumes and other submitted documents. This is called pre-screening or preselection phase. The final selection is then carried out with non-filtered candidates from pre-screening phase.

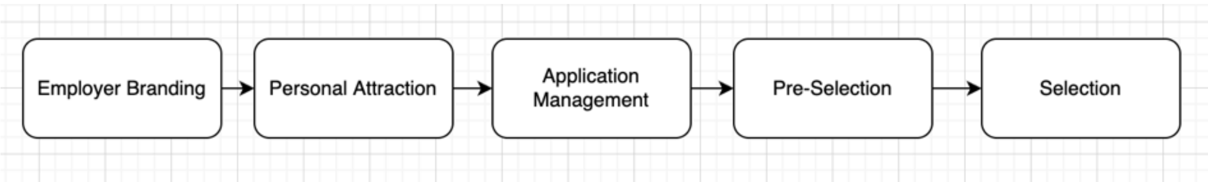


Figure 2.1. Recruitment Process

Figure 2.1 shows recruitment process which is adapted from (Armstrong et al., 2011b). It consists of five main tasks – short and long-term candidate attraction, application management, pre-selection followed by final selection of candidates. Application management includes communication with candidates, administration of applications, forwarding application to hiring managers, internal processes (Färber et al., n.d.). Each task has IT solutions for automation. This study mainly focusses on solutions for pre-selection and selection phase.

2.4 Challenges in Recruitment

Several challenges related to selection phase of recruitment process are summarized in this section from literature review. (Mahdi et al., 2021) have discussed problem of search engine optimization (SEO). Due to inconsistent terminology used by candidates and hiring manager or HRs, there is lack of correct keywords in job descriptions reducing chance of getting higher ranking for jobs. (Amin et al., 2019) have discussed problem of time complexity in matching resume against jobs due to large candidates applicable for job. (Vishruth et al., 2020) have discussed hiring complexity in manual screening.

(Dave et al., 2018) have discussed need of job and skill recommendation together. (Van-Duyet et al., 2017) have discussed need of determining related skills for better matching e.g., Java is object-oriented programming (OOP) language so job profile looking for OOP concept can be recommended with Java resumes. (Bhatia et al., 2019) have discussed problem of non-standard resume formats which increases complexity in resume parsing. They also mentioned problem of lack of ground truth for resume ranking task across multiple domains. (Maheshwary and Misra, 2018) have discussed challenge of lack of semantics in TF-IDF based bag of word (BOW) model. (Gugnani and Misra, n.d.) have discussed problem of lack of standard terms in resumes causing complexity in matching task.

(de Ruijt and Bhulai, 2021) have discussed lack of consideration of temporal and reciprocal attributes in most recommendation solutions. Temporal attribute suggests using lead time of vacancy during recommendation. Reciprocal attribute suggests reordering job recommendation based on response because job seeker tend to apply less popular jobs with relevant skills. (Lee and Brusilovsky, 2007; Wei et al., 2007) have discussed need of considering user interaction with job platform and learnt user preferences during job recommendation. (T. Al-Otaibi, 2012) have discussed common challenges in collaborative filtering (CF), content based (CBF), hybrid, knowledge based JRS. CF face cold start problem due to lack of historic data. CBF face ramp up problems. Knowledge based filter requires domain knowledge.

2.5 Literature Review in Recruitment

ML Research papers in recruitment domain can be found from late 2000. Researchers have used variety of approaches to solve challenges described in section 2.4. Earlier approaches have focussed more on standard NLP techniques followed by traditional ML modelling, content-based filtering, collaborative filtering, hybrid recommendation models while recent ones heavily use deep learning capabilities.

On high level, ML solutions for recruiting are classified into traditional ML with NLP and deep learning based models.

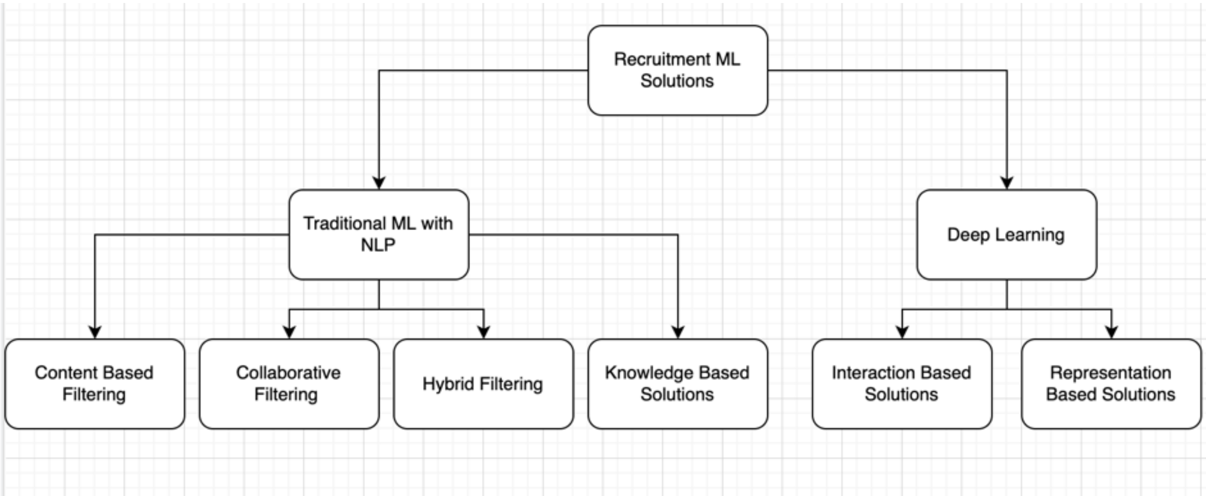


Figure 2.2. Types of Approaches

Traditional ML with NLP approach is further divided into content based, collaborative, hybrid, knowledge-based solutions.

Deep-learning based solutions are classified as interaction-based vs representation based. In Interaction-based approach, resumes and JD representations interact during NN training process to produce similarity matrix that captures term interaction which further undergoes analysis to calculate relevance score. Representation based approach trains NN to create embeddings independently for resume and JD and then uses similarity measures like cosine similarity to come up with ranking.

2.5.1 Content based Filtering (CBF) based JRS

Content based filtering models in context of JRS are models which take semantic consideration into picture while matching resumes to JDs. In CBF, one creates vector representations of resumes and JDs in unsupervised way i.e., dimensions of representation may not have intuitive interpretation. These types of models often use bag of word-based models with TF-IDF weighting, clustering algorithm like latent Dirichlet allocation or recently word2vec representation.

(Vishruth et al., 2020) have used bag of word with TF-IDF weighing to represent resume and JDs in common vector space and then used K nearest clustering algorithm to rank nearest resumes. (Amin et al., 2019) have used NER techniques to parse resumes and JDs and calculated relevance with custom formula for resume vs JD matching.

(Roy et al., 2020) have used standard NLP techniques like stop word removal, stemming, lemmatization etc and applied Gensim to summarize JD and resumes. It further used classifier for training to categorize resumes in high level categories. It further ranked resumes using collective weighting of content-based recommendation, cosine similarity, KNN to identify resumes that are nearest to provided JD. (Gugnani and Misra, n.d.) have used NER, POS tagging, word2vec to derive implicit skills using custom model like linear SVC and further used those implicit skills for matching task to improve ranking. Affinity scores were used to match jobs skills with resume skills.

(Lin et al., 2016) have used unsupervised feature extraction from resumes and JDs and applied supervised ensemble approaches like boosting for resume ranking. (Singh et al., 2010) have suggested system that extract information to provide more fine-tuned retrieval of resume match. It has used conditional random fields, NER to come up with filters which helped faster ranking e.g., resume matching with criteria like *at least 3 years of experience in java*.

(Mostafa and Beshir, 2021) have presented job candidate ranking model that calculated interviewer’s sentiments using basic NLP and naïve bayes, SVM classifier to assist HR managers make hiring decision.

2.5.2 Collaborative based Filtering (CF) based JRS

Collaborative filtering is based on behavioural data, typically stored in a user * item matrix format. In recruitment, like e-commerce setting, this matrix is usually filled with click behaviour (e.g., if job seeker clicks job to see details, then corresponding matrix entry will be 1 else 0). If interaction data is missing, then previous job history from resumes can be used to fill rating matrix.

CF is further divided into memory-based CF and model-based CF. In memory-based CF, recommendations are created using KNN type of approach. In model-based approach, missing matrix values are filled with regression like model. (Ahmed et al., 2017) created model using user, job interaction data which calculated user-user, item-item similarities for making recommendations.

2.5.3 Hybrid Recommendation based JRS

Hybrid recommendation combines multiple approaches into one recommender system. E.g., approach can be to use content-based filtering for semantic matching and collaborative filtering which considers job seeker, job post interaction during recommendation. (Liu et al., 2016) have exploited properties in user-item interaction to improve results.

2.5.4 Knowledge based (KB) JRS

KB is recommender system which relies on deep knowledge about product domain (job) to define best items (vacancies) to recommend to a user (Freire and de Castro, 2021). In such job recommender system, resumes and jobs are mapped to predefined job ontology after which two are matched.

Linked in have used this approach where it nudges user to complete profile with skills section and used this information while recommending jobs (Li et al., 2016). Such approaches simplify keyword-based engines and filtering.

2.5.5 Interaction Deep Learning based JRS

In this category, resumes and JDs are fed to neural network and training is done against ground truth of resumes vs JD matching. NN learns resume and JD mapping by interaction of two during training process.

(Maheshwary and Misra, 2018) have trained Siamese network where JD and resumes' doc2vec representations are passed as input to twin network. Network was trained with pair of matching/non- matching JD and resume combinations over contrastive loss function.

(Bhatia et al., 2019) have dealt with problem of nonstandard resume formats and lack of ground truth for resume ranking against JD. They have used BERT (Devlin et al., 2018) for resume standardization by training on Linked- in resume format (standard resume format) and then detected important portions on non-Linked-in format. For resume ranking, they have trained BERT sentence classification on resumes' prior work experience with positive samples belonging to same candidate and negative sample belonging to different candidates. Eventually trained network is used passing JD and candidate work experience as input to classify match or not.

(Van-Duyet et al., 2017) have extracted skills from JD to train skip gram model assuming skills in single JD are related to each other. So, for new JD, one can get related skills which can further be used for better resume matching.

(Luo et al., 2019) have used ELMo embedding to map all words/phrases in talent's resume and recruiter's job post into continuous representation. Bi LSTM used to learn aggregated representation for resume experiences. The high-level representations of resume skills and job posts were obtained by applying attention scheme and CNN for text processing. They have used adversarial network to improve job representation. Last step was to train network using resume, job post representation product over logistic loss function.

(Zhao et al., 2021) have implemented two stage embedding based recommender system for job to candidate matching. In first stage, custom NN was trained to create embedding representation. It was used to rank candidates at first level. In second stage, extra features were considered like location matching, minimum years of experience, education level to produce final combined ranking.

(Qin et al., 2018) have created ability aware person job fit neural network. In first step, RNN was used to project words of job posting and resumes onto latent representations. In second stage, hierarchal ability was learnt for JD and resumes using attention strategies. Overall solution not only identified matching resumes for job but also able to distinguish which JD requirements are met by resumes.

(Bian et al., n.d.) have proposed deep global match network for capturing global semantic interactions between two sentences from job posting and resume respectively. Paper have claimed cross domain applicability based on sentence and global sense match.

(Le et al., 2019) have trained two NN for bidirectional matching of resumes and jobs. (Nigam et al., n.d.) have leveraged progression of job selection by candidates using Bi-LSTM to recommend future jobs. (Yan et al., 2019) have used memory module to enrich representations of job and resumes based on previous preferences. (Bian et al., 2020) have used text-based match using BERT for sentences followed by transformer attention for complete resume/JDs. It further applied relational match for improvement.

2.5.6 Representation based Deep Learning JRS

In this category, NN are trained to produce representational embedding of resumes and JD. These embeddings are further compared using cosine, hadamard product or any similarity matrix to derive ranking.

(Mahdi et al., 2021) have used BERT representational embedding to extract topic which can be used for applications like search engine optimizations. (Dave et al., 2018) have extracted job sequence from resumes and skills from JDs. They have further trained NN against custom loss function to come up with job transition network, job-skill network, skill co-occurrence network using representational embedding of job and skills.

(Ramanath et al., 2018) have created custom embedding to represent resumes followed by shallow, deep NN training for resume ranking. Described model was used in linked in talent search.

2.6 Research Papers over the Years

As shown in figure 2.3, earlier papers are more driven by NLP, traditional ML, content, collaborative filtering. Researchers have explored hybrid deep learning based approaches in recent years.

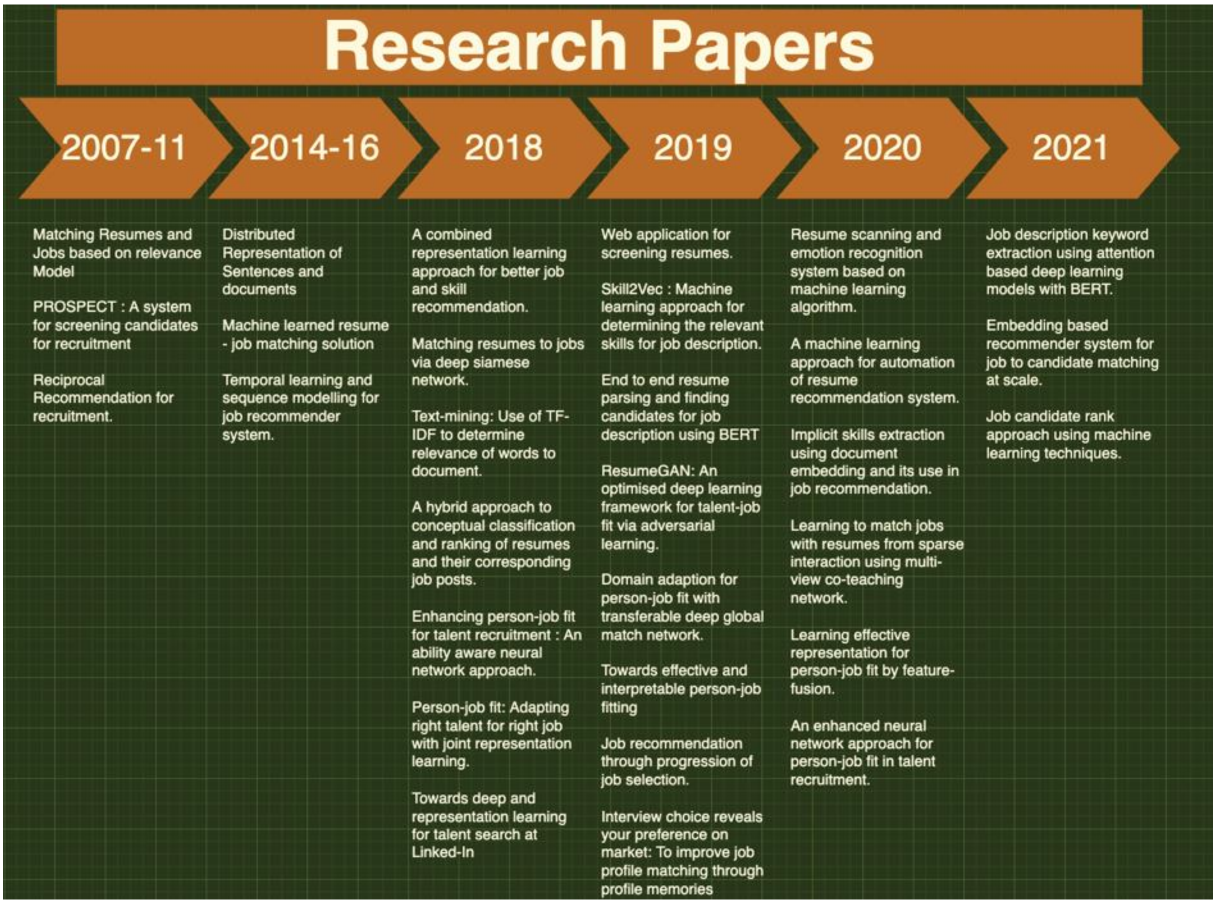


Figure 2.3. Research Papers Timeline

2.7 Discussion

As we discussed in this chapter, researchers till now have used term-based match between resume and JD, semantic considerations, user-item interaction data, historical interactions, previous experiences in resumes. Researchers have also used standard skill terms usage in JD for better search engine ranking, implicit skill extraction for improvement in matching. They further explored geolocation factors of candidate/job, screening based on mandatory skills vs optional skills in JD, matching using specific criteria in JD (min 3 years exp.). Researchers have also dealt with temporal and reciprocal attributes to improve recommendation.

Most work using deep networks is based on interaction-based models. As discussed in section 2.2 related to motivation for this study, there is room to explore solution, which is robust to words mismatch, takes semantics into consideration, works for longer sequence in context, open source and pre-trained for easy adoption and domain independent. So, we explored BERT representation embedding for this research. BERT expected to consider semantics, works with longer sequence in context, open source, pre-trained on diverse large dataset and hence expected to be general/domain independent. Recent survey papers on BERT motivates research in representational aspect of BERT embedding (Lin et al., 2020).

Dataset is another challenge while doing research in this area. While standalone resume and jobs dataset is easily available but dataset with interaction data is less frequently available.
Most researchers seem to have used datasets from competitions in particular CareerBuilder 2012 (Job Recommendation Challenge | Kaggle, 2021), RecSys 2016 (Abel et al., 2016) and RecSys 2017 (Zheng et al., n.d.). However, these datasets do not have resumes and JD in natural language format but rather they are in encoded format. Few researchers have used experiences in resume as proxy for JD for training (Roy et al., 2020).

Evaluation matrix which are used across papers can be divided in two parts. When ground truth of matching resumes for JD is known, confusion matrix, accuracy, recall, F1 score is used to fine tune network during training (Mahdi et al., 2021). In some cases, even ranking matrix are used like HR@10 (hit rate), Normalised Discounted Cumulative Gain NDCG@10 (Roy et al., 2020). When ground truth is not known, motivation from abstractive text summarization survey paper (Syed et al., 2021) can be taken to explore ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, n.d.) between JD and ranked resumes.

2.8 Summary

Like many industries, machine learning has changed the way organization perform recruiting and hire right talent. In this study, we have focussed on resume screening phase of recruiting. Over the years, researchers have applied various techniques ranging from NLP processing combined with traditional ML algorithms in early days to recent solutions which heavily use deep learning. In the process, researchers have addressed common challenges in recruiting which were discussed in this chapter.

Based on literature review of research progress over the years and success of BERT in many Semantic Textual Similarity tasks (STS), we have explored BERT representational embedding for recruiting task in this study.

CHAPTER 3

RESEARCH METHODOLOGY

As concluded in last chapter, our research methodology has revolved around Bi-directional encoder representation of transformer (BERT). Since inception of BERT by (Devlin et al., 2018), BERT has achieved state of the art performance on most STS tasks.

3.1 Introduction

In this chapter, we will start with data selection criteria, attributes, and challenges in data selection in section 3.2. In section 3.3, we will show high level picture on overall research approach and give brief explanation. In section 3.4, we will cover pre-processing step and discuss sequence length problem of BERT and ways to handle it for our dataset. In section 3.5, we will discuss parameters for choosing actual embedding model comparing with available options. Section 3.6 covers quick details around selected embedding model from literature to have clear understanding of selected model. Section 3.7 covers downstream processing on embedding and selection criteria for current research. Section 3.8 covers details on base model used for comparison. Section 3.9 covers evaluation matrix that will be used in research along with reason. Section 3.10 discusses go to architecture for this task and impediment to cover as part of current thesis. We will conclude chapter with summary in section 3.11.

3.2 Data Selection

Dataset is challenge while doing research in recruiting. Most researchers have used datasets from competitions in particular CareerBuilder 2012 (Job Recommendation Challenge | Kaggle, 2021), RecSys 2016 (Abel et al., 2016) and RecSys 2017 (Zheng et al., n.d.).

For our research question, we required dataset with collection of JDs and respective applied resumes in natural language format (since is BERT trained on Wikipedia text) along with ranking of resumes. However, such datasets are not available in public domain. It may be due to kind of PII data, resumes often carry. Datasets used in literature review e.g., RecSys 2017, contains encoded user/job data. They do not have resumes/JD in natural language hence cannot be used for current task.

To overcome this problem, we decided to use two separate datasets available on Kaggle for resume and JD. Job dataset is scrapped from the indeed website (Indeed Software Engineer Job Dataset | Kaggle, 2022). Similarly resume dataset is available on Kaggle (Resume Dataset | Kaggle, 2022). These datasets are having data from similar domains (engineering) and covers variety of roles within domain. So suitable for current research.

Job dataset has 10K entries and has following attributes: Job profile name, company, city, ratings, summary (JD), date. We have used summary column for our analysis which carries job description in natural language format. Maximum number of words in summary column of job dataset is 431.

Resume dataset has 962 entries and has following attributes: category and resume. We have used resume column for our analysis which carries candidate skills, experience, education etc. in natural language format. Maximum number of words in resume column is 2069.

Goal of our study is proposed approach should rank relevant resumes higher for given JD. E.g., for machine learning engineer job description, resume with python, algorithm, mathematics, programming attributes should rank higher than software testing skill resume. Later in this chapter, we will have relevant evaluation matrix defined to measure this.

3.3 Research Approach

In this section, we will cover overall approach for this research.

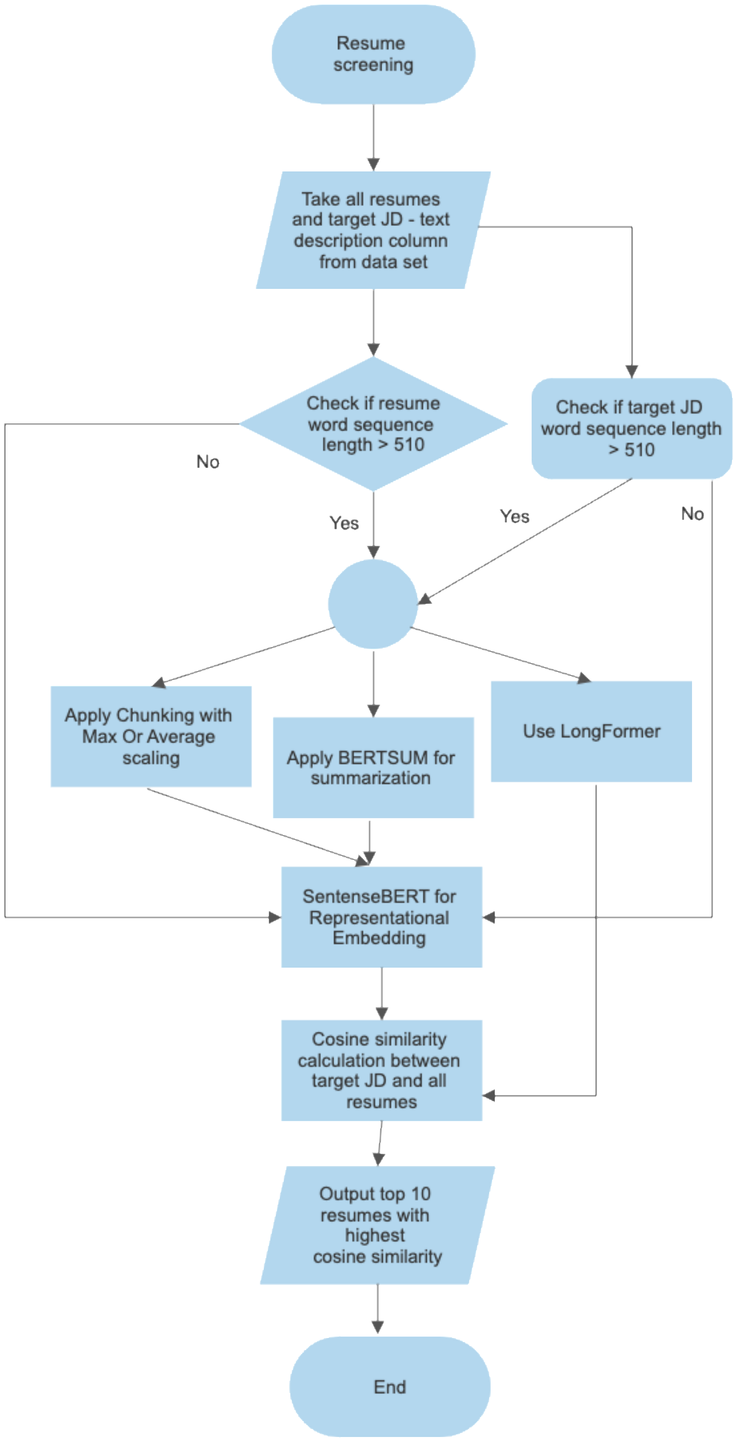


Figure 3.1. Research Approach Flowchart

Figure 3.1 is complete flowchart of research steps. We will first show quick picture of complete process and cover research choices in upcoming sections.

- We have started with reading target JD and resume dataset focussing on summary column in JD and resume column in resume csv.
- We have checked length of word sequence for each input and check if its greater than 510. BERT pretrained model has limitation of sequence length of 512 tokens including 2 special tokens. This limitation has its root in self attention mechanism in transformer where input sequence length quadratically increases number of parameters, memory footprint. More details in upcoming sections.
- If sequence length is less than 510 then directly go to embedding part. If length is greater than 510 then we are required to explore below options for best results
    - Apply chunking i.e., divide input sequence in chunk of 510 and go to embedding step. Take average or max of embedding of each chunk to come up with final representational embedding for that resume or JD
    - Apply summarization technique – BERTSUM (Liu and Lapata, 2019) which brings sequence length to less than 510. Details discussed in upcoming section.
    - Use Longformer (Beltagy et al., 2020) for embedding which is capable of handling long sequence but still work in progress model (Longformer, 2022).
- Once we have sequence of length 510 or less, we applied SentenseBERT (Reimers and Gurevych, 2019) to get representational embedding of resume and JD. Original BERT model is trained on masked language modelling and next sentence prediction. It requires pair of sentences to be passed as input and less suitable in clustering type of setting (Reimers and Gurevych, 2019). So, researchers came up with SentenceBERT for computational efficiency. Details discussed in upcoming section.
- Once we have representational embedding of all resumes and target JD, we took cosine similarity of combinations.
- We sorted cosine similarity descending to come up with ranking of resumes for given JD.
- We have calculated ROUGE (Lin, n.d.) for target JD and top ten resumes and compared with ranking from base model (covered in section 3.8) to check effectiveness of proposed approach.

3.4 Handling Sequence Length Problem in BERT

BERT model is based on transformer architecture (Vaswani et al., 2017) with main highlight of self-attention mechanism for input sequence. Due to this, every token in input sequence need to keep contextuality with every other token in same sequence. Hence number of parameters and memory footprint increases quadratically as input sequence length increases. Due to this, researchers of BERT have restricted input sequence length to 512 tokens while pretraining model (Devlin et al., 2018). Any sequence greater than 512 fed to BERT for embedding will get trimmed to 512 and processed, thus may incur loss of information.

Since resume and JD have sequence length greater than 512 like in our dataset, we are required to handle sequence length problem. As per literature review, researchers have used four different approaches on high level i.e., concatenating existing BERT, constructing long transformer using sequential transformer, constructing long transformer with sparse attention matrix, summarizing document to handle sequence with larger sequence.

3.4.1 Concatenating Existing BERT

If a sequence length is 2000 words long, one can concatenate the sentence representation of 4x BERT passes through each piece of text. One can perform averaging or max scaling or put RNN, LSTM on top of sequence to come up with document level embedding (Mulyar et al., 2019).

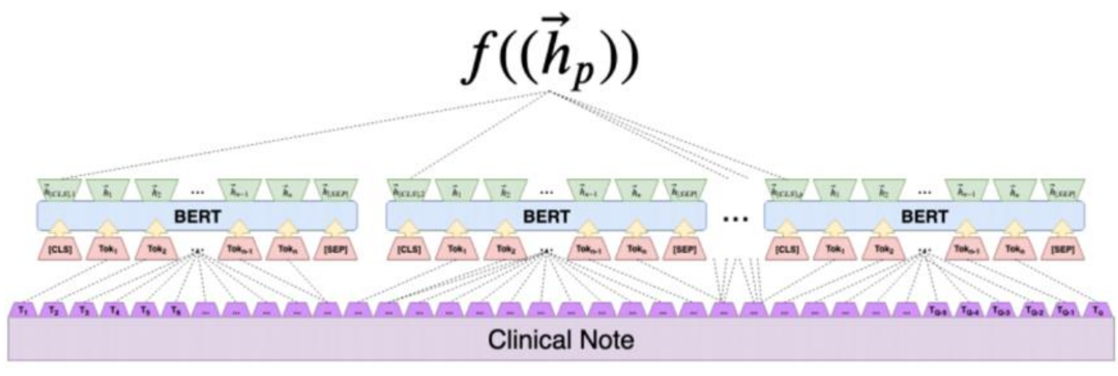


Figure 3.2. (Mulyar et al., 2019)

3.4.2 Using Sequential Transformer

This approach involves constructing a segment level recurrence mechanism over the move of fixed limited window context (of 512 max tokens) token by token from left to right on the long sequence (Dai et al., 2019). This approach may have limitation for transferred tasks which benefit from bi-directional context (because of left to right processing). Also, there is no pretrained model available for transfer learning task like the one required in this research (Beltagy et al., 2020).

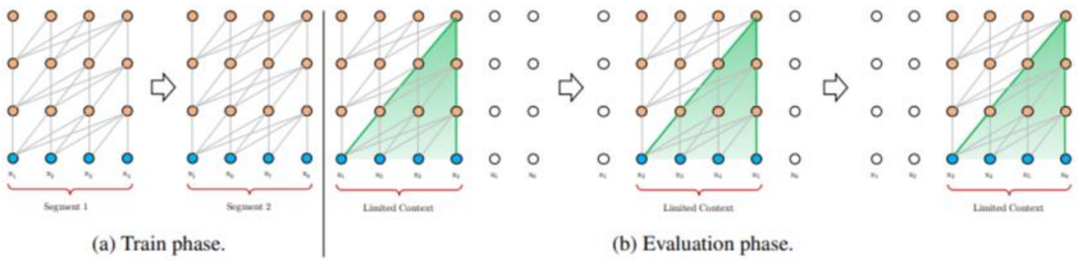


Figure 3.3. (Dai et al., 2019)

3.4.3 Using Sparse Attention Matrix

In this approach, rather than having full self-attention in base model, one chooses to have sparse attention to relax number of computations to scale of sequence length without compromising power of transformer (Beltagy et al., 2020). This is done by local windowed and global attention like below

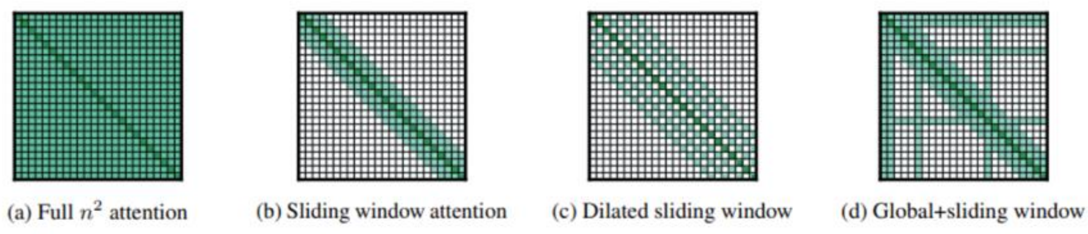


Figure 3.4. (Beltagy et al., 2020)

3.4.4 Summarizing Document

In this approach, sequence is passed through summarization module to reduce sequence length below 512. Since we are lacking dataset of resumes and corresponding summary to train custom NN, we have explored published state of art models for summarization to keep focus on research question. Out of various options available for summarization, we chose to use state of art BERTSUM (Liu and Lapata, 2019) model which is trained on CNN/daily mail dataset (NLP-progress/summarization.md at master · sebastianruder/NLP-progress, 2022). Summarization approach would have information loss and need to evaluate for effectiveness.

3.4.5 Target Approach

Since resume ranking task is transferred learning tasks for this research, we have not evaluated option 2. Sparse matrix approach, though promising, has challenges in clustering type of scenario as explained in next section. So, we have explored max or average scaling over concatenated BERT embedding and summarization for best results.

3.5 Choosing Correct Model for Embedding

To evaluate representational embedding power of BERT, we need network that produces embedding given a resume or JD. BERT has set state of art performance for many STS tasks. However, it requires pair of sentences to be passed as input because base model is pretrained with mask language modelling and next sentence prediction taking two sequences as input.

For sentence clustering tasks, this requirement results in massive computational overhead. Finding the most similar pair in collection of 10000 sentences requires 50 million inference computation (~65 hrs) with BERT (Reimers and Gurevych, 2019). For resume ranking task, we would need 462241 operations to find most similar resume pair out of total 962 resumes in dataset. Other approach is to pass given JD and each of resume to BERT and use sequence pair classification score at prediction time. This approach will need 962 (~number of resumes in system which may grow) passes and would impact model response time during prediction.

To address this aspect of BERT, (Reimers and Gurevych, 2019) came up with sentence BERT which takes single sequence and produces representational embedding vector. We have converted resumes to embedding representation during development phase and stored. During prediction time, we have compared representational embedding of input JD to resume representations. In effect, we would only need one BERT pass for input JD compared to 962 (~

number of resumes in system) in base BERT approach. So, we have used all-MiniLM-L6-v2 model from Hugging Face (Pretrained Models — Sentence-Transformers documentation, 2021) which provides good embedding quality and works faster. LongFormer for handling longer sequence currently behaves like plain BERT and hence require two sequence input so we will keep its comparison for future work.

3.6 Overview of Sentence BERT

In this section, we will quickly review sentence BERT architecture. Neural network is trained with Siamese and triplet network by pooling output of BERT.

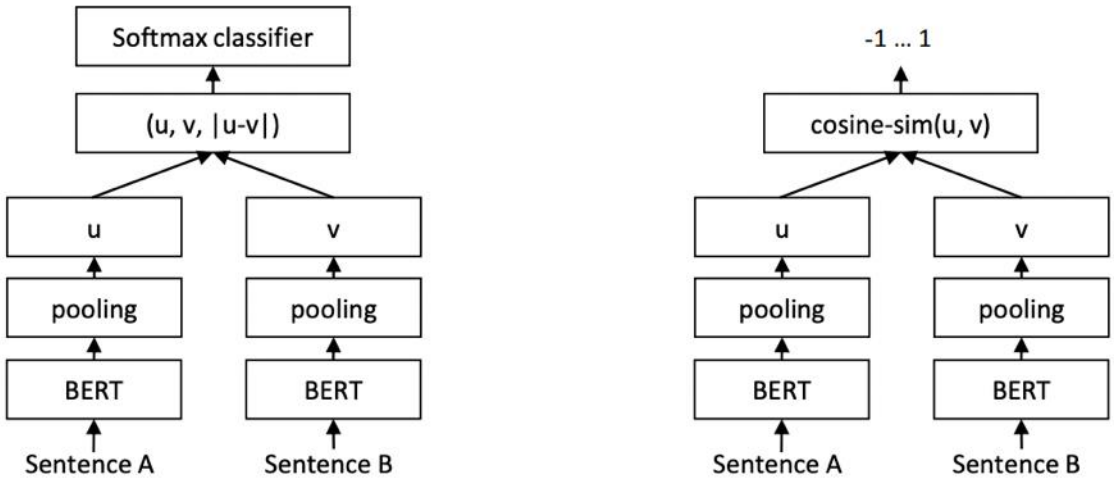


Figure 3.5. (Reimers and Gurevych, 2019)

Figure 3.5 shows architecture for classification and inference scheme. Output of pooling vector is used as representational vector for sentence once network is trained.

3.7 Choosing Downstream Model

To get overall gain with this approach, we have kept downstream model simple. There are couple of ways in which vectors are compared i.e., Jaccard distance, Cosine similarity, Euclidian distance. Jaccard distance compares common terms between documents, cosine similarity calculates angular distance between two documents, Euclidian distance calculates distance between two points (documents) in space.

We chose cosine similarity to make maximum use of contextuality. Representation in continuous vector space helped in covering semantic relationship on top BERT's inbuilt semantic considerations. Once we have embeddings for input JD and stored resumes, we have taken cosine similarity to identify distance between vectors. Distance between vectors indicated difference between actual JD and resume contents including context. We have used SciPy in build function to calculate cosine similarity and sort values descending to come up with top ten resumes for input JD.

3.8 Comparison with Base Model

To check effectiveness of BERT representational embedding for resume ranking task, we would need to compare it with base model using same downstream processing. We have used bag of word model with TF-IDF weighting. Steps are mentioned in below flowchart.

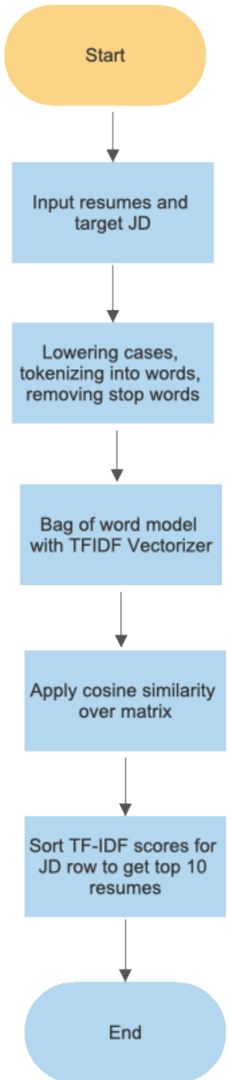


Figure 3.6. Base Model

3.9 Evaluation Matrix

Due to lack of ground truth in dataset, we have checked effectiveness of approach using matrix and manual way. Researchers have tried various approaches as explained in Literature review chapter here.

For matrix approach, we have used ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin, n.d.) as evaluation matrix. ROUGE compared gist of shortlisted resumes with input JD and produced score. ROUGE evaluation helped compare BERT based model against base TF-IDF models. Motivation for ROUGE comes from abstractive text summarization survey paper (Syed et al., 2021)

For manual approach, we have engaged domain SME from engineering field to validate matching of input JD with shortlisted resumes.

3.10 Discussion

Ideally, we should have collected domain data for JD and matching resumes. Further we should have fine-tuned sentence BERT with domain data for best results. As of now, we have leveraged resume dataset to create matching resume pairs and fine-tuned sentence BERT. We have explained this experiment in section 4.5 in detail and presented results in chapter 5. There is scope for improvement by collecting real data and further fine-tuning sentence BERT as shown in figure 3.7. We have checked performance with limited fine tuning due to time constraint of program and compared results with base model.

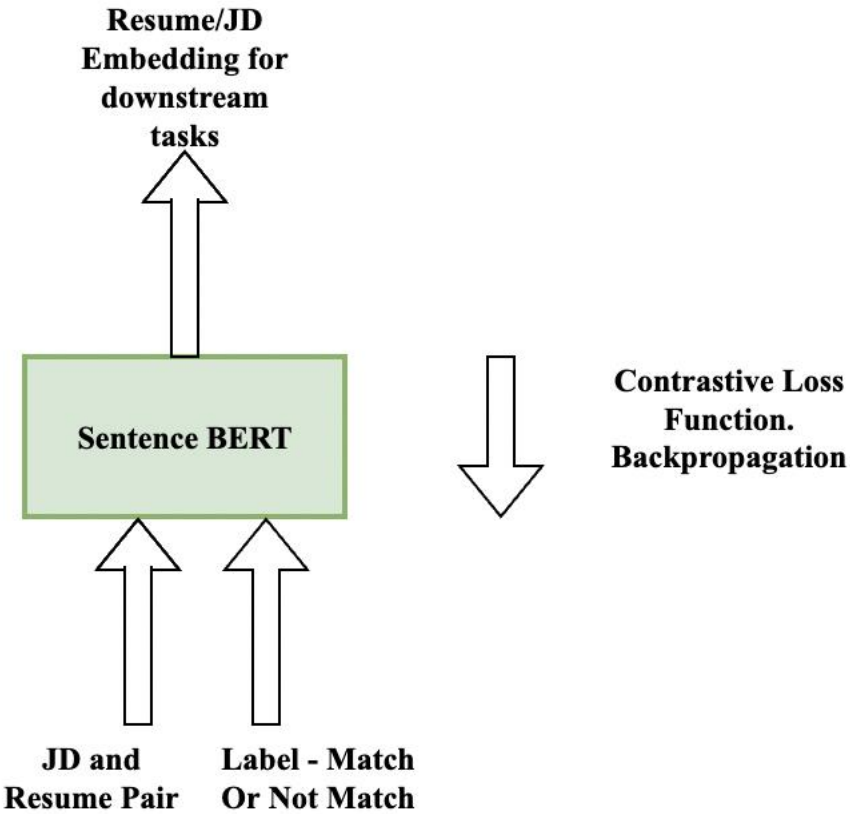


Figure 3.7. End State Approach

3.11 Summary

In this chapter, we explored research methodology including dataset selection criteria. We have decided to use sentence BERT for representational embedding and agreed to apply cosine similarity to derive ranking. We have also explained ways to handle sequence length problem due to large input size of current dataset. We will present experiments conducted in research in chapter 4. We will present comparison of results with base model using ROUGE evaluation in chapter 5.

CHAPTER 4

ANALYSIS AND EXPERIMENTS

This chapter presents findings from EDA on chosen datasets and explains experiments carried out on datasets aligning to research approach described in section 3.3

4.1 Introduction

In this chapter, we will start with EDA on resumes and job datasets in section 4.2. In Section 4.3, we will summarize various experiments conducted for this research. In Section 4.4, we will discuss resume ranking using sentence BERT and various experiments related to handling sequence length problem in BERT. In section 4.5, we will explain sentence BERT fine tuning experiment using recruitment domain data along with measures to handle sequence length problem. In section 4.6, we will discuss experiment of data cleaning on pre-trained model like BERT. In section 4.7 and 4.8, we will discuss experiments related to baseline models of TF-IDF weighting and word2vec for comparison with sentence BERT.

4.2 Data Analysis

In this thesis, we have used resume dataset and job dataset.

4.2.1 EDA on Resume Dataset

Resume dataset contained 962 resumes. Each resume contained resume category and resume description. Maximum word sequence length in any resume was 2069.

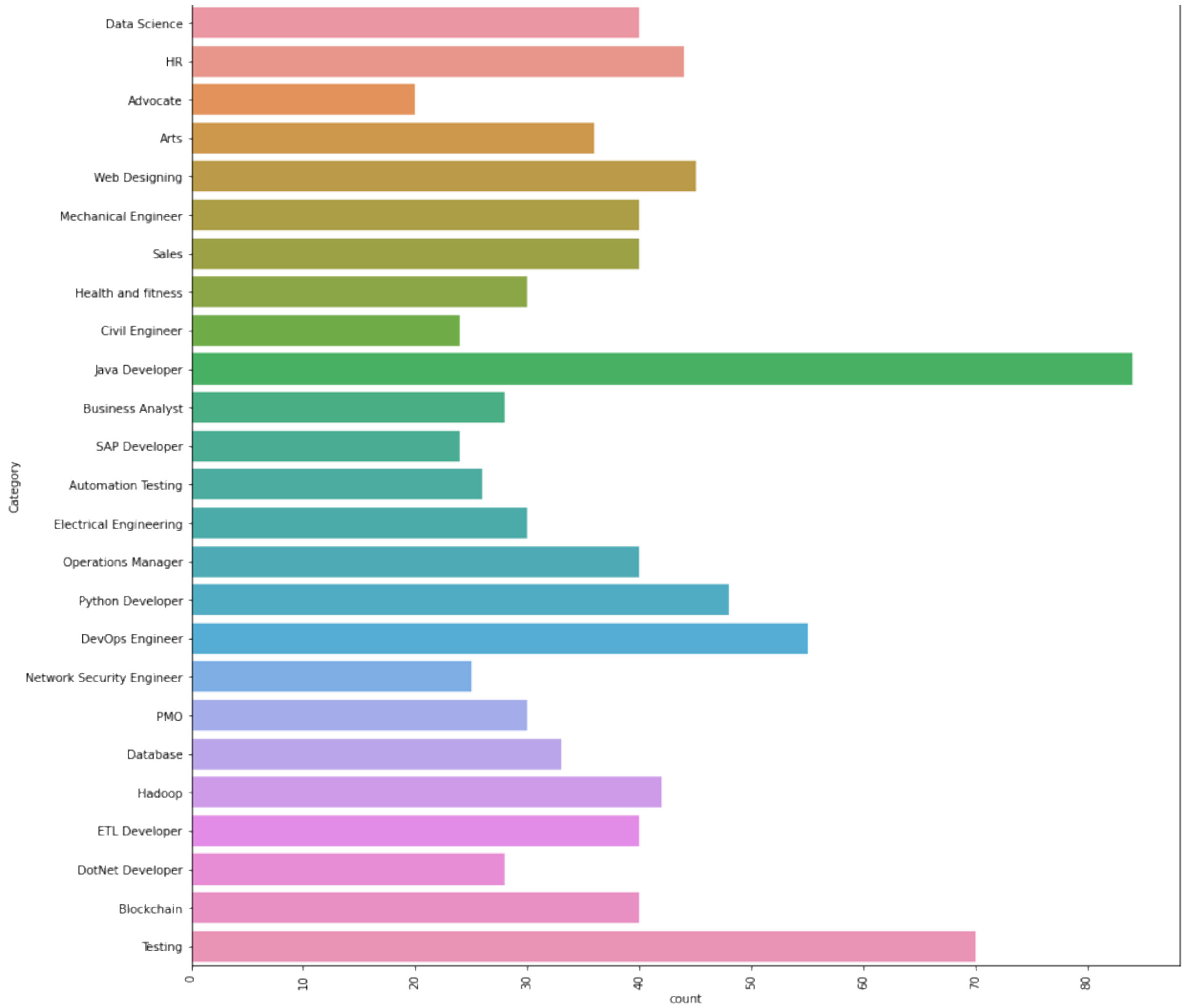


Figure 4.1. Resume Category Distribution

Figure 4.1 showed that resumes were from software industry. We wanted to get sense of most frequent words in dataset. So, we applied data cleaning and stop word removal on resume descriptions. We further created word tokenizer and applied lemmatization and plotted frequency distribution of top 50 words.

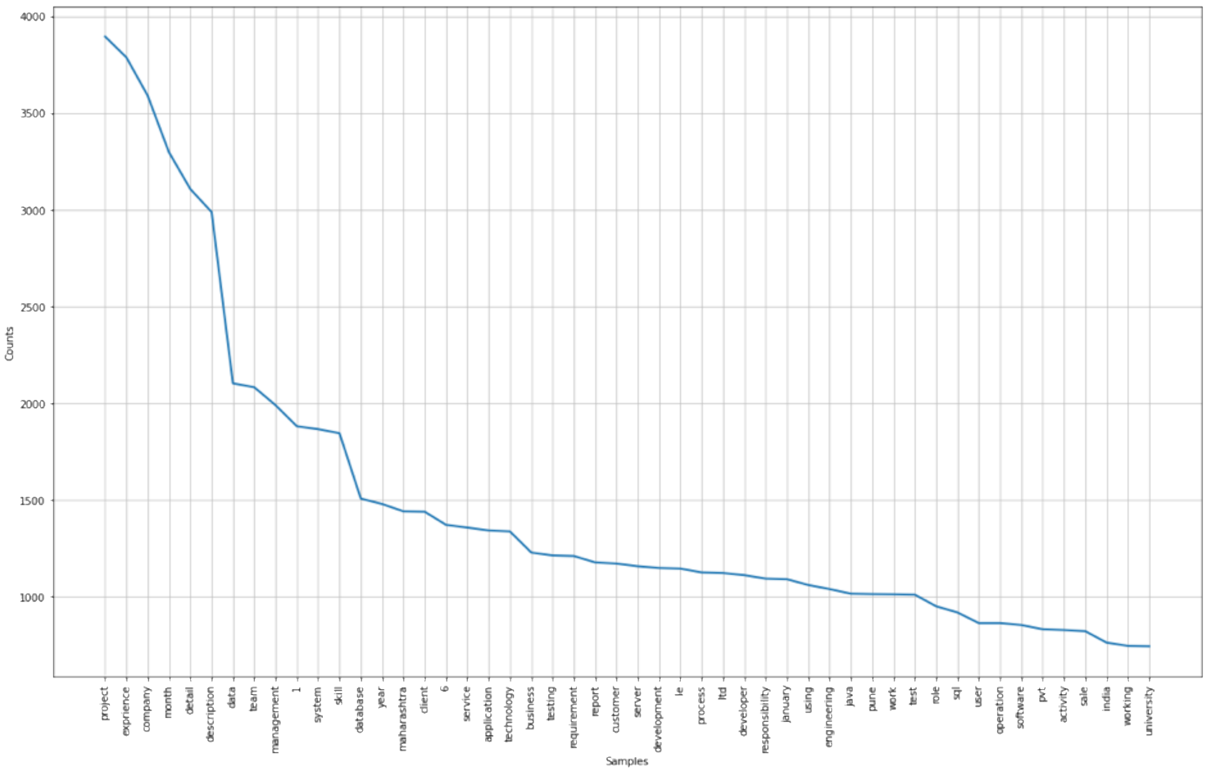


Figure 4.2. Frequency Distribution of Top 50 Words

Figure 4.2 revealed that most candidates were related to Maharashtra, engineering, technology, software industry roles.

Figure 4.3. Word Cloud of Resumes

We further took word cloud of resumes belonging to data science category. Python, deep learning, sentiment analysis were most frequent words as revealed in figure 4.4.

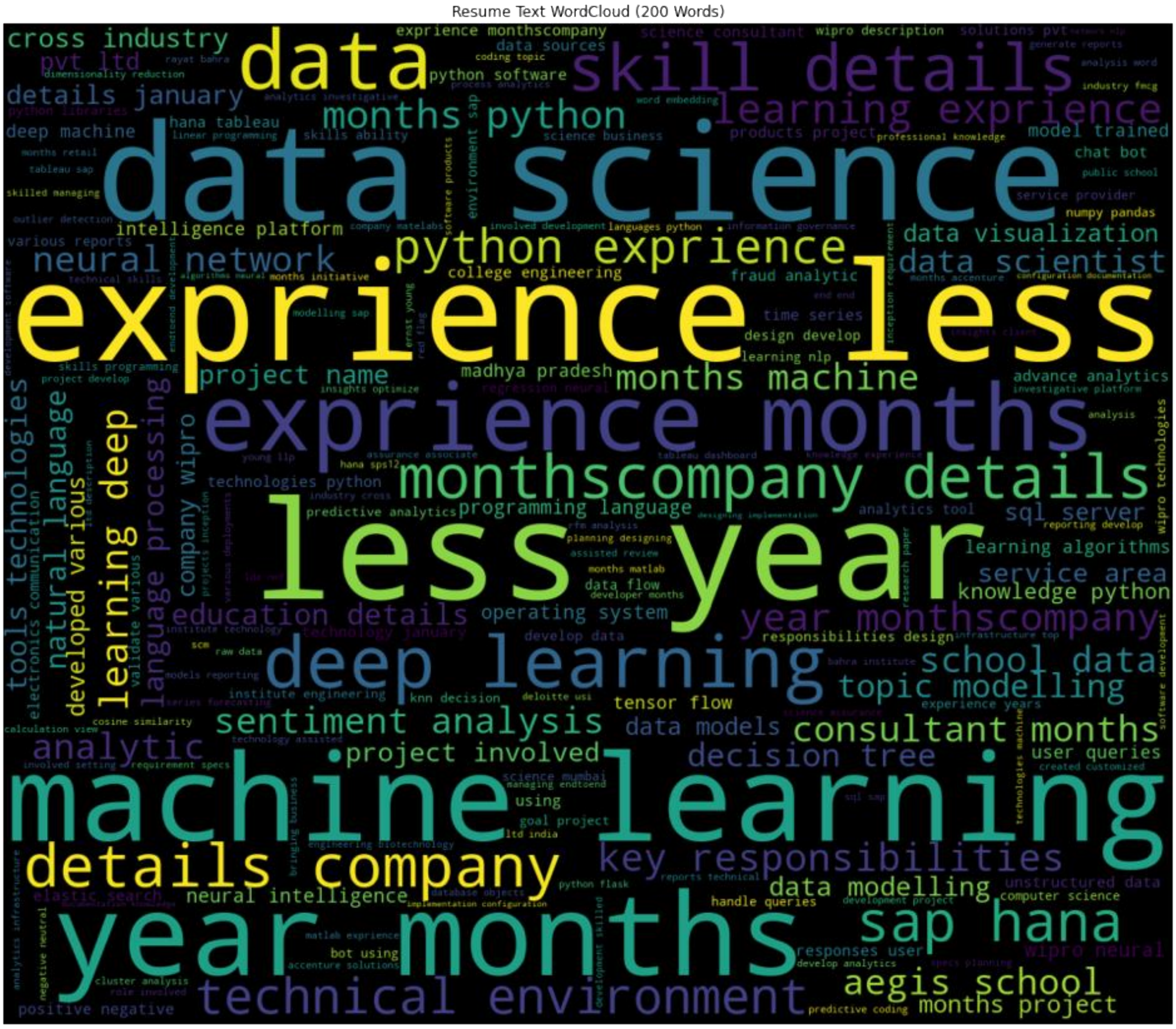


Figure 4.4. Word Cloud of Data Science Resumes

4.2.2 EDA on Job Dataset

Job dataset contained 10K jobs. Each job contained attributes like jobs title, company, city, ratings, summary of job description and date of posting. Maximum word sequence length in any job description was 431. We used job title and summary of job description for analysis.

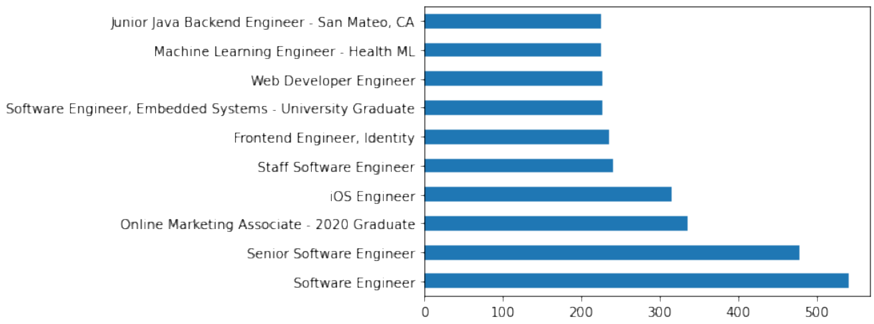


Figure 4.5. Top Ten Job Titles Distribution

Job title distribution in Figure 4.5 showed that jobs roles belong to software industry and same domain as resume dataset. Hence, suitable for resume ranking tasks. We wanted to get sense of most frequent words in dataset. So, we applied data cleaning and stop word removal on job descriptions. We further created word tokenizer and applied lemmatization and plotted word cloud of job descriptions.

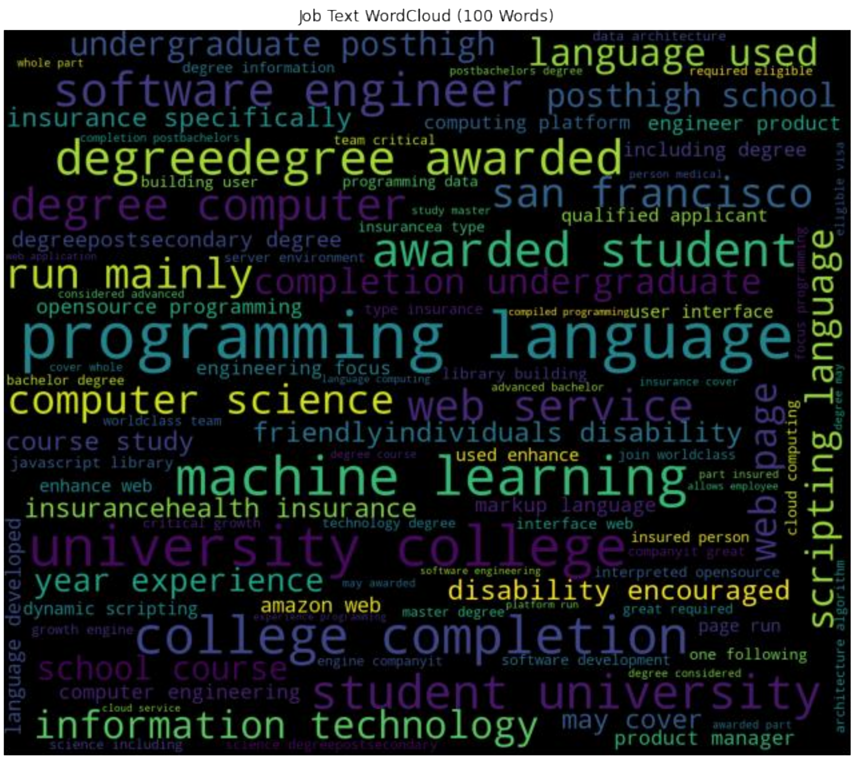


Figure 4.6. Word Cloud of Job descriptions

Figure 4.6 showed most jobs need bachelor’s degree and programming knowledge. We also observed job location belongs to United States.

4.3 Resume Ranking Experiments

We wanted to validate effectiveness of Sentence BERT for resume ranking tasks. For comparison, we experimented various NLP techniques to produce representations of resumes and JDs. We applied cosine similarity on representations and produced ranking for resumes for given JD.

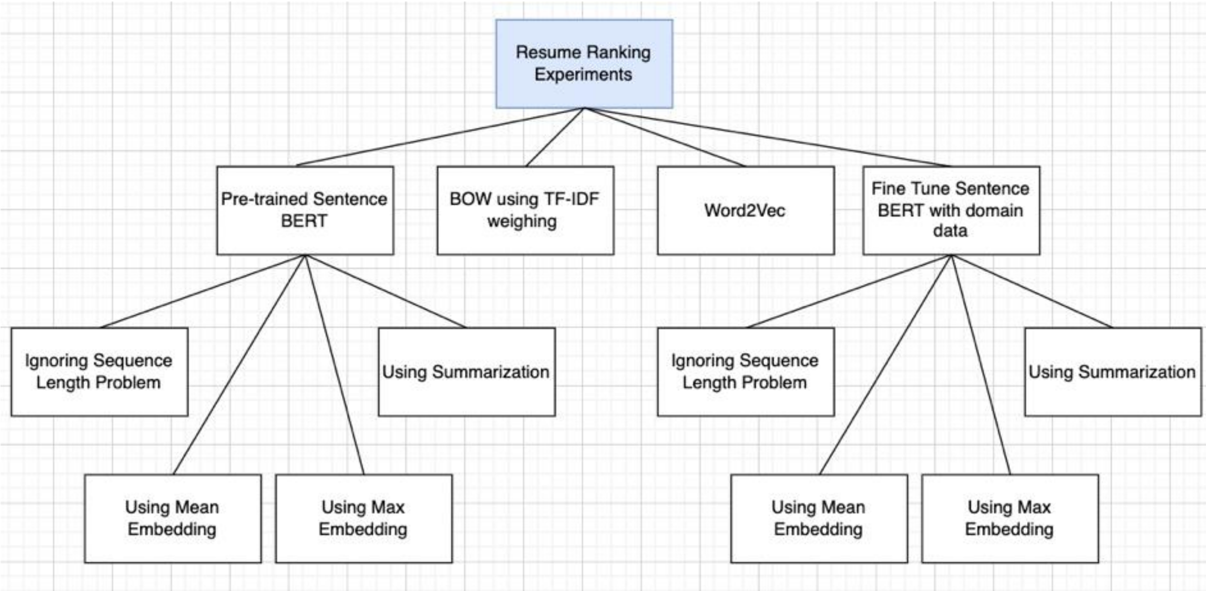


Figure 4.7. Overview Of Experiments

Figure 4.7 gives high level picture of experiments conducted in this research. We have described each experiment in upcoming section in detail and results are discussed in chapter 5.

4.4 Resume Ranking using Sentence BERT

In this experiment, we performed light data cleaning on resumes and JDs. Light data cleaning involved removing Unicode characters, links, removing words with apostrophe, alphanumeric words, removing extra white spaces, removing punctuations. We did not remove stop words because BERT is trained on Wikipedia, news dataset with stop words.

We then passed cleaned data into sentence BERT model. We got embeddings for resumes and JDs as output. Output was stored on disk so it can be directly used during ranking phase. Sentence BERT model took 23 sec on 12 GB GPU to convert 962 resumes with maximum word sequence length of 2069. It took 16 sec to convert 10K JDs with maximum word sequence length of 431.

During resume ranking phase, embedding of input JD was matched with resume embeddings using cosine similarity. Resumes with highest cosine similarity were used to produce output.

4.4.1 Ignoring Sequence Length Problem

Sentence BERT model used in this research can handle maximum sequence length of 256. We experimented by passing resume and JD without any modification to model input and hence ignored sequence length problem. BERT trimmed input sequence to 256 length and created embeddings. We used these embeddings during resume ranking phase with cosine similarity.

4.4.2 Using Mean Embedding

We further experimented by chunking input sequence into multiple chunks of sequence length 256. We then passed each chunk as model input to sentence BERT and took mean of embeddings of all chunks. We repeated this process for each resume and JD to get embeddings of all resumes and JDs in dataset. We used these embeddings during resume ranking phase with cosine similarity.

4.4.3 Using Max Embedding

Like previous section, we also experimented by applying max operation on embeddings of each chunk of input sequence to derive embedding of resume or JD. We used these embeddings during resume ranking phase with cosine similarity.

4.4.4 Using Summarization

Finally, we used BERTSUM summarization library to get summary of resume or JD. We restricted summary to maximum of 250 words for each resume and JD. This pre-processed data was used as input to sentence BERT. Output embedding were used to represent original resume or JD. We eventually applied cosine similarity to get resume ranking for given JD.

4.5 Fine Tuning Model with Domain Data

BERT is pre-trained model. It's trained on large corpus of Wikipedia and news using masked language modelling and next sentence prediction.

We experimented fine tuning of sentence BERT using recruitment dataset. We used resume dataset explained earlier for this purpose. Resume dataset contained resume category column with distribution explained in section 4.2.1.

We performed train/test split of resume data. Train data was further processed to create multiple random pairs of resumes. We created new dataset containing resume one and resume two for each random pair and new derived feature which states whether two resumes belong to same resume category. Total rows in new dataset were roughly 9000.

We used this new derived dataset to fine tune sentence BERT. We passed batch of rows, each containing two resumes as input along with label for match. We trained network over contrastive loss function over 5 epochs. We tried various batch lengths of size 2, 16, 32.

Such experiment expected to train sentence BERT to create embeddings of same resume category closer in vector space. Similarly, embeddings of different categories expected to push away from each other in vector space. Eventually, we used this fine-tuned model for deriving embedding for input resume or JD.

4.5.1 Ignoring Sequence Length Problem

Like section 4.4.1, we ignored input sequence length and passed resume or JD to fine-tuned model. Embedding from model output were used for resume ranking.

4.5.2 Using Mean Embedding

Like section 4.4.2, we passed chunks of resume or JD to fine-tuned model and took mean to derive embedding representation of resume or JD.

4.5.3 Using Max Embedding

Like section 4.4.3, we passed chunks of resume or JD to fine-tuned model and took max to derive embedding representation of resume or JD.

4.5.4 Using Summarization

Like section 4.4.4, we first took summaries of resume or JD restricting maximum sequence length to 250. Summaries are passed to fine-tuned model to get embedding representation of original resume or JD.

4.6 Data Cleaning Experiment with BERT

We experimented heavy and light data cleaning as pre-processing step on input resume or JD before passing sequence to sentence BERT.

Light data cleaning involved removing Unicode characters, links, removing words with apostrophe, alphanumeric words, removing extra white spaces, removing punctuations. Heavy data cleaning involved steps in light data cleaning as well as stop word removal.

4.7 Resume Ranking using TF-IDF Weighting

To compare with baseline model, we experimented bag of words model with TF-IDF weighting on same resume and job dataset for resume ranking task. We removed stop words and created TF-IDF vectorizer. We fitted vectorizer on dataset containing all resumes and job descriptions. Output of vectorizer was TF-IDF matrix. We calculated cosine similarity matrix on TF-IDF matrix. We eventually fetched resumes with maximum cosine score for given JD row in TF-IDF matrix.

4.8 Resume Ranking using Word2Vec Embedding

We also created baseline model for comparison using word2vec (Mikolov et al., 2013). We cleaned data and removed stop words from resumes and JDs. We fetched word2vec embedding for each word in input sequence. We took average of embeddings of all words to derive final embedding of resume or JD. We took cosine similarity to perform resume ranking for given JD.

4.9 Summary

In this chapter, we have presented results from EDA on resume and job datasets. We observed that resume and jobs were belonging to same domain i.e., software industry. Hence suitable for resume ranking experiments. We summarized various resume ranking experiments conducted in this research using BERT and base line TF-IDF or Word2Vec. We experimented impact of data cleaning on BERT performance. We have revealed results of experiments in next chapter.

CHAPTER 5

RESULTS AND DISCUSSIONS

This chapter presents results of various experiments explained in last chapter and discuss interpretation of results.

5.1 Introduction

In this chapter, we will recap usage of ROUGE matrix for current research and present evaluation of resume ranking with sentence BERT in section 5.2. In section 5.3, we will compare ROUGE matrix for various experiments conducted in last chapter. In section 5.4, we will interpret results of experiments with logic behind outcomes. In section 5.5, we will present impact of data cleaning on BERT embedding and downstream task output.

5.2 Evaluation of Resume Ranking with Sentence BERT

We have evaluated resume ranking using ROUGE score between ranked resumes and input JD. ROUGE is a matrix used to evaluate system generated summary against reference summary. So, logically we have used it to compare two documents. We have considered input JD as ground truth and ranked resumes as system generated suggestions. ROUGE provides recall, precision and f score as output and has ROUGE-1 to N versions. We have used ROUGE-1 score with recall which identifies number of terms matched with produced resume ranking against total terms in JD.

We took average of recall values of ROUGE-1 for top ten resumes for random 100 JD. We noted 0.2793 recall score with pre-trained sentence BERT. ROUGE score gave us instrument to compare performance of pre-trained BERT with fine-tuned BERT model and various baseline models experimented in this research for resume ranking task.

We also conducted manual verification of ranked resumes against input JD. We observed that ranked resumes are aligned to JD e.g., backend engineer JD were attracting resumes with Core java and database in descriptions. This proved effectiveness of BERT representational embedding for resume ranking tasks.

5.3 Comparing ROUGE Matrix for Experiments

We have measured ROUGE recall score with various experiments explained in section 4.4, 4.5, 4.7 and 4.8 and below were results. We have applied light data cleaning on input data without stop words removal.

Table 5.1. ROUGE matrix comparison

	ROUGE r score	Common words
Pre-Trained BERT	0.2793	{'agile', 'are', 'experience', 'java', 'team'}
Pre-Trained BERT with Mean embedding	0.3172	{'are', 'build', 'experience', 'java', 'software', 'team'}
Pre-Trained BERT with Max embedding	0.2583	{'java', 'software', 'team'}
Pre-Trained BERT with summarization	0.2690	{'C++,', 'Java', 'a', 'are', 'be', 'of', 'software', 'team', 'to'}
Fine Tuned BERT	0.2137	{'java', 'software'}
Fine Tuned BERT with Mean embedding	0.3079	{'are', 'build', 'java', 'team', 'use', 'using'}
Fine Tuned BERT with Max embedding	0.2574	{'are', 'build', 'experience', 'java', 'software', 'team'}
Fine Tuned BERT with summarization	0.2879	{'Agile', 'a', 'are', 'of', 'part', 'team', 'to', 'using'}
BOW with TF-IDF weighting	0.2983	{'build','developers','experience','java','methodologies','programming','software','team','using'}
Word2Vec	0.3186	{'are', 'build', 'experience', 'java', 'software', 'team'}

Table 5.1 shows ROUGE-1 recall scores and common words between ranked resumes and input JD. ROUGE score gave quantitative measure for comparison. Identifying common words gave insights on effect of various pre-processing and fine-tuning activities on actual outcome.

5.4 Interpreting Results for Experiments

While interpreting various ROUGE-1 r scores in table 5.1, we have observed pre-trained BERT ROUGE-1 recall score (0.2793) was slightly lower than TF-IDF (0.2983) and word2vec (0.3186) models. BERT has trimmed sequence to maximum 256 lengths so did not consider complete resume sentences into consideration. Hence there was information loss.

Pre-trained BERT with mean embedding has significantly increased score to 0.3172. Mean embedding has considered complete sequence into consideration and took mean of chunks so there was no information loss. Results were better than TF-IDF and comparable to word2vec.

Pre-trained BERT with max embedding has scored 0.2574 which was lowest amongst experiments. Max embedding had selected maximum values between chunks and given importance to chunks which were not actual topic of resumes.

Pre-trained BERT applied on BERTSUM summaries (0.2690) has resulted in lower score than base model (0.2983). BERTSUM is trained on news summaries which is different domain than recruitment data of current research. Fine tuning BERTSUM with resumes and their summaries and validating results would be future work for this research. Summaries also incur information loss than original text hence results expected to be lower than methods that considers complete sequence.

Fine tuning sentence BERT with derived dataset as explained in section 4.5, shown rather lower results (0.3079 for fine tuning vs 0.3172 for pre-trained with mean embedding). We used derived data of ~9000 rows from training set. Lowered results indicated need to collect real data of JD and matching resumes of larger sample size and tune sentence BERT with it.

Hence for current dataset, pre-trained BERT with mean embedding has produced best results (0.3172). However, decision is driven by underlying data attributes, distribution. So, all experiments should be conducted if data set or domain changes.

Comparing common words column in table 5.1 shown that BERT based models has produced results which were contextual and more aligned to business expectation. E.g., BERT has ranked resumes higher, containing words like *agile* which was also present in JD along with important words like *java*. When we have inputted JD containing words like *python, ML algorithm* *knowledge* to model, BERT based models has ranked resumes with *python, bagging, boosting* techniques higher. On the other hand, TF-IDF based model has only ranked resumes with *python* keywords. Hence not considered contextuality of *ML algorithm* keyword. Sentence BERT model used in this research is pre-trained on 1B sentence similarity pairs and trained to detect contextuality between sentences which can be seen in experiment results.

BERT models have performed better for JD containing new keywords. We have tested JD with *aerospace* keyword. BERT based model gave results with *mechanical engineer* resumes while TF-IDF results were random.

To check consistency of results, we have tested BERT over test sample size of ~3000 JD and average ROUGE score results were consistent with table 5.1 and stable.

Sentence BERT has processed 962 resumes in 23 seconds and 10000 JDs in 16 seconds. JD were of shorter lengths with max sequence length of 431. Sentence BERT has worked faster with shorter sequence lengths.

TF-IDF or word2vec models were producing common terms like *build, team, task* etc. BERT based model has produced domain specific focussed terms like *agile, java*. Hence reduced significance of frequent words like *build, task* in outcome.

5.5 Evaluating Data Cleaning Experiment on BERT

Table 5.2 shows effect of data cleaning on BERT.

Table 5.2. Data Cleaning Effect on BERT

	ROUGE-1 r score with light pre-process	ROUGE-1 r score without pre-process
Pre-Trained BERT	0.2793	0.2055
Pre-Trained BERT with Mean embedding	0.3172	0.2085
Pre-Trained BERT with Max embedding	0.2583	0.1706
Pre-Trained BERT with summarization	0.2690	0.2061
Fine Tuned BERT	0.2137	0.1729
Fine Tuned BERT with Mean embedding	0.3079	0.2085
Fine Tuned BERT with Max embedding	0.2574	0.2048
Fine Tuned BERT with summarization	0.2879	0.2386

As shown in table 5.2, BERT results were significantly lower without light cleaning. Light cleaning has aligned format of our input data with train data that was used during inception of BERT. Hence results were improved.

Table 5.3 compares light vs heavy data cleaning effect on BERT.

Table 5.3. Light Data Cleaning vs Heavy Data Cleaning

	ROUGE-1 r score with light pre-process	**ROUGE-1 r score with stop word removal**
Pre-Trained BERT	0.2793	0.1491

As shown in table 5.3, BERT performance was declined when stop words removed from input data. BERT pre-training is performed on data with stop words using masked language modelling. Hence stop word removal from input data has resulted in declined performance.

5.6 Summary

As presented in this chapter, pre-trained BERT embedding for resume ranking tasks performed better than BOW model with TF-IDF weighting. Model worked better with new unknown data and produced contextual outcomes aligned to a business goal. Model worked better with light data cleaning without stop word removal. Model worked faster with shorter sequence lengths within range of maximum sequence length of BERT.

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

This chapter provides closing remarks of this study and provides direction for future work.

6.1 Introduction

In this chapter, we will summarize research approach, experiments conducted, results obtained in section 6.2. We will discuss contribution to knowledge in the field of JRS or resume ranking tasks in section 6.3. In section 6.4, we will propose future recommendations based on limitation of this study and literature review performed leading to this study.

6.2 Discussion and Conclusion

In this study, we evaluated effectiveness of sentence BERT to represent sequence of text. We created representational embedding of resumes and JDs using sentence BERT. We further compared embeddings with cosine similarity to derive resume ranking.

We handled limited input sequence length problem of BERT using mean/max chunking, summarization. We have compared ranking results via mean/max chunking, summarization, no pre-process routes. We achieved ROUGE-1 recall score of 0.3172 between input JD and ranked resumes with pre-trained sentence BERT model using mean chunking. Results were better than BOW with TF-IDF weighting (0.2983) and comparable to word2vec (0.3186)

In contrast to interaction based deep learning models for resume ranking, our approach is computationally efficient. Our approach does not need NN pass for each resume and input JD pair during prediction phase to come up with ranking. Compared to other representation-based approaches such as BOW with TF-IDF weighting or word2vec, BERT based model worked better with new unknown data and produced contextual outcomes aligned to a business goal.

We also verified effect of input data cleaning on produced BERT embeddings and hence downstream resume ranking task. BERT based model produced better results with light data cleaning without stop-word removal.

We also performed fine-tuning of sentence BERT with recruitment domain data using contrastive loss function. Fine-tuned model produced embeddings of similar resumes, JDs closer in vector space and pushed away embedding vectors for unmatched resumes, JDs. However, there is scope of improvement in this experiment.

6.3 Contribution to Knowledge

This study has extended existing body of research in JRS by exploring BERT representational embeddings to represent resumes and JDs and compared embeddings to perform JRS or resume ranking. Previous representational embedding approaches for resume ranking has explored custom embeddings with private data and have not evaluated BERT (used as representation).

This study uniquely explored use of ROUGE as evaluation matrix for comparing ranking outcomes when dataset of JD and respective applied resumes is not available. Such evaluation matrix can extend to multiple domains for document ranking settings due to loose coupling.

6.4 Future Recommendations

In future work, we intend to collect labelled JD and matching resumes data and fine tune sentence BERT model with larger sample size. Fine-tuning sentence BERT with domain data expected to produce embeddings of matching JD and resumes closer in vector space. Similarly, better tuned model expected to push unmatched JD, resumes away in vector space. Hence, producing better resume ranking.

We also intend to train BERTSUM summarizer with domain data. It can produce more effective summaries of resumes/JDs representing original document. In effect, it will solve sequence length problem of BERT where we can use summaries instead of original documents for downstream resume ranking task.

One can also explore recent Longformers model for producing embedding representation. Longformers have capacity of processing longer sequence unlike base BERT model. However, Longformers would need to be pre-trained like sentence BERT to make them useful in clustering type of settings.

One can compare resume ranking results of representation-based approach with interaction deep learning approach (explained in section 2.5.5) on same dataset to compare two techniques. End state resume ranking system can be hybrid where representation-based model performing initial screening and interaction-based model producing further domain specific ranking.

REFERENCES

Abel, F., Benczúr, A., Kohlsdorf, D., Larson, M. and Pálovics, R., (2016) RecSys challenge 2016: Job recommendations. In: *RecSys 2016 - Proceedings of the 10th ACM Conference on Recommender Systems*. Association for Computing Machinery, Inc, pp.425–426.

Adomavicius, G. and Tuzhilin, A., (2005) *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*. *IEEE Transactions on Knowledge and Data Engineering*, .

Ahmed, S., Hasan, M., Hoq, M.N. and Adnan, M.A., (2017) User interaction analysis to recommend suitable jobs in career-oriented social networking sites. In: *Proceedings of 2016 International Conference on Data and Software Engineering, ICoDSE 2016*. Institute of Electrical and Electronics Engineers Inc.

Amin, S., Jayakar, N., Sunny, S., Babu, P., Kiruthika, M. and Gurjar, A., (2019) Web Application for Screening Resume. In: *2019 International Conference on Nascent Technologies in Engineering, ICNTE 2019 - Proceedings*. Institute of Electrical and Electronics Engineers Inc.

Anon (2021) *Job Recommendation Challenge | Kaggle*. [online] Available at: https://www.kaggle.com/c/job-recommendation/data [Accessed 31 Dec. 2021].

Anon (2021) *Pretrained Models — Sentence-Transformers documentation*. [online] Available at: https://www.sbert.net/docs/pretrained_models.html [Accessed 3 Nov. 2021].

Anon (2022) *Indeed Software Engineer Job Dataset | Kaggle*. [online] Available at: https://www.kaggle.com/samrat77/indeed-software-engineer-job-dataset [Accessed 2 Jan. 2022].

Anon (2022) *Longformer*. [online] Available at: https://huggingface.co/docs/transformers/model_doc/longformer [Accessed 2 Jan. 2022].

Anon (2022) *NLP-progress/summarization.md at master · sebastianruder/NLP-progress*. [online] Available at: https://github.com/sebastianruder/NLP-progress/blob/master/english/summarization.md [Accessed 3 Jan. 2022].

Anon (2022) *Resume Dataset | Kaggle*. [online] Available at: https://www.kaggle.com/gauravduttakiit/resume-dataset [Accessed 2 Jan. 2022].

Anon (n.d.) Semantic matching between job offers and job search requests.

Armstrong, D.J., ACM Digital Library. and ACM Special Interest Group on Management Information Systems., (2011a) *Proceedings of the 49th SIGMIS annual conference on Computer personnel research.* ACM.

Armstrong, D.J., ACM Digital Library. and ACM Special Interest Group on Management Information Systems., (2011b) *Proceedings of the 49th SIGMIS annual conference on Computer personnel research.* ACM.

Beltagy, I., Peters, M.E. and Cohan, A., (2020) Longformer: The Long-Document Transformer. [online] Available at: http://arxiv.org/abs/2004.05150.

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. [online] Available at: http://arxiv.org/abs/1910.03089.

Bian, S., Chen, X., Zhao, W.X., Zhou, K., Hou, Y., Song, Y., Zhang, T. and Wen, J.R., (2020) Learning to Match Jobs with Resumes from Sparse Interaction Data using Multi-View Co-Teaching Network. In: *International Conference on Information and Knowledge Management, Proceedings*. Association for Computing Machinery, pp.65–74.

Bian, S., Zhao, W.X., Song, Y., Zhang, T. and Wen, J.-R., (n.d.) *Domain Adaptation for Person-Job Fit with Transferable Deep Global Match Network*. [online] Available at: https://www.linkedin.com.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q. v. and Salakhutdinov, R., (2019) Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. [online] Available at: http://arxiv.org/abs/1901.02860.

Dave, V.S., al Hasan, M., Zhang, B., AlJadda, K. and Korayem, M., (2018) A combined representation learning approach for better job and skill recommendation. In: *International Conference on Information and Knowledge Management, Proceedings*. Association for Computing Machinery, pp.1997–2006.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K., (2018) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. [online] Available at: http://arxiv.org/abs/1810.04805.

Ekwoaba, J.O., Ikeije, U.U. and Ufoma, N., (2015) *THE IMPACT OF RECRUITMENT AND SELECTION CRITERIA ON ORGANIZATIONAL PERFORMANCE*. [online] *Global Journal of Human Resource Management*, Available at: www.eajournals.org.

Färber, F., Weitzel, T. and Keim, T., (n.d.) *Färber, Keim, Weitzel: Automated Recommendation in Personnel Recruitment AN AUTOMATED RECOMMENDATION APPROACH TO SELECTION IN PERSONNEL RECRUITMENT*.

Freire, M.N. and de Castro, L.N., (2021) e-Recruitment recommender systems: a systematic review. *Knowledge and Information Systems*, 631, pp.1–20.

Gugnani, A. and Misra, H., (n.d.) *Implicit Skills Extraction Using Document Embedding and Its Use in Job Recommendation*. [online] Available at: https://www.computerhope.com/.

Hanani, U., Shapira, B. and Shoval, P., (2001) Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 113, pp.203–259.

Le, R., Zhang, T., Hu, W., Zhao, D., Song, Y. and Yan, R., (2019) Towards effective and interpretable person-job fitting. In: *International Conference on Information and Knowledge Management, Proceedings*. Association for Computing Machinery, pp.1883–1892.

Lee, D.H. and Brusilovsky, P., (2007) Fighting information overflow with personalized comprehensive information access: A proactive job recommender. In: *3rd International Conference on Autonomic and Autonomous Systems, ICAS'07*.

Li, J., Arya, D., Ha-Thuc, V. and Sinha, S., (2016) How to get them a dream job?: Entity-aware features for personalized job search ranking. In: *Proceedings of the ACM SIGKDD International*

Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, pp.501–510.

Lin, C.-Y., (n.d.) *ROUGE: A Package for Automatic Evaluation of Summaries*.

Lin, J., Nogueira, R. and Yates, A., (2020) Pretrained Transformers for Text Ranking: BERT and Beyond. [online] Available at: http://arxiv.org/abs/2010.06467.

Lin, Y., Lei, H., Addo, P.C. and Li, X., (2016) Machine Learned Resume-Job Matching Solution. [online] Available at: http://arxiv.org/abs/1607.07657.

Liu, K., Shi, X., Kumar, A., Zhu, L. and Natarajan, P., (2016) Temporal Learning and Sequence Modeling for a Job Recommender System. [online] Available at: http://arxiv.org/abs/1608.03333.

Liu, Y. and Lapata, M., (2019) Text Summarization with Pretrained Encoders. [online] Available at: http://arxiv.org/abs/1908.08345.

Luo, Y., Wen, Y., Zhang, H. and Zhang, X., (2019) Resumegan: An optimized deep representation learning framework for talent-job fit via adversarial learning. In: *International Conference on Information and Knowledge Management, Proceedings*. Association for Computing Machinery, pp.1101–1110.

Mahdi, H.F., Dagli, R., Mustufa, A. and Nanivadekar, S., (2021) Job Descriptions Keyword Extraction using Attention based Deep Learning Models with BERT. In: *HORA 2021 - 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Proceedings*. Institute of Electrical and Electronics Engineers Inc.

Maheshwary, S. and Misra, H., (2018) Matching Resumes to Jobs via Deep Siamese Network. In: *The Web Conference 2018 - Companion of the World Wide Web Conference, WWW 2018*. Association for Computing Machinery, Inc, pp.87–88.

Malinowski, J., Weitzel, T. and Keim, T., (2005) *Analyzing the Impact of IS Support on Recruitment Processes: An E-Recruitment Phase Model. Operational Business/IT Alignment View project SENECA-The Role of External Social Networks for the Innovativeness of the Firm View project Analyzing the Impact of IS Support on Recruitment Processes: An E-Recruitment Phase Model*. [online] Available at: https://www.researchgate.net/publication/221229287.

Mikolov, T., Chen, K., Corrado, G. and Dean, J., (2013) Efficient Estimation of Word Representations in Vector Space. [online] Available at: http://arxiv.org/abs/1301.3781.

Mostafa, L. and Beshir, S., (2021) Job Candidate Rank Approach Using Machine Learning Techniques. In: *Advances in Intelligent Systems and Computing*. Springer Science and Business Media Deutschland GmbH, pp.225–233.

Mulyar, A., Schumacher, E., Rouhizadeh, M. and Dredze, M., (2019) Phenotyping of Clinical Notes with Improved Document Classification Models Using Contextualized Neural Language Models. [online] Available at: http://arxiv.org/abs/1910.13664.

Nigam, A., Peoplestrong, A.R., Saxena, A. and Singh Peoplestrong, H., (n.d.) *Job Recommendation: Leveraging Progression of Job Applications*.

Opatha, H., (2009) *Human Resource Management: Personnel*.

Qin, C., Zhu, H., Xu, T., Zhu, C., Jiang, L., Chen, E. and Xiong, H., (2018) Enhancing person-job fit for talent recruitment: An ability-aware neural network approach. In: *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*. Association for Computing Machinery, Inc, pp.25–34.

Ramanath, R., Inan, H., Polatkan, G., Hu, B., Guo, Q., Ozcaglar, C., Wu, X., Kenthapadi, K. and Geyik, S.C., (2018) Towards Deep and Representation Learning for Talent Search at LinkedIn. [online] Available at: http://arxiv.org/abs/1809.06473.

Reimers, N. and Gurevych, I., (2019) Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. [online] Available at: http://arxiv.org/abs/1908.10084.

Roy, P.K., Chowdhary, S.S. and Bhatia, R., (2020) A Machine Learning approach for automation of Resume Recommendation system. In: *Procedia Computer Science*. Elsevier B.V., pp.2318–2327.

de Ruijt, C. and Bhulai, S., (2021) Job Recommender Systems: A Review. [online] Available at: http://arxiv.org/abs/2111.13576.

Singh, A., Catherine, R., Visweswariah, K., Chenthamarakshan, V. and Kambhatla, N., (2010) PROSPECT: A system for screening candidates for recruitment. In: *International Conference on Information and Knowledge Management, Proceedings*. pp.659–668.

Syed, A.A., Gaol, F.L. and Matsuo, T., (2021) *A survey of the state-of-the-art models in neural abstractive text summarization*. *IEEE Access*, .

T. Al-Otaibi, S., (2012) A survey of job recommender systems. *International Journal of the Physical Sciences*, 729.

Van-Duyet, L., Quan, V.M. and An, D.Q., (2017) Skill2vec: Machine Learning Approach for Determining the Relevant Skills from Job Description. [online] Available at: http://arxiv.org/abs/1707.09751.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I., (2017) Attention Is All You Need. [online] Available at: http://arxiv.org/abs/1706.03762.

Vishruth, R.G., Sunitha, R., Varuna, K.S., Varshini, N. and Honnavalli, P.B., (2020) Resume Scanning and Emotion Recognition System based on Machine Learning Algorithms. In: *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. pp.1127–1132.

Wei, K., Huang, J. and Fu, S., (2007) *A Survey of E-Commerce Recommender Systems*. *Proceedings of the International Conference on Service Systems and Service Management*.

Yan, R., Zhang, T., Le, R., Zhang, X., Song, Y. and Zhao, D., (2019) Interview choice reveals your preference on the market: To improve job-resume matching through profiling memories. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, pp.914–922.

Zhang, B. and al Hasan, M., (2017) Name disambiguation in anonymized graphs using network embedding. In: *International Conference on Information and Knowledge Management, Proceedings*. Association for Computing Machinery, pp.1239–1248.

Zhao, J., Wang, J., Sigdel, M., Zhang, B., Hoang, P., Liu, M. and Korayem, M., (2021) Embedding-based Recommender System for Job to Candidate Matching on Scale. [online] Available at: http://arxiv.org/abs/2107.00221.

Zheng, Y., Pan, W., Sahebi, S. and Fernández, I., (n.d.) The 1st Workshop on Intelligent Recommender Systems by Knowledge Transfer & Learning (RecSysKTL). [online] 17. Available at: http://dx.doi.org/10.1145/3109859.3109951 [Accessed 31 Dec. 2021].

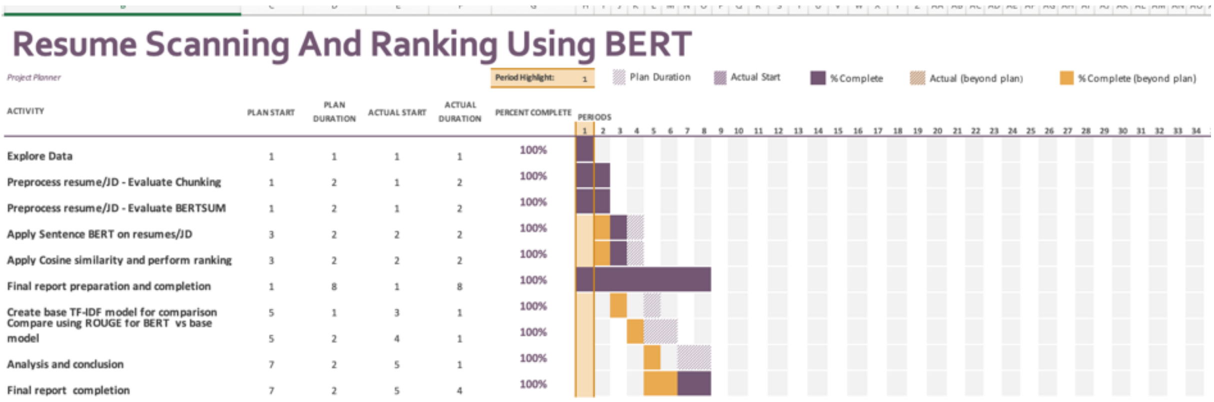
APPENDIX A: RESEARCH PROPOSAL

Attaching original research proposal submitted earlier.

https://drive.google.com/file/d/1B04aSTwdOwKIUdjkJ8Frxd0-B7mNnht6/view?usp=sharing

APPENDIX B: RESEARCH PLAN

Below plan is from mid thesis report submission to thesis completion (in weeks)



ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE
Explore Data	1	1	1	1	100%
Preprocess resume/JD - Evaluate Chunking	1	2	1	2	100%
Preprocess resume/JD - Evaluate BERTSUM	1	2	1	2	100%
Apply Sentence BERT on resumes/JD	3	2	2	2	100%
Apply Cosine similarity and perform ranking	3	2	2	2	100%
Final report preparation and completion	1	8	1	8	100%
Create base TF-IDF model for comparison	5	1	3	1	100%
Compare using ROUGE for BERT vs base model	5	2	4	1	100%
Analysis and conclusion	7	2	5	1	100%
Final report completion	7	2	5	4	100%

Figure 7.1. Research Plan