T.C.

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF GRADUATE STUDIES

RESUME RECOMMENDATION USING RNN CLASSIFICATION AND COSINE SIMILARITY

MASTER'S THESIS

Issa DİALLO

Department of Software Engineering

Artificial Intelligence and Data Science

SEPTEMBER 2023

T.C.
ISTANBUL AYDIN UNIVERSITY
INSTITUTE OF GRADUATE STUDIES

STRATEGIC PLAN TO IMPROVE MARKETING MANAGEMENT OF PRODUCTS

MASTER'S THESIS

Issa DİALLO
(Y2013.140012)

Department of Software Engineering
Artificial Intelligence and Data Science

Thesis Advisor: Assoc.Prof. (Ph.D.) ILHAM HUSEYINOV

SEPTEMBER, 2023

APPROVAL PAGE

DECLARATION

I hereby declare with respect that the study “Resume Recommendation using RNN Classification and Cosine Similarity**”,** which I submitted as a Master thesis, is written without any assistance in violation of scientific ethics and traditions in all the processes from the Project phase to the conclusion of the thesis and that the works I have benefited are from those shown in the References. (19/08/2023)

Issa DIALLO

FOREWORD

Welcome to my thesis, "Resume Recommendation using RNN Classification and Cosine Similarity." In this work, I delve into the exciting world of machine learning and recommendation systems, aiming to enhance the accuracy of resume recommendations by employing cutting-edge techniques.

I express my heartfelt gratitude to my esteemed advisors, Assoc. Prof. (Ph.D.) ILHAM HUSEYINOV, for his continuous guidance and insightful feedback that have been the cornerstone of this endeavor.

I would also like to extend my appreciation to the members of the jury, Dr. Öğr. Üyesi Rıfat BENVENİSTE, Dr. Öğr. Üyesi Osman SELVİ, Öğr. Gör. Dr. MHD WASIM RAED and Dr. Öğr. Üyesi ALPARSLAN HORASAN, for their valuable time, expertise, and constructive insights during the evaluation of this work.

I am grateful to my friends and family for their unwavering support throughout this academic journey, and to my colleagues for their engaging discussions and encouragement.

This thesis represents a step towards innovation in the recruitment landscape, showcasing the potential of Recurrent Neural Networks and Cosine Similarity in revolutionizing how resumes are recommended and matched. Join me as we navigate through the intricacies of this exciting field.

Şeptember 2021 Issa DIALLO

RESUME RECOMMENDATION USING RNN CLASSIFICATION AND COSINE SIMILARITY

ABSTRACT

One way to save time and resources in the human recruitment and hiring process is to post open job positions on the Internet. This allows for a wider reach and attracts a larger pool of potential candidates. However, the sheer volume of applications received often creates challenges for hiring managers and companies to identify the most suitable candidate efficiently.

To address this issue, intelligent tools can be employed, such as deep learning algorithms and recommender systems. These advanced technologies can expedite the hiring process and aid in the identification of the right candidate for a particular job. In this paper, we propose a two-fold algorithmic approach to tackle this problem.

Firstly, we suggest building a Recurrent Neural Network (RNN) classifier for resume classification. RNNs are a type of neural network specifically designed for sequence data, making them well-suited for analyzing and extracting relevant information from resumes. This classifier will help automate the initial screening process by categorizing resumes based on their suitability for a given job.

Secondly, we propose using cosine similarity to create a resume recommendation system. By calculating the similarity between the requirements of a job and the content of a candidate's resume, we can identify the best fit candidates more efficiently. This recommendation system assists hiring managers in shortlisting candidates who closely match the job requirements.

To evaluate the performance of our proposed RNN classifier, we assess several criteria, including accuracy, precision, recall, F-score, and the confusion matrix. Our experiments demonstrate that the RNN classifier outperforms other classifiers, such as Gaussian Naive Bayes (GNB), Linear Support Vector Machines (SVM), and Random Forest (RF), when tested on the same dataset. The same RNN

that we have in our research produced the same performance with BERT on the same dataset.

Keywords: Resume classification, recommender systems, RNN, NLP, cosine similarity, TF-IDF, Transformer

RNN SINIFLANDIRMASI VE KOSINÜS BENZERLIĞI KULLANARAK ÖNERIYE DEVAM ETME

ÖZET

İnsan işe alma ve işe alma sürecinde zamandan ve kaynaklardan tasarruf etmenin bir yolu, açık iş pozisyonlarını İnternet'te yayınlamaktır. Bu, daha geniş bir erişim sağlar ve daha büyük bir potansiyel aday havuzunu çeker. Bununla birlikte, alınan başvuruların hacmi, işe alım yöneticileri ve şirketler için en uygun adayı verimli bir şekilde belirleme konusunda genellikle zorluklar yaratır.

Bu sorunu çözmek için derin öğrenme algoritmaları ve tavsiye sistemleri gibi akıllı araçlar kullanılabilir. Bu ileri teknolojiler, işe alım sürecini hızlandırabilir ve belirli bir iş için doğru adayın belirlenmesine yardımcı olabilir. Bu yazıda, bu sorunu çözmek için iki katlı bir algoritmik yaklaşım öneriyoruz.

İlk olarak, özgeçmiş sınıflandırması için bir Tekrarlayan Sinir Ağı (RNN) sınıflandırıcısı oluşturmanızı öneririz. RNN'ler, dizi verileri için özel olarak tasarlanmış bir tür sinir ağıdır ve bu, onları özgeçmişlerden ilgili bilgileri analiz etmek ve çıkarmak için çok uygun hale getirir. Bu sınıflandırıcı, özgeçmişleri belirli bir işe uygunluklarına göre kategorize ederek ilk tarama sürecini otomatikleştirmeye yardımcı olacaktır.

İkinci olarak, özgeçmiş öneri sistemi oluşturmak için kosinüs benzerliğini kullanmayı öneriyoruz. Bir işin gereklilikleri ile bir adayın özgeçmişinin içeriği arasındaki benzerliği hesaplayarak, en uygun adayları daha verimli bir şekilde belirleyebiliriz. Bu öneri sistemi, işe alma yöneticilerine iş gereklilikleriyle yakından eşleşen adayları kısa listeye alma konusunda yardımcı olur.

Önerilen RNN sınıflandırıcımızın performansını değerlendirmek için doğruluk, kesinlik, hatırlama, F-skoru ve karışıklık matrisi dahil olmak üzere çeşitli kriterleri değerlendiriyoruz. Deneylerimiz, RNN sınıflandırıcısının aynı veri kümesi üzerinde test edildiğinde Gaussian Naive Bayes (GNB), Lineer Destek Vektör

Makineleri (SVM) ve Random Forest (RF) gibi diğer sınıflandırıcılardan daha iyi performans gösterdiğini göstermektedir.

Araştırmamızda sahip olduğumuz aynı RNN, aynı veri kümesinde BERT ile aynı performansı üretti.

Anahtar Kelimeler: Özgeçmiş sınıflandırması, tavsiye sistemleri, RNN, NLP, kosinsüs benzerliği,TF-IDF

TABLE OF CONTENTS

DECLARATION ..... **i**
FOREWORD ..... **ii**
ABSTRACT ..... **iii**
ÖZET ..... **v**
TABLE OF CONTENTS ..... **vii**
LIST OF TABLES ..... **x**
LIST OF FIGURES ..... **xi**
I. INTRODUCTION **1**
A. Purpose of the Study 1
B. Background and Statistic 1
C. Problem Definition 2
D. Research Question and Objective 2
E. Research Objective 3
F. Document Outline 3
II. LITERATURE REVIEW **5**
A. Resume Classifier Related Works 5
B. Resume Recommender Related Works 6
III. METHODOLOGY **8**
A. Workflow 8
B. Data Collection 9
1. Fields Filling 9
2. Files Scraping 10
C. Data Preparation 10
1. Text cleaning 10
2. Feature Extraction 10
3. Tokenization 11
4. Stop Words Removal 11
5. Lemmatization or Stemming 11

6. Handling Case Sensitivity 12
7. Feature Vectorization 12
8. Data Splitting 12
a. Training Set 13
b. Validation Set 13
c. Test Set 13
D. Algorithms 13
1. Recurrent Neural Network 13
2. Linear Support Vector Machine 14
3. Gaussian N Bayes 15
4. Random Forest 15
5. Cosine Similarity 16
6. Bert Model 17
E. Evaluation 17
1. Accuracy 17
2. Confusion Matrix 17
3. Recall 18
4. Precision 18
5. F1-score 18
IV. IMPLEMENTATION 19
A. Data Understanding 19
B. B. Feature Extraction 20
1. Data Preprocessing 21
C. Feature Vectorization and Label Encoding 22
D. Classification 23
1. RNN Parameters 24
2. Functions 24
3. Training 24
4. Tools 24
E. Similarity 25
V. EXPERIMET RESULTS & EVALUATION 26
A. Evaluation Metrics 26
B. Confusion Matrix 27
VI. DISCUSSIONS & CONCLUSION 28

VII. REFERENCES 29
RESUME 32

LIST OF TABLES

Table 1 attributes of first row of the dataset 19
Table 2 Dataset before and after cleaning..... 22
Table 3 Similarity score example..... 25
Table 4 Comparison of Evaluation metrics..... 26

LIST OF FIGURES

Figure 1 Resume classification and recommendation workflow 8
Figure 2 Data distribution over the different job classes 20
Figure 3 Feature extraction code 21
Figure 4 Data cleaning code 22
Figure 5 Label encoder code 23
Figure 6 Many-To-One RNN Architecture 23
Figure 7 Confusion Matrix of RNN Model 27

I. INTRODUCTION

A. Purpose of the Study

The purpose of this study is to propose and evaluate a Resume Recommendation system using RNN Classification (building a recurrent neural network (RNN) classifier for resume classification) and Cosine Similarity (using cosine similarity for resume recommendation to find a candidate that fits job requirements best). The study aims to explore the effectiveness of these techniques in matching job seekers with relevant job opportunities based on their resumes. By developing and implementing this system, the study seeks to improve the efficiency and accuracy of the job search process for both job seekers and employers.

B. Background and Statistic

The evolution of technology has led to the widespread use of big data analytics and artificial intelligence in various business domains. However, the successful application of such methods and techniques to effective Human Resource (HR) management, particularly in areas like recruiting and corporate population management within large organizations, remains relatively limited. The radical growth of the job market has proven that traditional methods of recruitment are becoming less useful and inefficient. In today's competitive job market, it has become increasingly challenging for job seekers to find suitable employment opportunities. Similarly, employers often face difficulties in identifying the most qualified candidates from a large pool of applicants. The use of technology, such as machine learning algorithms, has shown promising results in addressing these challenges. However, traditional methods based on keyword matching often overlook relevant skills and experiences, leading to mismatches between job seekers and job openings.

The proposed approach in this study combines two techniques: RNN (Recurrent Neural Network) Classification and Cosine Similarity. RNN

Classification is a deep learning algorithm capable of capturing sequential information in textual data, making it suitable for analyzing resume content. Cosine Similarity, on the other hand, is a metric that measures the similarity between two vectors, in this case, the similarity between a job seeker's resume and a job description.

To support the study, relevant statistics regarding the job market and the difficulties faced by both job seekers and employers can be presented. This may include data on the number of job applications per vacancy, the average time taken to fill a job opening, and the rate of job mismatches.

C. Problem Definition

The problem addressed in this study is the inefficiency and inaccuracy of the job search process for job seekers and employers. Traditional methods of matching job seekers with job openings often rely on keyword matching, which overlooks the nuanced skills and experiences mentioned in resumes. This leads to mismatches, where qualified candidates are overlooked, or irrelevant applicants are considered. Our research endeavors to address this challenge by introducing a novel Resume Recommendation system that harnesses the power of RNN Classification and Cosine Similarity. The proposed system aims to deliver highly precise and pertinent job recommendations, enhancing the overall job-seeking experience.

D. Research Question and Objective

The primary research question of this study is: How effective is the proposed Resume Recommendation system using RNN Classification and Cosine Similarity in matching job seekers with relevant job opportunities based on their resumes?

The specific objectives of the study are:

- To develop a Resume Recommendation system using RNN Classification and Cosine Similarity.
- To assess the efficacy of the proposed system in delivering precise and pertinent job recommendations.
- To conduct a comparative analysis, assessing the effectiveness of the proposed

system against existing methods that utilize keyword-based matching.

- To assess the user satisfaction and usability of the proposed system.

E. Research Objective

The overall research objective is to contribute to the improvement of the job search process by developing and evaluating a Resume Recommendation system using advanced techniques such as RNN Classification and Cosine Similarity. The study aims to provide insights into the effectiveness of these techniques in addressing the challenges faced by job seekers and employers in finding suitable matches. By achieving this objective, the study seeks to enhance the efficiency and accuracy of the job market, benefiting both job seekers and employers.

F. Document Outline

This section of the thesis document provides an overview of the entire report, which begins by clearly defining and explaining the research problem and highlighting its significance. The report will also delve into the specifics of the problem, outlining its purpose and identifying relevant research questions that will guide the research process.

Chapter 2 (Literature Review)

This chapter provides an in-depth analysis of past research conducted in this field, with a focus on comparing various supervised machine learning techniques or deep learning models that have been used to predict the class of the resume and the review of recommenders.

Chapter 3 (Design and Methodologies)

provides a detailed description of the design and methodology that were employed to address the research problem. This chapter offers an in-depth

explanation of the specific steps that were taken to conduct the study, and it delves into the details of each step.

Chapter 4 (Implementation and Results)

showcases the implementation details and results obtained from the study. It

provides a detailed explanation of the specific model that were selected and used for the research. Furthermore, this chapter provides a comprehensive justification for the selection of the model and evaluates it performance. The hypothesis of the research is also considered and evaluated by comparing the results obtained from the model to other machine learning model.

Chapter 5 (Conclusion)

provides an overall discussion of the research problem, the obtained results, and their evaluation. This chapter offers a comprehensive summary of the research and emphasizes its contribution to addressing the research question. Furthermore, it outlines potential avenues for future research in the related field.

II. LITERATURE REVIEW

In this Chapter as our system is a two folded system then we first review research papers related to Resume classifiers, then papers related to Resume recommenders.

A. Resume Classifier Related Works

Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, Rocky Bhatia proposed two-phase process of resume classification and content-based ranking using cosine similarity. The approach seems to be like the approach presented in our paper. But we proposed a RNN classifier, compared it with the classifiers given by them. The performance of the RNN classifier overperformed the performance of classifiers presented by them.

Q. Xu, J. Zhang, Y. Zhu, B. Li, D. Guan, and X. Wang proposed a block-level bidirectional recurrent neural network (BBRNN) model for resume block classification, which considers the contextual order of different blocks within a resume. The performance of the proposed model is 6% to 9% higher than existing methods. The paper also discusses related work on resume information extraction and block classification methods. However, the paper could benefit from further discussion of limitations and potential future research directions. We chose simple RNN because of the efficient computationally, simple architecture, easy to train.

Pal Riya; Shaikh Shahrukh; Satpute Swaraj; Bhagwat Sumedha presented the use of machine learning algorithms such as Naive Bayes, Random Forest, and SVM for resume classification and skill extraction. Also, an NLP method TF-IDF is used for vectorization.

Ali, I., Mughal, N., Khand Z. H., Ahmed J; Mujtaba, G., develop a resume classification system which is developed by integrating natural language processing and machine learning techniques. SW. Kim, JM. Gil are devoted to the classification problem of research papers. A comprehensive survey on applications of deep

learning techniques using NLP is given by P. Lavanya and E. Sasikala. K. Yu, G. Guan, and M. Zhou used the resume information extraction with cascaded hybrid model is presented.

Vedant Bhatia, Prateek Rawat, Ajit Kumar, Rajiv Ratn Shah proposed two stages strategies which is first extracting all the information from the candidates resumes and then using the BERT sentence pair classification which allows the ranking based on the job description. The proposed model used BERT which is a transformer model has Showed good results in the world of NLP. Bidirectional Encoder Representations (BERT) from Transformers, better known as BERT, is a revolutionary paper by Google that increased the state-of-the-art performance for various NLP tasks and was the steppingstone for many other revolutionary architectures .In their paper we have similar method but the only difference is the quantity of data used in the training because in model such as BERT we need a lot of data in order for the data to generalize , while for our model we don't have a lot of parameters therefore no need for a lot of data in order for the model to generalize.

B. Resume Recommender Related Works

CHEN ZHU, HENGSHU ZHU, HUI XIONG , CHAO MA , FANG XIE , PENGLIANG DING , PAN LI proposed a joint representation learning framework for predicting person-job fit. The framework combines information from both the job description and the candidate's resume to create a joint representation that captures the match between the candidate's skills and the job requirements. The proposed approach is based on a deep neural network architecture that uses a combination of convolutional and recurrent layers to process both textual and structured data. The authors conducted experiments on a publicly available dataset of job postings and resumes and showed that their approach outperformed several baseline methods in predicting person-job fit. Overall, the paper presents a promising approach to improving the recruitment process by automatically identifying candidates who are a good match for a particular job based on their resumes and job descriptions.

Sanjjushri Varshini, Ponshriharini V, Santhosh Kannan, SnekhaSuresh, Harshavardhan Ramesh, Rohith Mahadevan, Raja CSP Raman aim to identify the most suitable candidate for a job position based on their productivity and working pattern similarity. To achieve this, they collected data from workers in IT companies,

focusing on their activities. The similarity test is conducted based on the new candidate's resume and the position they are applying for. The authors used various scores, such as Turtle-Score, GitHub Turtle Score, Learning Analytics Turtle Score, and Kaggle Score, as data to determine similarity. They employed several similarity algorithms, including Chebyshev, Canberra, Euclidean, Manhattan, Minkowski, Bray-Curtis, and Cosine.

III. METHODOLOGY

A. Workflow

The workflow diagram, as depicted in Fig.1, provides a comprehensive overview of the step-by-step process involved in constructing of our Recurrent Neural Network (RNN) classifier and implementing a robust resume recommender system with the cosine similarity. This diagram serves as a visual representation of the sequential stages and key components necessary to accomplish these tasks effectively.

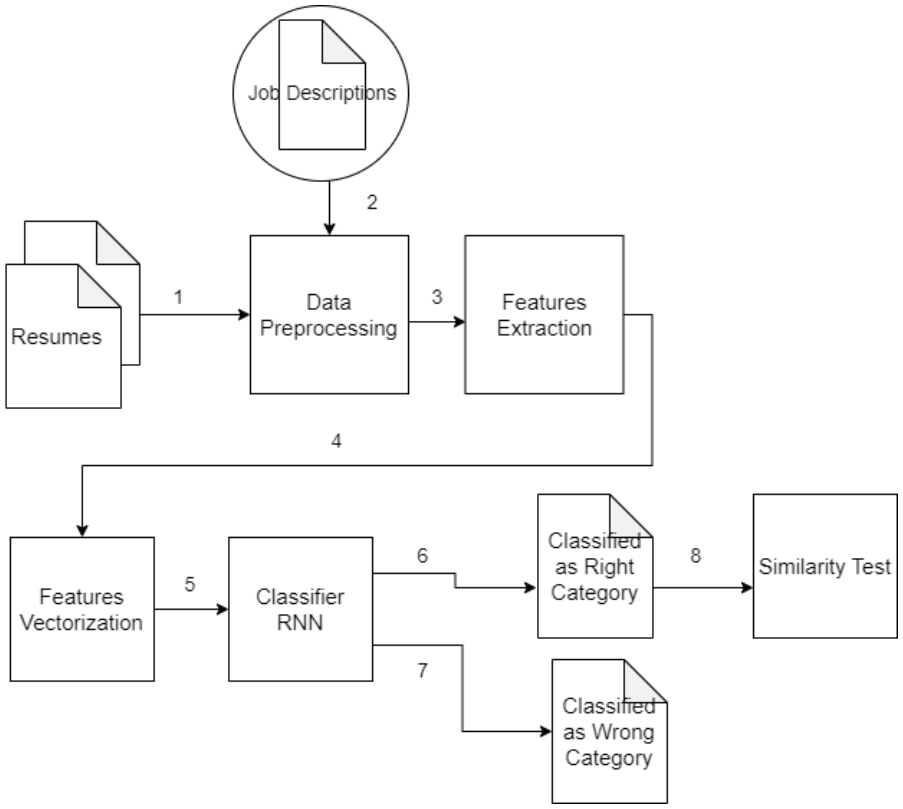

Figure 1 Resume classification and recommendation workflow

In the workflow, the first step is to preprocess both the candidate's resume and the labeled job description. Preprocessing refers to performing certain tasks or transformations on the data to make it suitable for further analysis or processing.

In this context, preprocessing the candidate's resume involves applying

various techniques to extract relevant information from the resume. This may include removing any unnecessary formatting. Similarly, the labeled job description also undergoes preprocessing. This process involves cleaning the text by removing any unwanted characters.

- After preprocessing the candidate's resume and the labeled job description, the next step in the workflow is feature extraction from both documents. Feature extraction involves identifying and selecting the most relevant and informative attributes or characteristics from the data.

In the context of the candidate's resume, feature extraction aims to capture the key qualifications, skills, experience, and other relevant details that are important for the job application.

- classes of other categories not related to the job category will not go for the similarity testing.
- verifying the similarity of the right category with the cosine similarity.

B. Data Collection

When it comes to collecting data for resume recommendation using RNN classification and cosine similarity, there are various methods available. These methods offer different possibilities for gathering the necessary information. Some common approaches to collect data include:

1. Fields Filling

One method of data collection for resume recommendation is through fields filling. This approach involves obtaining data by filling out specific fields or sections within a resume. In this method, individuals or users are typically presented with a form or interface that prompts them to input relevant information about their qualifications, skills, work experience, education, and other relevant details. These fields are carefully designed to capture the necessary information needed for the resume recommendation system.

Users are usually guided through the process of filling out each field, which may include providing details such as job titles, companies worked for, dates of employment, job responsibilities, educational institutions attended, degrees obtained,

and so on. The form may also include optional fields for additional information that could enhance the accuracy and effectiveness of the recommendation system.

2. Files Scraping

Another method for data collection in the context of resume recommendation is file scraping. This approach involves extracting relevant data from resumes or CVs in various file formats.

With file scraping, the system is designed to automatically parse and extract information from resumes in formats such as PDF, Word documents, or plain text files. This technique utilizes algorithms and techniques to identify and extract specific data points from the resume files.

C. Data Preparation

Data Preparation or Preprocessing is described as the process of transforming unstructured data into a format suitable for creating and training machine learning models. It is emphasized that the success rate of the task is directly impacted by this crucial step. The quality of the data plays a direct role on the performance of the machine learning model. Therefore, data preparation, also known as preprocessing, plays a crucial role in machine learning projects. This is particularly important in the case of Natural Language Processing (NLP) projects.

Data preparation involves a series of steps that are essential to make sure the used data is in a good fit the model. These steps typically include:

1. Text cleaning

Removing irrelevant characters, special symbols, or punctuation marks from the text data. Note that the text cleaning depends on the specific studied task for the model to learn well the case. By eliminating these unnecessary elements, the data becomes cleaner and more focused, allowing the model to concentrate on extracting meaningful patterns and relationships from the text.

2. Feature Extraction

Feature Extraction improves the efficiency and accuracy of machine learning to better serve their intended purpose. Feature selection is a critical step in data

preparation where the most relevant and informative features are selected from the available dataset. In many cases, datasets may contain numerous structured or unstructured features, some of which may be redundant or irrelevant to the target variable. Feature selection aims to identify and retain only the most valuable features that contribute significantly to the model effectiveness. Triggering feature selection involves various techniques such as statistical tests, correlation analysis, or machine learning algorithms designed specifically for this purpose. By carefully choosing the subset of relevant features, feature selection helps reduce the dimensionality of the data, mitigates the risk of overfitting, and improves the model's efficiency and generalization performance.

3. Tokenization

Tokenization is a fundamental process that break down textual data into more granular components, known as tokens. It helps models understand the semantic structure of the text, enabling it to analyze and process the data more effectively. By splitting the text into tokens, the model gains a better understanding of the individual components of the text and their relationships within the context of the NLP task at hand.

Example: text data: “the chosen file was so correct that I could not talk bad about it “tokenizes data:[“the”,”chosen”,”file”,”was”,”so”,”correct”,”that”,”I”,”could”,”not”,”talk ”,”bad”,”about”,”it”]

4. Stop Words Removal

In NLP projects, certain words, such as "the," "is," and "and" are often used in English but do not have significant contribution in the context of the specific NLP task. Removing these stop words can help reduce noise in the data and focus the model's attention on the more relevant and informative words that contribute to the overall meaning of the text. In addition, a custom stop words removal can be defined depending on the use case, for example in the case of website classification based on the description some stop words can be “websites”,” webs”, “platform” etc.….

5. Lemmatization or Stemming

Lemmatization and stemming are both techniques used to reduce words to

their base or root form. Lemmatization aims to convert words to their base forms, while stemming involves removing prefixes or suffixes to obtain the base form of words. By performing lemmatization or stemming, the model can capture the core meaning of words and reduce the potential duplication of information caused by different grammatical forms of the same word.

6. Handling Case Sensitivity

In NLP projects, words with different capitalizations may carry the same meaning. To avoid treating the same word differently based on its case, it is common practice to convert all text to a consistent case, such as lower case. This ensures that words with similar meanings are treated as identical, regardless of their capitalization, allowing the model to focus on the semantic content of the text rather than superficial variations.

7. Feature Vectorization

Feature vectorization, or feature encoding, is the process of converting textual or categorical features into numerical representations that can be understood and processed by machine learning algorithms. This step is crucial when dealing with non-numeric data, such as text, categories, or nominal variables.

Feature vectorization involves different methods depending on the nature of the data. For textual data, Techniques such as bag-of-words or term frequency-inverse document frequency (TF-IDF) are commonly employed in various natural language processing tasks. These methods convert text into numerical vectors based on word frequencies or statistical relevance within the document corpus.

For categorical variables, one-hot encoding or label encoding can be employed. One-hot encoding converts each category into a binary vector, where each feature represents the presence or absence of a particular category. Label encoding assigns a unique numerical label to each category.

8. Data Splitting

Data splitting plays a crucial role in preparing data for machine learning tasks. It involves dividing the available dataset into separate subsets, commonly referred to as the training set, validation set and test set. Each subset serves a specific

purpose in the model development and evaluation process.

a. Training Set

The training set constitutes the largest portion of the dataset and serves the purpose of training the machine learning model. It has labeled samples the model learns from to identify patterns, relationships, and statistical trends. The model adjusts its internal parameters based on the input features and their corresponding labels in the training set. A larger training set provides more data for the model to learn from, improving its ability to generalize and make accurate predictions.

b. Validation Set

The validation set is used to fine-tune the model during the training process and optimize its hyperparameters. It helps in assessing the model's performance on unseen data and provides feedback on its generalization ability. By evaluating the model's performance on the validation set, various hyperparameter configurations can be compared, and the best-performing model can be selected. The validation set helps prevent overfitting, where the model becomes too specific to the training data and fails to generalize well to new, unseen data.

c. Test Set

The test set is a separate subset of the dataset that is used to evaluate the final performance of the trained model. It represents real-world, unseen data that the model is expected to make predictions on. By evaluating the model on the test set, an unbiased assessment of its performance and generalization ability can be obtained. The test set should not be used during the model development or hyperparameter tuning process, as it would introduce bias and overestimate the model's performance.

D. Algorithms

1. Recurrent Neural Network

The RNN (Recurrent Neural Network) is an artificial neural network type specifically designed for processing sequential data or time series data. Consequently, this deep learning technique finds common application in solving ordinal or temporal problems, such as language translation, natural language processing (NLP), speech recognition, and image captioning. The Recurrent Neural

Network (RNN) can also be defined has a deep learning model widely used for processing sequential data, which can be structured or unstructured. It finds applications in various domains, such as speech recognition, sentiment analysis, and video activity recognition. Unlike traditional neural network models, RNNs can handle inputs and outputs of varying sizes and lengths. One key distinction between RNNs and traditional neural networks is that the latter have fixed input and output sizes determined at the model's creation, while RNNs can accommodate variable input and output sizes. This flexibility allows RNNs to effectively model sequential data, where the length of each sequence may vary. RNNs are designed with different architectures, depending on the task at hand. Two common architectures are the many-to-one and many-to-many. In the many-to-one architecture, the RNN takes in a sequence of inputs and produces a single output. This type of architecture is often used in sentiment analysis, where the sentiment of an entire text needs to be determined based on its sequential components.

On the other hand, the many-to-many architecture of RNNs generates a sequence of outputs corresponding to a sequence of inputs. In this case, the output at each step depends on the attention given to the input sequence. An example of the many-to-many architecture is language translation, where the RNN translates a sentence from one language to another by producing a sequence of translated words that align with the input sequence.

The versatility of the RNN model lies in its ability to capture dependencies and patterns across sequential data, making it well-suited for tasks that involve analyzing and processing sequences. By leveraging the recurrent connections within the network, RNNs can effectively learn and utilize context information from the past inputs to inform predictions or classifications at each step of the sequence.

2. Linear Support Vector Machine

The Linear Support Vector Machine (SVM) stands as a potent and extensively utilized machine learning algorithm, particularly proficient in binary classification tasks. It possesses a reputation for adeptly handling high-dimensional data and generating precise decision boundaries. To tackle multiclass problems, the one-vs-rest approach is commonly employed, wherein K linear SVMs are independently trained, with the data from the other classes serving as negative cases.

The primary objective of the Linear SVM is to identify an optimal hyperplane within the feature space, effectively maximizing the separation between instances belonging to different classes. This hyperplane serves as a decisive boundary, enabling the SVM to classify new instances based on their relative position with respect to the hyperplane.

The distinguishing characteristic of the Linear SVM lies in its formulation as a linear classifier within the feature space. It seeks to identify the most optimal hyperplane that maximizes the margin, representing the distance between the hyperplane and the nearest instances from each class. Through this process of maximizing the margin, the SVM establishes a robust decision boundary that is less influenced by noise, leading to improved generalization performance.

3. Gaussian N Bayes

The Gaussian Naive Bayes model is a widely adopted and efficient algorithm used for classification tasks, especially when handling continuous features assumed to follow a Gaussian (normal) distribution. It is based on the principle of Naive Bayes, which assumes that the features are conditionally independent given the class labels.

The Gaussian Naive Bayes model is particularly well-suited for problems involving many features. It exhibits excellent scalability, enabling fast training and prediction times, making it an efficient choice for such scenarios. The Gaussian Naive Bayes model estimates the probability distribution of each feature in every class using the training data.

The Naive Bayes classifier utilizes Bayes' theorem, which states that the posterior probability of a class given the observed features is proportional to the prior probability of the class multiplied by the likelihood of the features given the class. In the case of Gaussian Naive Bayes, it assumes that the likelihood follows a Gaussian distribution for each feature and each class.

4. Random Forest

The Random Forest is a potent and versatile machine learning algorithm extensively employed for regression and classification tasks. As a part of the ensemble learning family, it combines multiple individual models to produce more

accurate predictions.

he Random Forest algorithm forms an ensemble of decision trees, with each tree trained on a randomly selected subset of the training data. These individual decision trees are often referred to as "weak learners" due to their limited predictive power in isolation. However, as an ensemble, they collaborate to create a robust and powerful model with improved performance.

The Random Forest algorithm introduces randomness in two key aspects: data sampling and features selection. During the construction of each decision tree, a random subset of the training data, known as bootstrapped samples, is used. This process, called bagging, helps to introduce diversity in the training data and reduce overfitting.

When making predictions, the Random Forest combines the predictions of all the individual trees. For classification tasks, the mode of the predicted class labels is taken as the final prediction, while for regression tasks, the average of the predicted values is used.

5. Cosine Similarity

Cosine similarity serves as a widely used metric to gauge the similarity or dissimilarity between two vectors in a high-dimensional space. It finds extensive application in various natural language processing (NLP) and information retrieval tasks, including resume recommendation. By calculating the cosine of the angle between two term vectors, the similarity between two documents can be derived.

Cosine similarity computes the cosine of the angle formed by two vectors, representing how closely they align in the vector space. These vectors can represent diverse entities, such as documents, sentences, or words, and their similarity is determined by their alignment.

In the context of resume recommendation, cosine similarity can assess the similarity between resumes based on their textual content. Each resume is typically transformed into a vector, where each dimension corresponds to a specific term or feature present in the resume. The vector values can represent different metrics, such as term frequency-inverse document frequency (TF-IDF) scores or word embeddings.

The resulting similarity score ranges from -1 to 1, with 1 indicating that the two vectors are identical, 0 indicating they are orthogonal, and -1 suggesting they are diametrically opposed.

6. Bert Model

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art pre-trained natural language processing model that has significantly advanced various NLP tasks. Developed by Google AI, BERT is part of the transformer-based architecture family, which has revolutionized the field of NLP.

BERT can be used on a wide variety of language tasks: sentiments analysis, question answering, text prediction, text generation, summarization.

From the examples of tasks of the Bert model we could conclude that the Bert model is an LLM (Large Language Model).

E. Evaluation

In machine learning applications after building the models there comes the evaluation process where you will evaluate if the model you are working with is going to be good on unseen data in the future, the evaluation of models can vary from model to model, but the most used evaluation metrics are: Accuracy, confusion matrix , recall , precision , F1-score , addition to that you can use the loss function to evaluate how hard the model predict and ROUGE for transformer model in NLP problems.

1. Accuracy

Accuracy is a metric that assesses the overall correctness of the model's predictions by comparing them to the true labels. It is computed as the ratio of correct predictions to the total number of predictions.

2. Confusion Matrix

A confusion matrix presents a comprehensive breakdown of the model's performance by revealing true positive, true negative, false positive, and false negative predictions. This matrix provides valuable insights into the model's performance for each class, aiding in the identification of potential areas for

improvement. For instance, it can help determine whether data augmentation is required for a specific class or if other transformations are necessary.

3. Recall

Recall, also referred to as sensitivity or true positive rate, gauges the model's capacity to accurately identify positive instances among all actual positive instances. It is calculated as the ratio of true positives to the sum of true positives and false negatives.

4. Precision

Precision measures the accuracy of positive predictions made by the model. It is calculated as the ratio of true positives to the sum of true positives and false positives. Precision provides insights into the model's ability to avoid false positive predictions.

5. F1-score

The F1-score is a metric that assesses the performance of a classification model by considering both precision and recall simultaneously. It is calculated as the harmonic mean of precision and recall, providing a single value that balances the two metrics. The F1-score is particularly valuable in scenarios where there is a class imbalance or when precision and recall hold equal importance as evaluation criteria.

Furthermore, the selection of evaluation metrics depends on the specific use case being addressed. Different applications may require different metrics to effectively evaluate the model's performance and suitability for the given task.

IV. IMPLEMENTATION

In this chapter, we will provide a detailed explanation of the implementation process for our study. We will discuss the steps taken to gather the necessary data and provide insights into the implementation of the primary model, which is the Recurrent Neural Network (RNN). Additionally, we will explore the implementation details of other models used for comparison with the RNN.

Furthermore, we will delve into the data preprocessing phase, where we describe the steps undertaken to prepare the data for analysis. Specifically, we will elaborate on how we implemented the cosine similarity measure, which played a crucial role in ranking the resumes based on their similarity.

The primary objective of this chapter is to provide a comprehensive overview of the methods employed throughout the study. By sharing these implementation details, we aim to ensure the accuracy and reliability of our research findings. Readers will gain insights into the practical aspects of the study, enabling a deeper understanding of the experimental process.

A. Data Understanding

Table 1 attributes of first row of the dataset

	Category	Resume
0	Data Science	Skills * Programming Languages: Python (pandas...
1	Data Science	Education Details \r\nMay 2013 to May 2017 B.E...
2	Data Science	Areas of Interest Deep Learning, Control Syste...
3	Data Science	Skills â□¢ R â□¢ Python â□¢ SAP HANA â□¢ Table...
4	Data Science	Education Details \r\n MCA YMCAUST, Faridab...
5	Data Science	SKILLS C Basics, IOT, Python, MATLAB, Data Sci...
6	Data Science	Skills â□¢ Python â□¢ Tableau â□¢ Data Visuali...
7	Data Science	Education Details \r\n B.Tech Rayat and Bahr...
8	Data Science	Personal Skills â□¢ Ability to quickly grasp t...
9	Data Science	Expertise â□□ Data and Quantitative Analysis â...
10	HR	TECHNICAL SKILLS â□¢ Typewriting â□¢ TORA â□¢ ...
11	HR	I.T. Skills â□¢ Windows XP, Ms Office (Word, E...
12	HR	Education Details \r\n BA mumbai UniversityH...
13	HR	Education Details \r\nJune 2012 to May 2015 B....
14	HR	Education Details \r\nJune 2012 to May 2015 B....

The dataset used for this study was gather from Kaggle [20] which was in csv format, the target of the dataset is a multiple classification, so this is a multiple classification problem since we have more targets which represents the resume categories estimated about 25 categories, the data consist of 169 resumes and the distribution of each category is shown on the bellow figure.

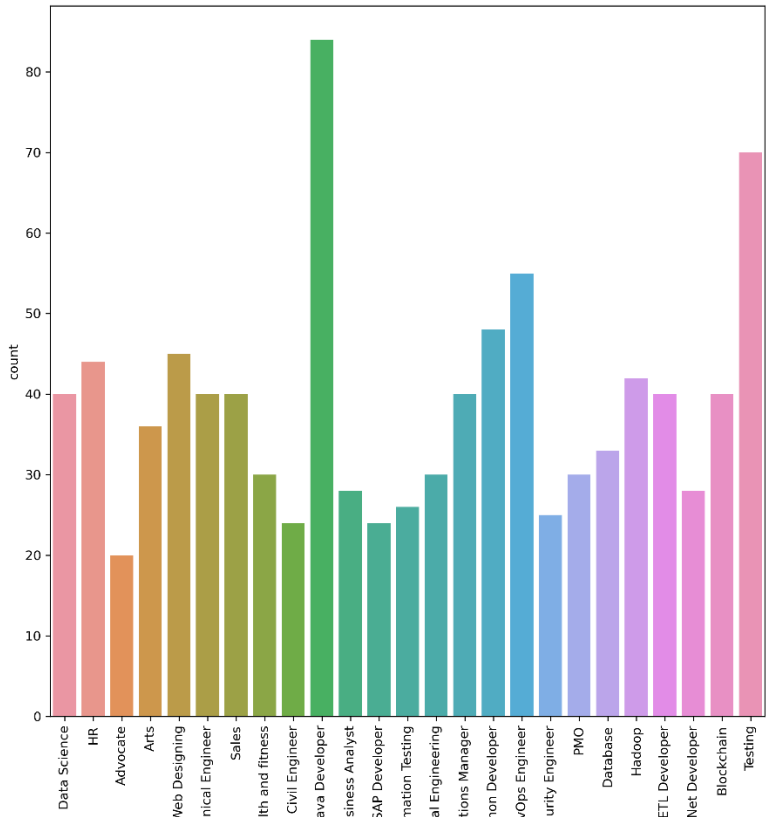

Figure 2 Data distribution over the different job classes

As we can see from the figure, we can see that the three categories that have the most resumes are Java developer, Testing and DeVops Eng.

We only have two attributes for the dataset which are category (represents the class of each resume) and Resume (which represents the corpus of text of different resume)

B. B. Feature Extraction

Feature extraction also called feature selection Is an important process in modelling. Since our data contains only one main feature the resume descriptions. To

build a more accurate predictive model, we need to extract additional features such as skills, education, and work experience of the job applicants. To extract these features, we developed a custom function that utilizes NLP techniques. The function first identified sections of the resume where there might be information about features: skills, education, and work experience, and then extracted these features from the corresponding sections. For skills, the function identified relevant keywords and phrases using techniques such as part-of-speech tagging and named entity recognition. For education, the function identified the highest degree attained and the field of study, as for work experience, we get the different domain of work experience the candidates have because those work experiences might be related to the candidate main field or not. For work experience, in the future we will improve the function to identify the number of years of experience and the job titles held.

```
import re

def extract_text_blocks(text):
     skill = re.compile(r'Skills. *? (? =Skills|Work Experence|Education|$)', re.DOTALL)
     work = re. compile(r'Work Experience.*?(?=Work Experience|Skills|Education|$)', re.DOTALL)
     education=re.compile(r'Education.*?(?=Education|Skills|Work   Experience|$)', re.DOTALL)
     Skills = re.findall(skill, text)
     Work=re.findall(work, text)
     Education=re.findall(education, text)
     return Skills,Work,Education
```

Figure 3 Feature extraction code

1. Data Preprocessing

in our use case the data preprocessing done is showed on the following code where we removed special characters, put text to lower cases, and removing stop words, and remember this process is applied on the selected features.

```
import re
import string
import nltk
nltk.download('stopwords')
stopwords = nltk.corpus.stopwords.words('english')
def clean_text(text):
    #convert text to lowercase
    text = text.lower()
    #remove any numeric characters
    text = ''.join([word for word in text if not word.isdigit()])
#     text = [word for word in text if re.search("\d", word)== None]
    # remove URLs
    text = re.sub('http\S+\s*', ' ', text)
    # remove RT and cc
    text = re.sub('RT|cc', ' ', text)
    # remove hashtags
    text = re.sub('#\S+', '', text)
    # remove mentions
    text = re.sub('@\S+', '  ', text)
    #punctuations removal
    text = "".join([word for word in text if word not in string.punctuation])
    text = re.sub("\W", " ", str(text))
    #stopwords removal
    ext = [word for word in text.split() if word not in stopwords]
    #replace consecutive non-ASCII characters with a space
    text = re.sub(r'[^\x00-\x7f]',r' ', text)
    #extra whitespace removal
    text = re.sub('\s+', ' ', text)
    return text
```

Figure 4 Data cleaning code

after applying this function on the dataset, we get the following table that contains the clean resume.

Table 2 Dataset before and after cleaning

Category	Resume	Cleaned text
Data science	Skills * Programming languages: python(pandas….	skills programming languages python pandas….
Data science	Education Details \r\nMay 2013 to May 2017….	education details may to may …….
Data science	Area of Interest Deep Learning. Control System.....	area of interest deep learning control system….

C. Feature Vectorization and Label Encoding

We used the TextVectorization layer from tf.keras.layers module for feature vectorization in our deep learning model. Especially we used the max_tokens parameters to set the maximum number of unique tokens(word) to consider it in the

vocabulary, and it was set to 20000 in our case. This layer is commonly used in natural language processing tasks to convert text inputs into numerical vectors that can be fed into a deep learning model. The TF-IDF (Term Frequency Inverse Document Frequency) is utilized for text vectorization.

For the categories of the dataset we needed to put in into numerical value in order for the model to understand the labels in order to achieve that we used the label encoder from sklearn :

```
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
label_encoder.fit(data["Category"])
numeric_labels=label_encoder.transform(data["Category"])
```

Figure 5 Label encoder code

D. Classification

This phase represents the implementation of our model for the classification phase and the comparison of it to other models. Those other models that we compared to our models are.: Linear Support Vector (LSV), Gaussian N Bayes, Random Forest.

The architecture of our RNN model for resume classification is based on the Many-to-one approach. The many-to-one architecture is efficient and effective because it processes the entire input sequence and produces a single output label, rather than producing an output at each time step as in the many-to-many architecture. tensorflow.keras.callbacks. The RNN architecture is presented in Figure.

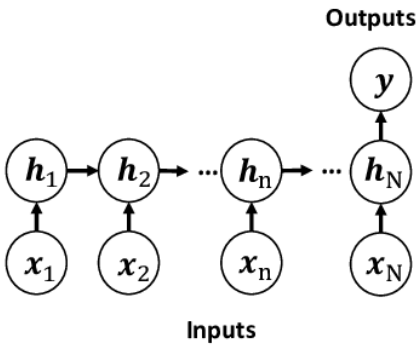

Figure 6 Many-To-One RNN Architecture

1. RNN Parameters

- number of layers: 1 (SimpleRNN layers).
- number of units in each layer: 128.
- activation function: 'relu' (in dense layers) and default activation function (tanh) for simple RNN layers.
- learning rate: 0.001 (in Adam optimizer).

2. Functions

The input function processes the input by applying the encoder layer to convert text to numerical, ***the output function* g**enerates the output based on the final Dense Layer with a SoftMax activation function, which predicts the probability distribution across the 25 possible Categories, and ***the hidden function*** maintains the internal state of the model by updating the current state based on the previous state and the current input using the SimpleRNN layers.

3. Training

The RNN model is trained to classify resumes into one of 25 categories using a categorical cross-entropy loss function and the Adam optimizer. The training data is represented as numerical vectors obtained from the encoder layer, and the labels are converted to one-hot encoded vectors using the to categorical function. Early stopping is employed to prevent over fitting, and a learning rate scheduler is used to adjust the learning rate during training. The model is trained for a maximum of 50 epochs with a batch size of 32.

4. Tools

We implemented our proposed deep learning model using Python and the TensorFlow library. The functions imported from TensorFlow are: Sequential which is imported from tensorflow.keras.models, [Dense, SimpleRNN, and Flatten] imported from tensorflow.keras.layers, [Early Stopping, LearningRateScheduler] imported from.

E. Similarity

In this part we used the cosine similarity to measure the similarity between the job description and the resumes and after that the resumes that have closer similarity to the job description will be rank, note that this similarity will be checked only if the resume passes the category checking test.

The used package to calculate the cosine similarity is TensorFlow and CountVectorizer from sklearn.feature_extraction.text. The following table illustrates it for an example.

Table 3 Similarity score example

Resume	Cosine Similarity	Distance similarity
1st resume	0.26	0.74
2 resume	0.24	0.76
3 resume	0.32	0.68

V. EXPERIMET RESULTS & EVALUATION

A. Evaluation Metrics

To evaluate the performance of our RNN model, we compared it to three other commonly used machine learning models: Gaussian Naive Bayes, Linear Support Vector Machines, and Random Forest classifiers and a generative model which is BERT. We used the same dataset to train and test each of these models and evaluated their accuracy, their Precision, F1 and the Recall. The results are presented in the following table:

Table 4 Comparison of Evaluation metrics

Models	Accuracy	Precision	Recall	F1-score
RNN	0.993	0.992	0.996	0.993
Bert	0.993	0.993	0.98	0.98
L SVM	0.992	0.90	0.994	0.992
Gaussian N Bayes	0.992	0.90	0.994	0.992
Random forest	0.867	0.865	0.858	0.867

The table presents the evaluation results, highlighting the performance of each model across the indicated metrics. The accuracy metric indicates the overall correctness of the model's predictions, while precision focuses on the accuracy of positive predictions. The F1-score represents the harmonic mean of precision and recall, providing a balanced measure. Recall assesses the ability of the model to correctly identify positive instances.

The fact that the performance of our model is almost the same as the BERT is that we trained the BERT model in 500 epochs while our model was trained in only 50 epochs. That show that our RNN architecture is efficient in terms of computational resource.

These evaluation metrics provide a comprehensive assessment of the

performance of each model, enabling a comparison of their effectiveness in the given task. The results suggest that the RNN model performed the best in terms of accuracy, precision, F1-score, and recall, these findings will further contribute to the validation and interpretation of our study's outcomes.

B. Confusion Matrix

The confusion matrix is a crucial tool that can help us see where errors in the model were made. It is table where the rows represent the actual categories the outcomes should have been, the columns represent the predicted categories we have made. It can easily be visualized by utilizing a heatmap plot from the Seaborn library. It is given in Fig. 4.

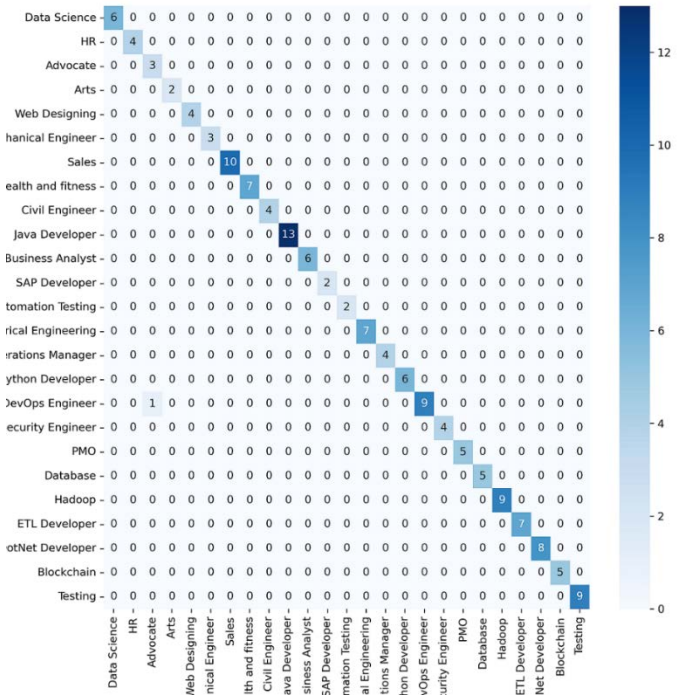

Figure 7 Confusion Matrix of RNN Model

As seen from the Figure 6, most of the classes have been classified correctly, with only one class (DevOps Engineer) being misclassified only once. This is a positive result that indicates our model's high level of accuracy and reliability in predicting most of the classes.

VI. DISCUSSIONS & CONCLUSION

In this Thesis, we proposed a novel approach for the task of resume classification using a Recurrent Neural Network (RNN). Our RNN model was trained on a dataset of 962 resumes classified into 25 different categories. We used various hyper parameters tuning techniques to optimize the performance of our model. The results showed that our model achieved an accuracy of 99.3% on the testing dataset, which outperformed other commonly used models such as Gaussian Naive Bayes, Linear Support Vector Machines, and Random Forest classifiers. Furthermore, we applied cosine similarity on the classification results to measure the similarity between the input resume and the classified category. Our approach provides a promising solution for the task of resume classification, which can be used in various real-world applications such as job recruitment and career counseling. However, there is still room for improvement, and future work can explore the use of more advanced deep learning techniques such as Transformers or hybrid models that combine RNNs with Convolutional Neural Networks (CNNs) for better performance.

VII. REFERENCES

ARTICLES

Lorenzo Ricciardi Celsi, Jesus Fernando Cevallos, Moreno, Federico Kieffer; Valerio Paduano, **HR- Specific NLP for the Homogeneous Classification of Declared and Inferred Skills**, Applied Artificial Intelligence, 36:1, DOI: 10.1080/08839514.2022.2145639 (2022)

Disha Lamba, Shivam Goyal , V. Chitresh , Neha Gupta" **An Integrated System for Occupational Category Classification based on Resume and Job Matching**" International Conference on Innovative Computing and Communication(ICICC,2020) https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3607282

Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, Rocky Bhatia, **A Machine Learning approach for automation of Resume Recommendation system**, Procedia Computer Science, Volume 167, Pages 2318-2327, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.284 (2020)

Q. Xu, J. Zhang, Y. Zhu, B. Li, D. Guan and X. Wang, Block-**Level RNN Model for Resume Block Classification**; IEEE International Conference on Big Data (Big Data), pp.5855-5857, doi: 10.1109/BigData50022.2020.9377771, (2020)

Pal Riya; Shaikh Shahrukh; Satpute Swaraj; Bhagwat, Sumedha, **Resume Classification using various Machine Learning Algorithms**. ITM Web of Conferences. 44.03011. 10.1051/itmconf/20224403011, (2022)

Ali, I., Mughal, N., Khand Z. H., Ahmed J; Mujtaba, G., **Resume classification system using natural language processing and machine learning techniques**. Mehran University Research Journal of Engineering & Technology, 41(1), 65–9.https://search.informit.org/doi/10.3316/informit.263278216314684, (2022)

SW. Kim, JM. Gil, **Research paper classification systems based on TF-IDF and LDA schemes**. Hum.Cent. Comput. Inf. Sci. 9, 30 (2019). https://doi.org/10.1186/s13673-019-0192-7

P. Lavanya and E. Sasikala; **Deep Learning Techniques on Text Classification Using Natural Language Processing (NLP) In Social Healthcare Network**: A Comprehensive Survey; 3rd International Conference on Signal Processing and Communication (ICPSC), pp.603-609, doi:10.1109/ICSPC51351.2021.9451752, (2022)

K. Yu, G. Guan, and M. Zhou**, Resume information extraction with cascaded hybrid model**. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), pages 499– 506, https://doi.org/10.3115/1219840.1219902, 2005

Vedant Bhatia, Prateek Rawat, Ajit Kumar, Rajiv Ratn Shah: **End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT**". arXiv:1910.03089v2 [cs.IR] 15 Oct 2019 https://doi.org/10.48550/arXiv.1910.03089

CHEN ZHU,HENGSHU ZHU , HUI XIONG , CHAO MA , FANG XIE , PENGLIANG DING , PAN LI "**Person-Job Fit: Adapting the Right Talent for the Right Job with Joint Representation Learning**" https://doi.org/10.48550/arXiv.1810.04040

Sanjjushri Varshini, Ponshriharini V, Santhosh Kannan,SnekhaSuresh, Harshavardhan Ramesh,Rohith Mahadevan, Raja CSP Raman" **Turtle Score - Similarity Based Developer Analyzer**" May 11, 2022 chttps://doi.org/10.48550/arXiv.2205.04876Ilham Huseyinov, Omobola Okocha, "**A Machine Learning Approach To The Prediction Of Bank Customer Churn Problem**" 3rd International Informatics and Software Engineering, 10.1109/IISEC56263.2022.9998299, (2022)

Yichuan Tang" **Deep Learning using Linear Support Vector Machines**" Contribution to the ICML 2013 Challenges in Representation Learning Workshop https://doi.org/10.48550/arXiv.1306.0239

Rahutomo, Faisal, Teruaki Kitasuka, and Masayoshi Aritsugi ;**Semantic cosine similarity**; The 7th international student conference on advanced science

and tecnology ICAST. Vol. 4. No. 1, (2012).

G. Salton and C. Buckley, "**Term-weighting Approaches in Automatic Text Retrieval**," Information Processing and Manafement, vol.24, no.5, pp. 513-523, (1988) https://doi.org/10.1016/0306-4573(88)90021-0

ELECTRONIC SOURCES

https://www.snowflake.com/guides/featureextraction-machine-learning , last accessed 2023/05/21.

https://www.ibm.com/topics/recurrent-neural-networks, last accessed 2023/05/21.

Articles, https://wandb.ai/mukilan/BERT_Sentiment_Analysis/reports/An-Introduction-to-BERT-And-How-To-Use-It--VmlldzoyNTIyOTA1

https://huggingface.co/blog/bert-101

https://www.kaggle.com/datasets/dhainjeamita/resumedataset

RESUME

Name Surname: Issa Diallo

EDUCATION:

BACHELOR: 2020, FACULTE DES SCIENCE DHAR EL MAHRAZ, APPLIED MATHEMATICS.

MASTER :2023, ISTANBUL AYDIN USNIVERSITY, ARTIFICIAL INTELLIGENCE and DATA SCIENCE.

PROFESSIONAL EXPERIENCE AND AWARDS:

-Machine learning and Backend developer at TERZION|DX (TURKEY) (11/2022-07/2023)

- Internship Data Analyst at INSAT, (MALI) (07/2020-12/2020)

PUBLICATIONS FROM DISSERTATION, PRESENTATIONS AND PATENTS:

Ilham Huseyinov, Issa Diallo, Mhd Wasim raed “Resume Recommendation using RNN Classification and Cosine Similarity.” Proceedings of the Seventh International Scientific Conference” Intelligent Information Technologies for Industry” IITI’23 (paper accepted)