2023 Global Conference on Information Technologies and Communications (GCITC)
Karnataka, India. Dec 1-3, 2023

ResumAI: Revolutionizing Automated Resume Analysis and Recommendation with Multi-Model Intelligence

Prisly Mary Jacob*, Susan Jacob†, Jo Cheriyan‡, Lekshmi S Nair§
†*‡Department of Computer Science and Engineering, Saintgits College of Engineering, Kerala, India
§Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Amritapuri, India
*prisly.csb2024@saintgits.org, †susan.csb2024@saintgits.org
‡jo.cheriyan@saintgits.org, §lekshmisn@am.amrita.edu

2023 Global Conference on Information Technologies and Communications (GCITC) | 979-8-3503-0816-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/GCITC60406.2023.10426042

***Abstract*—Finding and organizing suitable candidates for a vacant job position can pose challenges, especially when there is a high volume of submissions. This can impede team growth as it becomes challenging to identify the most suitable individual in a timely manner. To address this issue, an automated system called "Resume Recommendation and Classification" can expedite the selection process and aid in decision-making, while also reducing the time-consuming task of fair screening and shortlisting. This research utilizes Natural Language Processing to collect resumes and extract the necessary information. Subsequently, the resumes are ranked using BERT and TF-IDF word embeddings to obtain similarity scores. A classification model is then developed to categorize resumes based on job designations. To categorize the resumes, various techniques were employed, including Multinomial Naive Bayes, Linear Support Vector Classifiers, and k-NN Classifiers. These approaches enable the classification of resumes based on specific criteria.**

***Index Terms*—Recommendation, Classification, BERT, NLP, Word Embedding, Similarity Score**

I. Introduction

Talent acquisition is critical and complex for Human Resources (HR) departments in the modern business landscape. The sheer scale of the Indian market adds a layer of complexity to the process. One pain point organizations face is assigning new member selection, recommendation, and categorization. HR departments rely on manual procedures to evaluate and analyze resumes, which could be more efficient and time-consuming. This article presents a system designed to address these challenges by automating the filtering, recommendation, and classification of resumes based on applicant talents, job experience, designations, and other relevant information.

The proposed system leverages Natural Language Processing (NLP) techniques to filter resumes [1]. NLP allows for extracting meaningful information from textual data, such as resumes, by analyzing the structure, context, and content [9]. By implementing NLP algorithms, the system can efficiently extract essential elements from resumes, including qualifications, skills, work experience, and education.

The system employs advanced techniques such as BERT (Bidirectional Encoder Representations from Transformers) and TF-IDF (Term Frequency-Inverse Document Frequency) word embeddings to streamline the recruitment process [2]. BERT, a state-of-the-art language model, captures the contextual relationships between words and produces rich text representations. Conversely, TF-IDF measures the importance of each word in a document relative to the entire corpus. By leveraging these techniques, the system calculates similarity scores between job descriptions and resumes, recommending top candidates.

The goal is to eliminate the need for manual crawling by extracting the most important keywords from the resume's unstructured language. Our method extracts significant job-related material (skills, experience, education) from uploaded candidate resumes using techniques like POS Tagging, stemming, tokenization, and Named Entity Recognition. The outcome is a CSV-formatted summary of each resume used for this resume screening system's following stage of processing activities.

The tokenization process begins after converting several resume formats into text. The meaning of the original text series is derived from these sentences. Dividing vast volumes of text into smaller tokens is known as tokenization. The whitespaces and punctuation are dropped or separated in order to do this. Tokens are broken down into individual words. By tokenizing any text, we may learn the number of words in the text, how often a particular keyword appears, and more. Data is converted into tokens through tokenization, using tools like the Natural Language (NLP) Toolkit, the spaCy [3]. Additional text processing operations like stemming, stop word removal, and lemmatization need tokenization as a precondition.

Stemming is the process of distilling sentences down to their stem or original form. It is a rudimentary heuristic strategy that involves cutting off the ends of words to achieve this goal and often eliminates derivational affixes. Assigning grammatical information to a word based on context and connections with other words in the sentence is known as "parts of speech tagging." Depending on where a word appears in the phrase, the part-of-speech tag defines whether it is a noun, pronoun, verb, adjective, or something else. These tags are used to parse sentences correctly and create knowledge bases for named entity recognition. Comparing a word's part of speech tags to its dictionary definition is less challenging than doing this. This is more difficult than simply mapping a phrase to its part-of-speech tags.

Named Entity Recognition separates chunks of disorganized text into defined categories such as people's names, skills, contact information, URLs, school qualifications, designation, job experience, and abilities to retrieve user in-

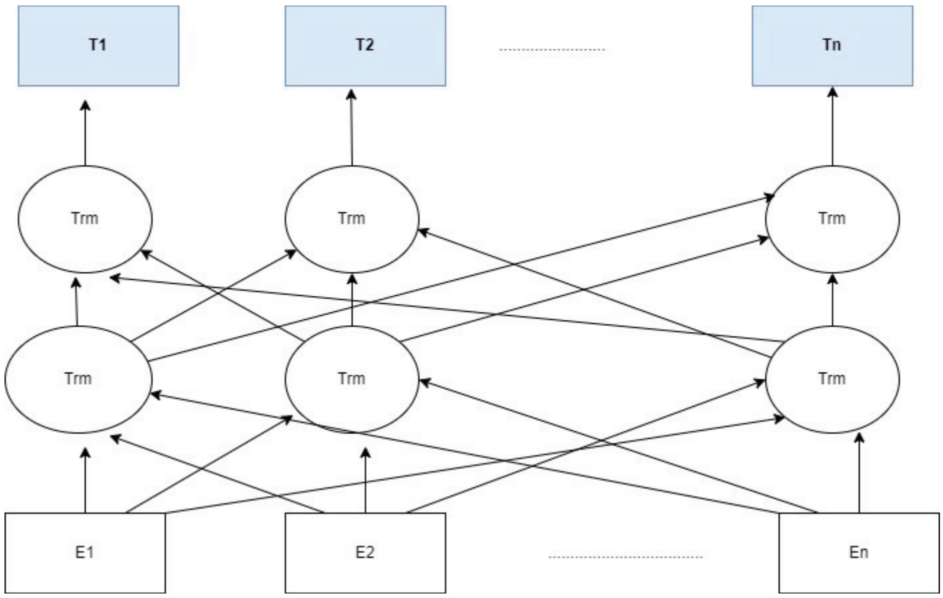


Fig. 1. BERT Architecture

formation. A recommendation model is a system of data filtration process that attempts to forecast how a client would assess or favor a particular item. Two prominent strategies for recommendation are collaborative filtering and content-based filtering. Recommendations from content-based filtering methods are connected with the topic of interest and its features. It is more of a suggestion in collaborative filtering since it is based on the interests of many people.

A statistic for assessing how related two data items are is the similarity measure. A similarity measure in a machine learning or data mining setting is a distance with a dimension representing object properties. The traits resemble more closely when the distance between them is small. Conversely, a considerable distance will result in a low degree of similarity. The five most used similarity measurements are Euclidean distance, Manhattan distance, Minkowski distance, Cosine Similarity, and Jaccard Similarity [4]. The Euclidean distance between two points is the length of the path connecting them. The Manhattan distance is a measurement that measures the distance between two points by adding their absolute differences in Cartesian coordinates. Put another way, it is the absolute difference between the x-axis and y. A metric variation of the Euclidean and Manhattan distances, the Minkowski distance is a distance between two points. To use the cosine similarity metric, we can find the normalization dot product of two attributes. By calculating the cosine of the angle between the two items, we would essentially be attempting to determine the cosine of the angle between two things. Every other angle's cosine is less than 1. However, the cosine of $0°$ is 1. By dividing the overall cardinality of set crossings by the cardinality of set unions, the Jaccard similarity calculates the similarity between finite sample sets.

We can apply the TF-IDF and BERT transformations to the texts to produce two real-valued vectors. BERT's real identity is bidirectional encoder representation from transformers since the decoder becomes unable to collect the directional encoder representation from transformers. The pre-train method, which incorporates a masked language model and next sentence prediction, and captures expression and sentence-level representation, is the model's significant originality. However, because interactive computation with BERT is complex, it is rarely used to compute similar text in downstream operations [5]. The BERT model architecture is shown in Figure 1.

Because machine learning techniques commonly use numerical data, any textual data or natural language processing work, a sub-field of ML/AI that deals with text, must first be vectorized. TF-IDF vectorization means calculating the TFIDF score of each word in our corpus connected to that document and then putting that information data into a vector. As a result, each document in our data would have its vector, with its TF-IDF score for each word. We use cosine to compare the TF-IDF vectors of two documents once we have these vectors [33].

Word2vec is another popular word embedding method. The vector creation technique is carried out by recognizing which words contain the target word most frequently across the full text. In this method, the semantic closeness of the phrases is underlined.

Classification is done using machine learning approaches [6]. Identifying, interpreting, and organizing ideas and objects into preset groupings or sub-populations is the classification process. Using post-training datasets and a variety of methodologies, machine learning systems categorize future datasets into categories. The classification issue in statistics is broad and may use several classification approaches depending on the data we are dealing with. Here are five of the most popular machine learning algorithms on the market.

- Logistic Regression: To forecast a binary outcome, logistic regression is a technique utilized. The binary result is produced by looking at separate components and categorizing the outcomes into one of two categories. Whether the independent components are category or numeric, the variable is always categorical.
- Naive Bayes: The Naive Bayes approach determines if a data point falls into a particular category. To ascertain

whether or not sentences and words match a specified "tag," text analysis may be utilized (categorization).
- K-Nearest Neighbors: A pattern recognition technique called K-nearest Neighbors uses training data to identify the k nearest k in in hypothetical future scenarios. When categorizing data with k-NN, choose the category to which the data belongs based on its closest neighbor. If $k = 1$, it will be distributed to the class that is nearest to 1. By a majority decision of its neighbors, K is categorized.
- Support vector machine: A support vector machine (SVM) is a sort of technology that goes beyond X/Y prediction by using methods to train and categorize data according to degrees of polarity.
- Random Forest: As a variant of the decision tree approach, the random forest algorithm uses training data to fit new data into one of the massive decision trees as a random forest. Data is connected to the nearest tree based on data size by averaging it. Random forest methods are beneficial because they remove the possibility of decision trees mistakenly labeling data points.

The following is a summary of the article's structure: Section II includes a summary of the literature. Section III describes the proposed framework for resume recommendation and classification methods. The discussion about the result obtained in the resume classification by various machine learning process is shown in IV. Finally, the article concludes in section V.

II. Related Work

The increasing number of job seekers has led to a significant influx of resumes for each job posting, making it challenging for recruiters to identify the most qualified candidates [12] [13]. To overcome this information overload, efficient filtering and recommendation systems are required to extract critical information from the overwhelming amount of available data on the web [46]. Recommendation systems, such as collaborative filtering, have emerged as effective solutions for providing personalized information and services to users in various domains, including news, e-commerce, books, and movies [10], [11], [21], [44], [45]. Furthermore, the use of content-based approaches has proven successful in promoting unrated items and explaining recommendations based on the characteristics of the objects themselves [25], [26], [34], [38], [39].

Content-based filtering utilizes the information about the objects to provide recommendations. This approach offers the advantage of suggesting items to users with specific preferences and explaining the reasoning behind the recommendations [15], [27], [41]. Collaborative filtering, on the other hand, identifies users with similar preferences and makes suggestions based on their input. These recommendation techniques have been widely employed in various domains, including e-commerce, where they have proven to be effective in increasing sales and improving customer satisfaction [22], [24], [35].

Recent advancements in Artificial Intelligence (AI) and Machine Learning (ML) have contributed to significant breakthroughs in semantic analysis and text classification [17], [28], [29]. Deep Neural Networks (DNNs) have demonstrated exceptional capabilities in text input classification, leveraging multiple layers of neurons to enhance accuracy. However, traditional ML models have limitations when it comes to categorizing resumes outside the domains they were initially trained on. To address this, our proposed method utilizes Natural Language Processing (NLP) techniques to extract essential entities from resumes. We then employ AI and ML algorithms to develop a recommendation model based on similarity scores calculated using BERT (Bidirectional Encoder Representations from Transformers) and TF-IDF (Term Frequency-Inverse Document Frequency) word embeddings. Additionally, a classification model is built to categorize resumes based on job designations, utilizing algorithms such as Multinomial Naive Bayes, Bernoulli Naive Bayes, Linear Support Vector Classifier, and k-NN classifier [32].

To assess the performance of the classification models, we conducted comparative research using multiple algorithms [30], [36]. This evaluation helps determine the most accurate and efficient approach for categorizing resumes, considering factors such as skills, experience, and education. The computational intelligence can be used for many application domain such as social to biology, natural to technology [8]. The combination of AI, ML, and NLP techniques provides a powerful solution for enhancing resume filtering, recommendation, and classification processes. By automating these tasks, recruiters can efficiently identify the most qualified candidates from a large pool of applicants [37]. The integration of content-based and collaborative filtering approaches, along with advanced classification algorithms, contributes to personalized recommendations and accurate categorization. The proposed method offers significant potential for improving talent acquisition processes across various domains, resulting in streamlined recruitment and increased organizational efficiency.

III. Proposed Method

Finding suitable personnel quickly and efficiently has become a significant challenge for businesses today, mainly when resources and time are scarce. To address this, we improve the overall process: Selecting the best candidates from a pool of resumes: Identifying the most qualified candidates from numerous resumes is time-consuming. Automating this selection process would provide a solution to this challenge.

Understanding candidate curriculum vitae (CV): Making sense of the information in a candidate's CV is crucial to evaluating their qualifications and suitability for the job. Automating this task would streamline the process and ensure efficient evaluation of CVs. Assessing job fit: Before making a hiring decision, it is important to ensure that the candidate possesses the necessary skills and capabilities to perform the job effectively. Automating this assessment process would help verify job fit and improve decision-making.

The proposed solution involves developing a system capable of locating the appropriate CVs from vast databases, irrespective of the CV format, and extracting the information required from the screened resumes. Additionally, the system will include recommendation and classification models to effectively sort resumes based on specific requirements. This article aims to effectively identify the top candidate resumes from a large pool of applicants. We have developed

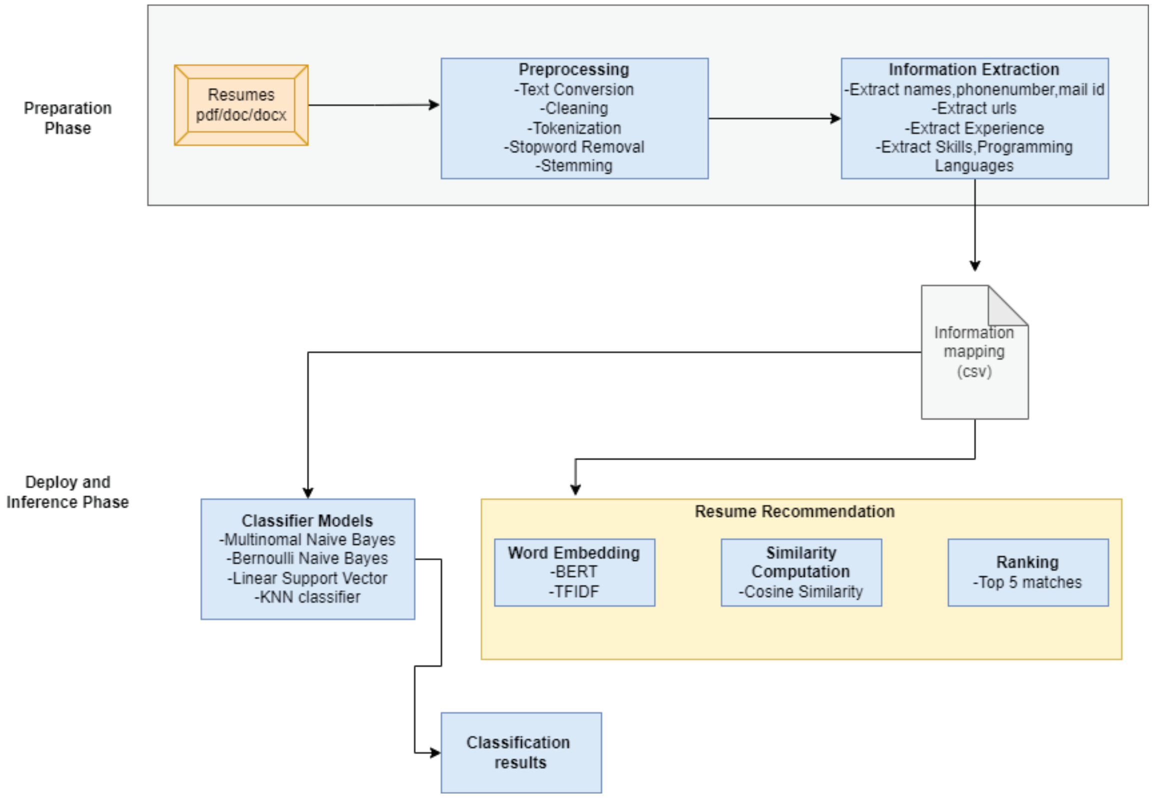


Fig. 2. A couple framework of the proposed model

an approach based on machine learning and deep learning techniques to achieve this. The framework for this approach is illustrated in Figure 2. The initial phase of the process is resume screening, which falls under the preparation phase. Companies receive numerous resumes in formats such as PDF, DOC, and DOCX for job postings. In our proposed model, we consider a bundle of resumes submitted in these formats as the input.

From these resumes, we perform entity extraction to extract the necessary information. This step, also known as information or entity extraction, involves using techniques such as Named Entity Recognition (NER). NER is an information extraction approach that categorizes segments of unstructured text into predefined categories. These categories may include names of individuals, contact information, educational qualifications, experience, and skills. The goal is to retrieve relevant and valuable information from the resumes.

By employing NER and other techniques, we aim to streamline the process of extracting essential entities from resumes, enabling efficient analysis and evaluation of candidate qualifications. The Figure 4 represents the extracted entity type and the NLP library used in-order to extract it from the text.

The initial step in information extraction involves the use of named entity recognition (NER) to identify and categorize named entities in the text. These entities can include names of people, organizations, locations, expressions of time, quantities, monetary values, percentages, and more. To perform NER, we utilize the open-source Python NLP software called spaCy, which is designed to extract data and process natural language. It provides a user-friendly API and is suitable for production use. In our case, spaCy is used to extract the names of individuals from the resumes.

Before performing the entity extraction, all the resumes are converted into a standardized text format using tools like pdfminer and doc2text. pdfminer is a software specifically designed to extract data from PDF documents, focusing on extracting and analyzing text data. It provides detailed information about the text placement, typefaces, and lines within a PDF. On the other hand, doc2text improves text extraction from images by correcting resolution, text field (crop), and skew, enhancing the accuracy of text extraction from photos.

Following the pre-processing phase, named entity extraction is performed. Contact information in the resumes is extracted using regular expressions (regex) and a URL extractor. Regular expressions are text strings that enable the design of patterns to match, locate, and manipulate text. They are widely used in languages like Perl. The URLExtractor is a Python class that extracts URLs from text using a Top-Level Domain (TLD) locator.

After extracting the entities, the information is mapped into a tabular format for further analysis. This mapped file serves as the basis for the next phase, the deploy and inference phase. In this phase, the resumes are ranked based on the extracted elements from the previous phase. The ranking process utilizes techniques such as vectorization and significance assigning algorithms like TF-IDF (Term Frequency-Inverse Document Frequency) and BERT (Bidirectional Encoder Representations from Transformers). Similarity metrics such as cosine distance are also employed to determine the similarity between the content of different documents. Vector

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Sl.No	Name	Mobile	Email ID	GitHub Lin	LinkedIN L	Education	Primary Skills	Secondary Skills	Programm	Total Expe	Company I	Designation
2	Person1	Abhijit Kun	97664106:	kumar.abh	nill	nill	BE,BS,X	Machine_learning	cloud	Python,C+	103	Company1	Data Scientist
3	Person2	Robotics P	81232301:	afzalhussa	nill	nill	BE,BS,X	Deeplearning,Dataanalytics,Robotics	Machine_learning,computervision	Python,C+	103	Company1:-19 Months,C	
4	Person3	Anand B	89047918:	anand.ma	github.con	linkedin.cc	ME,MTECH	Deeplearning	Machine_learning	Python,Jav	0	Company1	Senior Softwa

Fig. 3. Mapped entities pattern of few resumes

ENTITY TYPE	LIBRARIES USED
Names	SpaCy
Contact Information	Phone Number&Email ID:Regex LinkedIn Id &Github :URL Extractor
Educational Details&Experience	SpaCy,Regex,Datetime
Skills & Designation	Regex

Fig. 4. Entities extracted and the libraries used for extraction

TFIDF	BERT
0.95594	0.935621

Fig. 5. Average similarity score achieved for different word embedding followed by cosine similarity

space, a geometric structure generated by a set of components known as vectors, is used to represent the text information, aiding in information extraction, natural language processing, and text mining tasks.

In the second part of this phase, a classification model is developed to categorize the resumes based on their job designations. The mapped entities are used to generate a collection of tokens, which are then classified into appropriate designations based on the talents and skills mentioned. This classification ensures that applicants are assigned to the appropriate work functions.

To evaluate the effectiveness of the classification, a comparison study is conducted using different types of classifiers, including Multinomial Naive Bayes, Linear Support Vector, and k-NN classifiers. These classifiers provide insights into the best approach for accurately categorizing resumes based on job designations.

IV. Results

The proposed solution involves developing a system capable of locating the appropriate CVs from vast databases, irrespective of the CV format, and extracting the information required from the screened resumes. Additionally, the system will include recommendation and classification models to effectively sort resumes based on specific requirements. This article aims to effectively identify the top candidate resumes from a large pool of applicants. We have developed an approach based on machine learning and deep learning techniques to achieve this. Resumes are collected and subjected to preprocessing, extracting relevant entities. The extracted information is then organized and mapped for future reference. The Figure 3 illustrates the pattern of mapped entities derived from a subset of resumes.

Once the entities are mapped, the next step is to rank the resumes based on specific criteria, considering the extracted skills. For instance, if the desired skills are related to Natural Language Processing (NLP), the resumes will be ranked by evaluating their skills in this area, and the top 10 resumes will be considered. The ranking process involves assessing the cosine similarity score values. Two-word embedding techniques, namely pre-trained BERT and TF-IDF, are utilized to determine the most suitable score. The resume ranking shows a similar outcome in both cases, but the execution time is comparatively better when TF-IDF is employed. The Figure5 represents the comparison between the similarity score achieved using both methods.

The classification phase involves utilizing the Multinomial Naive Bayes classifier to predict the job designation based on the extracted and mapped features. The classifier achieves a result with an accuracy of 73.97 percent. Subsequently, we explored the performance of other classifiers, namely the k-NN classifier and Linear SVC. The accuracy measures of these classifiers are depicted in Figure 6. Notably, the Linear SVC exhibits significantly higher accuracy compared to the other classifiers. This classifier proves to be reliable and the most suitable for achieving our objective.

To further evaluate the performance of our best model (LinearSVC), we examined the confusion matrix, as illustrated in Figure 7. The confusion matrix highlights the differences between the predicted and actual labels, providing insights into the accuracy of the classification model.

The implementation of the proposed algorithm allows recruiters to efficiently identify resumes that align with the required skills and job requirements. This automated process enables the rapid review of thousands of resumes, surpassing the capabilities of a human recruiter in terms of speed and accuracy. By prioritizing the top candidates, the hiring process becomes more effective and efficient. This approach also reduces time and resource costs associated with manual resume screening. Furthermore, the algorithm facilitates the ranking of resumes based on their relevance to the job qualifications. This feature simplifies the recruiter's task by organizing the resumes in order of relevance, making it easier to identify the most suitable candidates. While the current model provides recommendations for multiple industries, it can be further improved by focusing on specific industries, enhancing its effectiveness and generating better recommendations.

Classifier	Accuracy
Multinomal Naive Bayes	0.7397
Linear Support Vector Machine Classifier	0.9863
KNN Classifier	0.7260

Fig. 6. Accuracy Measure for different classifiers

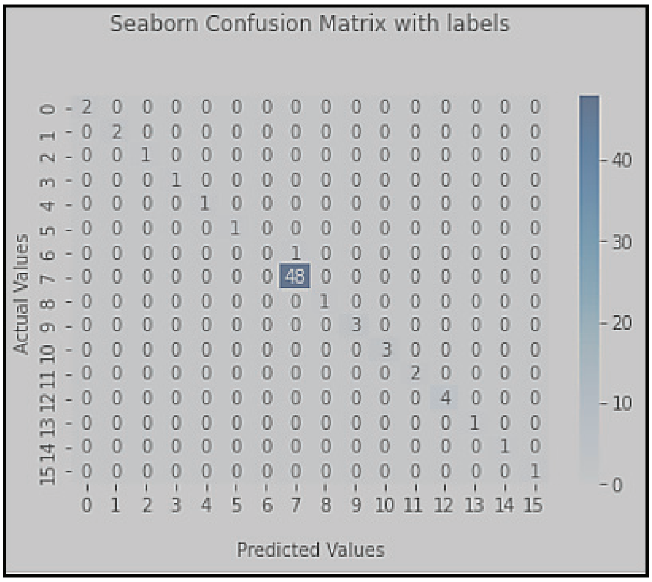


Fig. 7. Confusion matrix of Linear SVC

V. Conclusion

In conclusion, increasing applicants for each job position requires an automated solution to streamline the resume screening and categorization process. The proposed machine learning-based system effectively recommends resumes of qualified applicants by extracting relevant entities, categorizing resumes based on skills and designations, and providing accurate recommendations. The employed Linear SVM classifier achieves an impressive accuracy of 98.63 percent. Future improvements can involve industry-specific models created in collaboration with domain experts, such as HR professionals, to enhance the system based on their valuable feedback continuously.

References

[1] Sinha, A.K., Amir Khusru Akhtar, M. and Kumar, A., 2021. Resume screening using natural language processing and machine learning: A systematic review. Machine Learning And Information Processing: Proceedings Of ICMLIP 2020, pp.207-214.

[2] Walkowiak, T., 2021, May. Subject classification of texts in Polish-from TF-IDF to transformers. In International Conference on Dependability and Complex Systems (pp. 457-465). Cham: Springer International Publishing.

[3] Orosz, G., Szántó, Z., Berkecz, P., Szabó, G., & Farkas, R. (2022). HuSpaCy: an industrial-strength Hungarian natural language processing toolkit. arXiv preprint arXiv:2201.01956.

[4] Santini, S., & Jain, R. (1999). Similarity measures. IEEE Transactions on pattern analysis and machine Intelligence, 21(9), 871-883.

[5] Masala, M., Ruseti, S., & Dascalu, M. (2020, December). Robert–a romanian bert model. In Proceedings of the 28th International Conference on Computational Linguistics (pp. 6626-6637).

[6] Wu, H., Liu, Y., & Wang, J. (2020). Review of Text Classification Methods on Deep Learning. Computers, Materials & Continua, 63(3).

[7] Kang, Y., Cai, Z., Tan, C.W., Huang, Q. and Liu, H., 2020. Natural language processing (NLP) in management research: A literature review. Journal of Management Analytics, 7(2), pp.139-172.

[8] Nair LS, Cheriyan J. SIR-M Epidemic Model: A SARS-CoV-2 Perspective. InComputational Intelligence and Data Analytics: Proceedings of ICCIDA 2022 2022 Sep 2 (pp. 499-508). Singapore: Springer Nature Singapore.

[9] Reza MT, Zaman MS. Analyzing CV/resume using natural language processing and machine learning (Doctoral dissertation, BRAC University).

[10] Ricci, F., Rokach, L., & Shapira, B. (2010). Introduction to Recommender Systems Handbook. Recommender Systems Handbook, 1–35. doi:10.1007/978-0-387-85820-3-1

[11] Basu, C., Hirsh, H., & Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. In Proceedings of the 15th National Conference on Artificial Intelligence (pp. 714-720). AAAI Press.

[12] Cunningham, R. B. (2020). Resume sourcing: How recruiters find your resume. LinkedIn Talent Blog. Retrieved from https://business.linkedin.com/talent-solutions/blog/recruiting-tips/2020/resume-sourcing-how-recruiters-find-your-resume

[13] Bolles, R. N. (2019). What color is your parachute? 2020: A practical manual for job-hunters and career-changers. Ten Speed Press.

[14] Pu, P. (2007). Trust-based recommendation systems: An axiomatic approach. In Proceedings of the 2007 ACM Conference on Recommender Systems (pp. 49-56). ACM. DOI: 10.1145/1297231.1297242

[15] Lee, J. H., & Cho, K. S. (2012). A text categorization and information extraction approach to a content-based book recommendation system. Expert Systems with Applications, 39(2), 2070-2078. DOI: 10.1016/j.eswa.2011.08.040

[16] Das, A. S., Datar, M., Garg, A., & Rajaram, S. (2007). Google news personalization: Scalable online collaborative filtering. In Proceedings of the 16th International Conference on World Wide Web (pp. 271-280). ACM. DOI: 10.1145/1242572.1242610

[17] Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys (CSUR), 34(1), 1-47. DOI: 10.1145/505282.505283

[18] Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. IEEE Computational Intelligence Magazine, 9(2), 48-57. DOI: 10.1109/MCI.2014.2307227

[19] Chen, J., Du, N., & Sun, L. (2019). Deep learning-based text classification: A comprehensive review. Journal of Computer Science and Technology, 34(4), 631-661. DOI: 10.1007/s11390-019-1967-6

[20] Huang, Y., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991. Retrieved from https://arxiv.org/abs/1508.01991

[21] Resnick, P., & Varian, H. R. (1997). Recommender systems. Communications of the ACM, 40(3), 56-58. DOI: 10.1145/245108.245121

[22] Adomavicius, G., & 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, 17(6), 734-749. DOI: 10.1109/TKDE.2005.99

[23] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370. DOI: 10.1023/A:1021240730564

[24] Hsu, J. Y., Huang, L. K., & Yen, D. C. (2017). Exploring the impact of recommender systems on E-commerce sales. Information & Management, 54(7), 872-883. DOI: 10.1016/j.im.2017.03.008

[25] Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 13(5-6), 393-408. DOI: 10.1023/A:1006567501465

[26] Balabanović, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. Communications of the ACM, 40(3), 66-72. DOI: 10.1145/245108.245126

[27] Gulla, J. A., & Pasi, G. (1999). Enriching user profiles with extracted information: A step towards personalization of search engines. In International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (pp. 1097-1104). Springer. DOI: 10.1007/3-540-48728-2-136

[28] Niu, X., Zhou, X., & Zhang, Z. (2011). Collaborative filtering for personalized community-based question recommendation. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1147-1148). ACM. DOI: 10.1145/2009916.2010076

[29] Karypis, G. (2001). Evaluation of item-based top-N recommendation algorithms. In Proceedings of the 10th International Conference on Information and Knowledge Management (pp. 247-254). ACM. DOI: 10.1145/502585.502692

[30] Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882. Retrieved from https://arxiv.org/abs/1408.5882

[31] Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2018). Relation extraction using gated convolutional neural network. In Proceedings of the 27th International Conference on Computational Linguistics (pp. 233-244). Association for Computational Linguistics. DOI: 10.18653/v1/N18-1018

[32] Chen, J., Liu, Y., He, Y., & Li, Y. (2017). Distilling knowledge from deep networks with applications to healthcare domain. arXiv preprint arXiv:1711.09589. Retrieved from https://arxiv.org/abs/1711.09589

[33] Nair, Lekshmi S., and M. K. Shivani. "Knowledge graph based question answering system for remote school education." 2022 International Conference on Connected Systems & Intelligence (CSI). IEEE, 2022.

[34] Zhang, L., Yang, M., & Chen, X. (2015). Text classification based on deep learning and knowledge integration. Neurocomputing, 160, 23-31. DOI: 10.1016/j.neucom.2014.12.067

[35] Nair, L. S., Cheriyan, J., & Swaminathan, J. (2022). Microscopic structural analysis of complex networks: An empirical study using motifs. IEEE Access, 10, 33220-33229.

[36] Cheriyan, J., & Nair, J. J. (2022). Influence Minimization With Node Surveillance in Online Social Networks. IEEE Access, 10, 103610-103618.

[37] Nair, L. S., & Cheriyan, J. (2023, January). Multi-Featured Movie Recommendation Using Knowledge Graph. In 2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) (pp. 590-594). IEEE.

[38] Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. DOI: 10.1038/nature14539

[39] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems (pp. 3111-3119).

[40] Kotsiantis, S., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging Artificial Intelligence Applications in Computer Engineering, 160(1), 3-24. DOI: 10.1007/978-3-540-72689-7-1

[41] Yildirim, S. (2008). An investigation of content-based filtering methods for news recommendation. Knowledge-Based Systems, 21(8), 823-826. DOI: 10.1016/j.knosys.2008.06.001

[42] Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12), 61-70. DOI: 10.1145/138859.138867

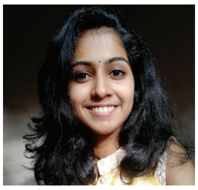
[43] Shapira, B., & Stoller, A. (1999). Automated collaborative filtering on Usenet news articles. In Proceedings of the 16th International Joint Conference on Artificial Intelligence (Vol. 1, pp. 235-240). Morgan Kaufmann Publishers. DOI: 10.5555/1624312.1624349

[44] Adomavicius, G., & 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, 17(6), 734-749. DOI: 10.1109/TKDE.2005.99

[45] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4), 331-370. DOI: 10.1023/A:1021240730564

[46] Liu, Y., Wang, Z., Li, S., & Li, M. (2018). A hybrid recommendation model based on collaborative filtering and deep learning. In Proceedings of the 27th International Joint Conference on Artificial Intelligence (pp. 3290-3296). AAAI Press. DOI: 10.24963/ijcai.2018/458

Prisly Mary Jacob , an undergraduate student at Kerala Technological University, pursuing B.Tech degree in Computer Science and Engineering, with Honors in Machine Learning. Her current research interests include machine learning and deep learning, specifically privacy-preserving and distributed machine learning. She has secured an Elite Gold & Silver badge in her studies and competative examinations conducted by MHRD, Govt of INDIA. She is currently the Vice President of the college coding club, CODE@Saintgits and Vice President of the PMI Kerala Student chapter of Saintgits College of Engineering.



Susan Jacob , an undergraduate student at Kerala Technological University, pursuing B.Tech degree in Computer Science and Engineering. She has secured prizes in various technological contests, like ideathon by the Digital University of Kerala. Her interests lie in machine learning and cloud computing. she actively contributes to the academic community, holding the position of secretary at CODE@Saintgits and serving as the Chief Marketing Officer for the Cybersecurity Club at Saintgits. Susan also fulfils the role of IE(I) representative.

Jo Cheriyan , received M.E. degree in Computer Science and Engineering from Sathyabama University, India in 2007. He is currently pursuing the Ph.D. in the Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham. He serves as a Faculty of Computer Science Engineering, Saintgits College of Engineering, India. He is a Senior member-IEEE and Member of IEEE-Computer Society. His research interests include Graph Mining, Complex Networks Analysis, Social Network Analysis, and Machine Learning.

Lekshmi S. Nair received her B.Tech and M.Tech degrees in Computer Science and Engineering from Anna University, Chennai, India in 2006 and 2012 respectively and doing PhD. in Complex Networks at Amrita Vishwa Vidyapeetham. Ms Nair also serves as Assistant Professor in Department of Computer Science, Amrita Vishwa Vidyapeetham. She has authored and co-authored many publications in journals and conferences. Ms Nair served as Technical Programme Committee Member and Reviewer for many conferences. She is a member IEEE and ACM professional member. Her research interests include Complex networks, Knowledge graphs and Ontology, Machine Learning.