Resume Matching Framework via Ranking and Sorting using NLP and Deep Learning

Senem Tanberk
Research and Innovation
Huawei Turkey Research and Development Center
Istanbul, Türkiye
senem.a.tanberk@gmail.com

Selahattin Serdar Helli
Research and Innovation
Huawei Turkey Research and Development Center
Istanbul, Türkiye
serdar.helli2@huawei-partners.com

Ege Kesim
Research and Innovation
Huawei Turkey Research and Development Center
Istanbul, Türkiye
ege.kesim1@huawei.com

Sena Nur Cavsak
Research and Innovation
Huawei Turkey Research and Development Center
Istanbul, Türkiye
sena.nur.cavsak@huawei.com

2023 8th International Conference on Computer Science and Engineering (UBMK) | 979-8-3503-4081-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/UBMK59864.2023.10286605

***Abstract*—Online job search through websites has been a remarkably advantageous tool for both job seekers and employers, effectively serving their needs for numerous years. The job-resume matching system uses natural language processing (NLP) techniques to analyze the content of resumes and job descriptions. In this study, we used AI models to analyze resumes and rank each resume based on its similarity to the job description. This proposed framework uses optical character recognition and object recognition models to detect text and sections from resumes. Afterward, the text preprocessing step is applied. The preprocessed section is classified with the BERT model. The section of resumes classified into 5 different categories (education, experience, skills, personal, and language) are sent to the information extraction module. By using the named entity recognition method, the module identifies 8 different categories in the text (title, degree, major, skill, language, city, country, and date). Based on extracted sections and named entities similarity between the job description and resume is calculated using cosine similarity. Each section's similarity score is calculated independently. This allows users to assign each section a weight based on the importance for the total score. Finally, in the developed web application, these weights are loaded into the system and resumes are compared with job applications to be sorted. The proposed resume matching framework accurately extracts relevant information from resumes, displays the similarity results of the candidates by ranking and sorting, and helps recruiters or managers to select the best-scored candidates.**

Keywords — *Job Resume Matching, Named Entity Recognition (NER), Optical Character Recognition (OCR), Text Classification*

I. Introduction

Today, Human Resource Management (HRM) key operations use technology to minimize manual workload, improve management, and save time. In the era of digital HR transformation, online applications have increased in frequency and become essential [1]. This makes it difficult for recruiters to match a job applicant's resume to the job description [2]. Effective resume screening requires an expert to assess a profile's suitability and applicability to the job description. In this work, we propose an appropriate framework by calculating the similarity between the job description and each resume and ranking candidates by similarity score.

Information extraction (IE), which is one of the purposes of natural language processing (NLP), has recently gained importance and is frequently used in the calculation of the similarity between job descriptions and resumes [3]. There are many studies in the literature in which information extraction is applied to resumes [4,5,6]. Trinh and Dang [4] introduced a machine-learning model that automates the process of matching candidates to job positions. This model utilizes a set of features extracted from a candidate's resume, categorizes them, and then maps the classified resume to the relevant job description. Then, suggests the profile of the most suitable candidate to the HR department. Guo *et al.* [5] proposed a system for matching job seekers with suitable employment opportunities. Their approach involved utilizing a knowledge extraction library based on finite-state converters to extract models from both job descriptions and resumes. Additionally, they devised a novel statistical similarity metric to compare the extracted resume models with the job models. Yu *et al*. [6] presented a cascading information extraction approach for resume information extraction to support automated resume management and referral.

Named entity recognition (NER) is the problem of finding the names of various entities such as addresses, personal and company names in texts. Although name entity recognition is a popular problem, entity name recognition studies in this area are limited due to limited open-source resume datasets. In most studies, resumes are collected and used over the Internet. Gaur *et al*. [7] introduced a model based on CNN-BILSTM architecture, aiming to identify degrees and institute names in resumes with high accuracy. Zu and Wang [8] developed an advanced model based on CNN-BILSTM-CRF architecture, which effectively recognized 19 different entity types using a dataset of 5000 resumes. Their research demonstrated that this model exhibited superior performance compared to using

BILSTM-CRF alone. Satheesh *et al.* [9] conducted a study where they designed an automated resume-scanning system. The system was capable of extracting suitable details from resumes and generating a score graph for each individual resume. This score played a crucial role in the selection process by identifying suitable candidates while eliminating unqualified applicants, thereby preventing an overload of irrelevant resumes. Apart from these, there are also Chinese resumes in the literature. Due to the existence of this data set, studies are also available [10,11].

Resume classification has become an important issue in recent years and research on this subject is increasing. Nasser *et al*. [12] created a CNN model to categorize 500 job descriptions and 2K resumes after hierarchically segmenting resumes into sub-domains, particularly for technical occupations. Gopalakrishna *et al*. [13] focused on classifying resumes in the IT sector. They discuss the design and execution of a resume classifier application that uses an ensemble-based voting classifier to classify a candidate's profile into a suitable domain based on the candidate's interest, work experience, and expertise mentioned in the profile.

Niti Khamker *et al*. [14] have implemented a system in which the candidate's resume and job description overlap and, as a result, show a share of similarity. It shows the candidates' similarity results to the recruiter, which helps the recruiter evaluate top-rated candidates. Faliagka *et al*. [15] demonstrated a novel method for ranking candidates in online achievement systems. The suggested approach takes a set of objective criteria from the candidates' LinkedIn profiles and infers their personality traits from their blog posts using linguistic analysis. In recent years, resume parsing and resume matching systems, which have been developed by taking advantage of NLP, Machine Learning, and Deep Learning, have been the focus of researchers [16, 17, 18].

This study's main contributions are three-fold:

(1) We propose a novel prototype of a resume-matching framework to enhance the recruitment product, making the recruitment procedure more efficient, optimizing the time-consuming process, and accurate.
(2) We combine state-of-the-art approaches in NLP, deep learning, and OCR in the framework architecture.
(3) we present a new approach to ranking and sorting resumes with a heuristic approach.

II. METHOD

A. *Proposed Framework*

This section covers detailed descriptions of the proposed framework for resume matching system including all components, input and output. The input of the resume matching system is raw data containing unstructured documents of resumes in pdf type. The output of the pipeline is in a structured format that includes candidates after ranking and sorting by analyzing all the sections and entities, then connected to a web application. The components of the framework are section recognition, section text conversion, section classification, similarity evaluation between job descriptions and resumes, and ranking & sorting to determine top candidates, respectively.

The business lifecycle for recruiting and hiring process starts with receiving resumes through various sources via cumulative applications in a flexible format. Then, these resumes are moved and stored in the HR database of the company. From job posting to sending job offers, HR professionals and employers spend time recruiting new employees with mapped talent needs. Our framework is versatile via components employing AI technology, so it can automate recruiting and hiring process by replacing repetitive and time-consuming tasks for them. The proposed framework takes resume files and the text of the job descriptions and then performs pipeline steps leveraging deep learning and machine learning algorithms to streamline the process. Finally, the final scores are computed by considering the weights of information extraction entered by users. The output of the proposed framework is the resume list on a web screen after the ranking and sorting process in order for each job description.

The flow in Figure 1 will be followed while describing the presented approach. The input of the system consists of a resume. This approach consists of 5 main stages. In the first stage, the section location is determined by object recognition. Then from each section texts are extracted with an OCR model. The third stage is the classification of the obtained text with the Bert model. The fourth stage is divided into two. For skill and experience sections we calculate similarity directly. The rest of the sections are fed into the NER model for further filtering. The similarity of these sections is calculated using extracted entities that are recognized in the resume and job description. In the last stage, the job application and the resume are ranked and matched based on similarity scores.

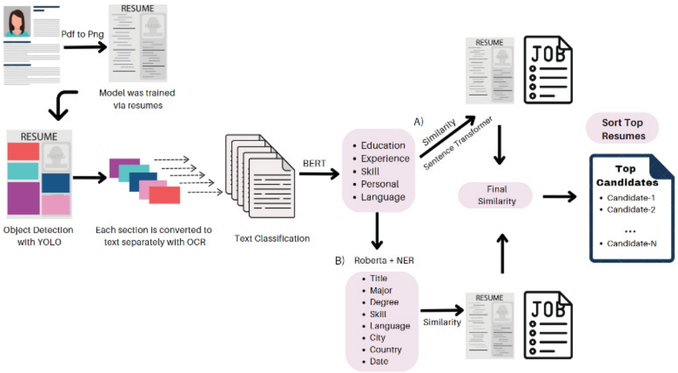


Fig. 1. Proposed Framework for Resume Matching System.

a) Optical Character Recognition and Object Recognition

The resume, which is a pdf/word file, is converted to png format. The purpose of converting to this format is that the YOLO model can work with this format. Then the text groups for each section in the resume should be classified. The YOLOV8 model, which is an object recognition approach, was used to detect the text section separately and was trained with this model. The reason for using object detection is to determine the location of each section in the resume. Texts are defined by giving the trained YOLOv8 model outputs to the pre-trained model for OCR [19, 20].

b) Section classification

In this work, text classification was done using the Bert model. The BERT [21] model is a bidirectional converter-based deep-learning language model that is trained in two stages. In the initial training phase, it is trained with tasks to predict the next sentence using unlabeled data and masked language modeling. Then, extra output layers are added to the model for different language problems, and the model is trained for problems such as entity name recognition and text classification. Here, it is aimed to classify text groups into different classes or categories. These are education, experience, ability, personal and language. Using the BERT model category of each section is determined.

c) NER (Named Entity Recognition)

The Resume Classification Named Entity Recognition (NER) model is specifically designed to analyze and extract relevant information from resumes. In this model, the NER model is used in Automatic Information Extraction, Resume Screening and Filtering and Candidate Matching. The data consist of the resume of people working in five different fields in the IT sector, which are labeled to cover eight different asset types. The NER model detects entities in the text and categorizes them. Here, entity name types consist of title, degree, major, skill, language, city, country, and date. Pre-trained RoBERTa model was adapted to the entity name recognition problem using our data.

*d) RoBERTa (*Robustly Optimized BERT Approach*)*

This proposed framework, pre-trained RoBERTa model is adapted to the entity name recognition problem. The RoBERTa [22] model is a language model that has the same structure as BERT but has more optimized training. It is a bidirectional converter-based language model similar to the BERT model and has been trained with various performance-enhancing optimizations [23]. In addition, RoBERTa implements the masked language modeling task more precisely and includes the task of predicting the order of the sentence.

e) Similarity

Cosine Similarity is a well-known measure employed in diverse NLP applications, such as the classification of resumes [24]. When classifying resumes, cosine similarity can be utilized to measure the similarity between a given resume and a set of predefined categories or job descriptions. To calculate cosine similarity, it is necessary to convert text information into a vector, so a sentence transformer is used [25]. Sentence transformers provide easy methods to compute embeddings for sentences, paragraphs, and images. And as a result, the percentage similarity between these two vectors is calculated. This embedding can be extracted on a word or sentence level. For sections skill and experience the similarity is calculated based on the whole section whereas for the rest of the sections similarity is calculated based on the named entities extracted by our NER model.

B. Dataset

a) Dataset for NER and Section Classification

The dataset consists of 286 resumes [26]. Each resume consists of 5 different sections: education, experience, ability, personal, and language. The personal part here represents a location. Because the personnel part in each resume is different and since there is a location in each resume, the location part is taken in the personnel part. There are around 1400 different sections. Each section was individually labeled and given as separate inputs to the trained models. The language of all resumes in the dataset is determined as English. While the resumes written in another language were removed in the data collection section, the words of a small number of other languages in the resumes are translated into English.

b) Dataset for Object Recognition and Classification

For the section object recognition, the dataset consisting of 1198 resumes were collected from the open source internet and labeled as text sets, and the model was trained on the resumes [27]. Each text group in the dataset is framed as an object. The size of the image is fixed at 640x640.

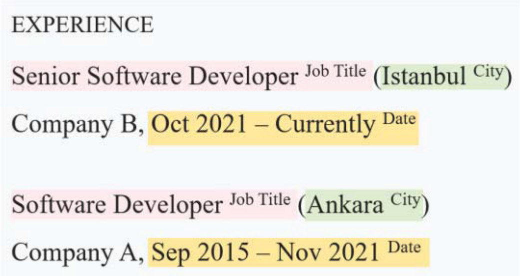
C. Preprocessing and Data Augmentation

For section classification, all words were converted to lowercase. The class names of the departments in resumes were removed from the data in order to prevent a bias such as memorization in deep learning models. The conjunctions and punctuation marks in the dataset were removed.

Data augmentation was applied to increase the performance of the models or to avoid the margin of error in the data obtained from the optical character recognition method. For the data augmentation part in section classification, methods such as deleting the characters in the words randomly, replacing the characters with other characters, and adding new random words to the data were used. Therefore, the number of data has tripled.

D. Labeling the data

In this study, the resumes of people working in five different fields in the IT sector were labeled with Label Studio [28], an open-source tool. These entity noun types consist of title, degree, major, skill, language, city, country, and date.



EXPERIENCE

Senior Software Developer Job Title (Istanbul City)

Company B, Oct 2021 – Currently Date

Software Developer Job Title (Ankara City)

Company A, Sep 2015 – Nov 2021 Date

Fig. 2. An example of labeled data [26].

Labeling resumes is a challenging task due to the large number of entity names. When a resume only includes a person's undergraduate education, there is only one diploma entry, whereas there may be numerous skills mentioned in the same resume. Insufficient occurrence of certain entity names can lead to oversight, while excessive occurrence makes the labeling process hard. To solve this problem, a semi-automatic labeling approach consisting of four steps is used.

- In the initial stage, a small portion of the dataset is automatically labeled using pre-trained entity recognition models.
- The labels are then manually reviewed and corrected by a human in the second stage.

- In the third step, an entity name model is trained using these data. With this new system, all of the data are labeled automatically, just like in the first stage.
- In the final stage, all data is reviewed and corrected by a human.

After completing these stages, the dataset is obtained in its final form.

III. Experiments

The software and AI modules that we used in our framework are developed using the Python programming language. In the optical character recognition part, we used an already existing model from EasyOCR [29]. In the text classification part, the Bert model was trained and fine-tuned.

Table 1 gives the micro, macro and weighted F1 score of the BERT model used in text classification. The Bert model has been fine-tuned. In addition, the error matrix has been extracted to better observe the result. The proportions of the clusters were divided into 70% for training, 15% for validation, and 15% for testing, respectively.

TABLE I. BERT MODEL PERFORMANCE [27].

		F1-Score		
Train Plan	**Model**	**F1 Mikro**	**F1 Makro**	**F1 WeightAverage**
Fine Tune	BERT	0.9391	0.9391	0.9398

In Table 2, the object recognition dataset is divided into two training and testing. The proportions of the clusters were divided as training 85% and testing 15%, respectively. The Yolov8 model is trained in 3 different versions up to the convergence limit. The mean average precision was used to compare the results.

TABLE II. RESULTS OF THE OBJECT RECOGNITION MODEL [27].

	Result	
Model	mAP ^{50}	mAP $^{50-95}$
YoloV8 Large	0.8456	0.6362
YoloV8 Medium	0.8370	0.6390
YoloV8 Small	0.7970	0.5800

Table 3 contains the entity name recognition performance table. Here, RoBERTa 's model was applied. As a result of this score, the F1 Weight Average value of 0.8963 was obtained.

TABLE III. NAME ENTITY RECOGNITION PERFORMANCE [26].

Model Name	**F1 Mikro**	**F1 Makro**	**F1 WeightAverage**
RoBERTa	**0.8958**	0.8928	**0.8963**

In the RoBERTa model, the lowest F1 score was obtained for the degree entity type with 72, and an F1 score above 86 was obtained for the remaining entity types as seen in Table 4. The three highest-performing entity types in the RoBERTa model are language, country, and city.

TABLE IV. ENTITY TYPE COMPARISON FOR ROBERTA MODEL [26].

Name Entity	**Precision**	**Recall**	**F1**
City	96.67	96.67	**96.67**
Date	83.52	95.3	89.02
Degree	64.29	81.82	72.0
Major	88.71	84.62	86.61
Title	89.29	85.03	87.11
Language	96.59	100	**98.27**
Country	94.74	96.43	**95.58**
Skill	87.25	90.78	88.98

$$Score = \sum_{i=0}^{5} w_i s_i \qquad (1)$$

The final score for similarity is calculated from the formula (1) above with inspiration from [4]. The W value represents the weight entered by the users and the S value represents the similarity of the resumes while i corresponds to the sections in the resumes. Users assigned a weight to each section, considering its importance in determining the overall score. The final score is sorted in descending order, showing that the resumes with the highest scores are positioned at the top of the talent pool.

We calculate each factor based on extracted features of resumes and then add their custom weights to evaluate the final score via the formula (1) :

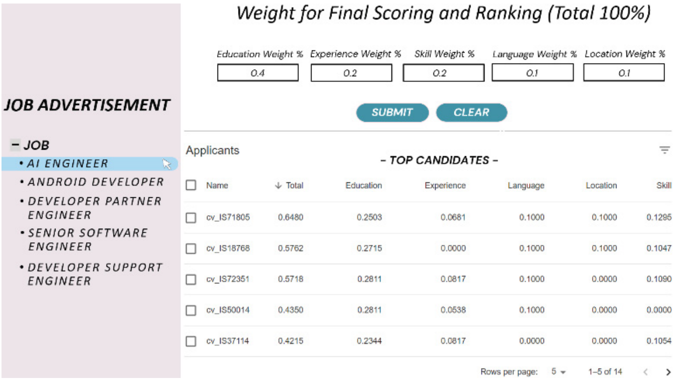
- Education Value: NER model for further filtering is applied and then the cosine similarity is calculated between the job description and the education section of the resume.
- Experience Value: The cosine similarity is calculated between the job description and the experience section of the resume.
- Language Value: First, NER model for further filtering is applied. If the resume matches the preferred language in the job descriptions, then the similarity value of the language is equal to 1, otherwise 0.
- Location Value: First, NER model for further filtering is applied. If the resume matches the preferred provinces in the job descriptions, then the similarity value of the location is equal to 1, otherwise 0.
- Skill Value: The cosine similarity is calculated between the job description and the skill section of the resume.

In the following, first, we present our main resume matching framework results, including cosine similarity evaluation with weights, final scoring, and ranking. Then, we present additional results of fresh graduate use cases for test purposes.

Main Results for General Scenarios The formula (1) is unfolded and converted to equation (2) below to be used for purposes in general use cases. Rate values for education, experience, skill, language, and location are calculated as mentioned previously.

$$0.4*EducationRate + 0.2*ExperienceRate + 0.2*SkillRate + 0.1*LangRate + 0.1*LocRate\ = Total\ Ranking \tag{2}$$

A web application for resume ranking is designed to streamline the hiring process and help employers efficiently identify the most qualified candidates. Here, the desired value can be entered for each section and the sum of these values should be 1. As seen in Figure 3, the weight values here can be adjusted. For example, the education part is 0.4, which shows that it is more important than other sections. With this web application, recruiters can focus on the most promising candidates, saving time and effort, leading to more efficient hiring decisions and ultimately bringing the best talent into the organization.



Weight for Final Scoring and Ranking (Total 100%)
JOB ADVERTISEMENT
– JOB
• AI ENGINEER
• ANDROID DEVELOPER
• DEVELOPER PARTNER ENGINEER
• SENIOR SOFTWARE ENGINEER
• DEVELOPER SUPPORT ENGINEER
SUBMIT
CLEAR
Applicants
– TOP CANDIDATES –
Rows per page: 5
1–5 of 14

Education Weight % 0.4 | Experience Weight % 0.2 | Skill Weight % 0.2 | Language Weight % 0.1 | Location Weight % 0.1

	Name	↓ Total	Education	Experience	Language	Location	Skill
□	cv_IS71805	0.6480	0.2503	0.0681	0.1000	0.1000	0.1295
□	cv_IS18768	0.5762	0.2715	0.0000	0.1000	0.1000	0.1047
□	cv_IS72351	0.5718	0.2811	0.0817	0.1000	0.0000	0.1090
□	cv_IS50014	0.4350	0.2811	0.0538	0.1000	0.0000	0.0000
□	cv_IS37114	0.4215	0.2344	0.0817	0.0000	0.0000	0.1054

Fig. 3. Web Application for Resume Ranking.

A Specific Scenario for Fresh Graduate Reconstructed formula of (1) can be seen below (3) to be used for the specific scenario in fresh graduate use cases. Particularly, we skipped values of experience and location attributes and chose higher weight values for the education of fresh graduates than general ones. Similarity, total ranking, and rank order results are presented in Table 5. Based on these promising results of our tool, we guess it will lead to reducing the gap between university education and the talent requirements in the industry allowing fresh graduates to start their career at proper-job.

$$0.7*EducationRate + 0.2*SkillRate + 0.1*LangRate\ = Total\ Ranking \tag{3}$$

TABLE V. COSINE SIMILARITY AND RESUME RANKING RESULTS ON THE USE OF HIRING FOR FRESH GRADUATE CANDIDATES.

Rec Num	Names	Education	Lang	Skill	Total Ranking	Rank Order
1	cv_ IS93499	0.559	0	0	0.391	12
2	cv_IS18768	0.679	1	0.523	0.68	2
3	cv_IS19741	0.43	0	0.55	0.411	10
4	cv_IS31219	0.613	0	0.409	0.511	7
5	cv_IS37114	0.586	0	0.527	0.516	5
6	cv_IS50014	0.703	1	0	0.592	4
7	cv_IS63954	0.483	0	0.481	0.434	9
8	cv_IS71805	0.626	1	0.648	0.668	3
9	cv_IS72351	0.703	1	0.545	0.701	1
10	cv_IS92858	0.608	0	0.388	0.503	8
11	cv_IS59077	0.562	0	0	0.393	11
12	cv_IS81612	0.736	0	0	0.515	6

IV. CONCLUSION

In this paper, a prototype of a resume-matching framework for enhancing the recruitment product was recommended. Firstly, the proposed framework detects resume sections (education, experience, skill, personal, language) with OCR and text classifications. Secondly, similarity measurement between resumes and job descriptions by employing NER and Transformer is evaluated. Finally, weights are added for ranking and final scoring. Then sorting is applied to the list of resumes in descending order for each job description as the expected output.

In this study, job application advertisements in the field of IT and the resumes of the people who applied were compared. In extensive experiments, text groups were labeled by considering the image in the resume and trained with the YOLOV8 model, which is object recognition. As a result of the model outputs, the texts were defined by giving the pre-trained model for OCT. After this step, text classification was done with the Bert model and the F1 Weight Average value was 0.9398. For Name Entity Recognition Performance, RoBERTa model was used and the F1 Weight Average value was 0.8963. In this section, cosine similarity was used to determine the similarity between vectors. Thus, the entity in the ad and the entity in the resume are calculated and the highest one is taken and ranked. This prototype of the resume matching system can be a valuable reference for the community in automatically processing resumes or developing a recruitment software product.

In the future, we aim to diversify the resume dataset for the presented method, enrich it to include different formats, structure the unprocessed resumes, combine company requests with potential candidates and make the website more functional. The framework would be integrated into the recruitment product and employed in various business sectors besides IT and used to choose skilled candidates, therefore reducing the workload on the decision-makers.

REFERENCES

[1] E. Kambur, and T. Yildirim. "From traditional to smart human resources management." International Journal of Manpower 44.3 (2023): 422-452.

[2] R. P. Kumar, S. S. Chowdhary, and R. Bhatia. "A machine learning approach for automation of resume recommendation system." Procedia Computer Science 167 (2020): 2318-2327.

[3] V. Bakır Oğuzalp, et al. "Banking Order Classification and Information Extraction." 2022 30th Signal Processing and Communications Applications Conference (SIU). IEEE, 2022.

[4] T. Quynh, and T.T. Dang. "Automatic Process Resume In Talent Pool By Applying Natural Language Processing." International Conference On Logistics And Industrial Engineering 2021. 2021.

[5] G. Shiqiang, F. Alamudun, and T. Hammond. "RésuMatcher: A personalized résumé-job matching system." Expert Systems with Applications 60 (2016): 169-182.

[6] Y., Kun, G. Guan, and M. Zhou. "Resume information extraction with cascaded hybrid model." Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05). 2005

[7] G. Bodhvi, et al. "Semi-supervised deep learning-based named entity recognition model to parse education section of resumes." Neural Computing and Applications 33 (2021): 5705-5718.

[8] Z. Shicheng, and X. Wang. "Resume information extraction with a novel text block segmentation algorithm." Int J Nat Lang Comput 8 (2019): 29-48.

[9] K. Satheesh et al. "Resume Ranking based on Job Description using SpaCy NER model." International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395–0056 7.05 (2020).

[10] Z. Yuying, G.Wang, and B.F. Karlsson. "CAN-NER: Convolutional attention network for Chinese named entity recognition." arXiv preprint arXiv:1904.02141 (2019).

[11] H. Xiaokai, et al. "Multi-Feature Fusion Transformer for Chinese Named Entity Recognition." 2022 41st Chinese Control Conference (CCC). IEEE, 2022.

[12] S. Nasser, C. Sreejith, and M. Irshad. "Convolutional neural network with word embedding based approach for resume classification." 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR). IEEE, 2018.

[13] S. T. Gopalakrishna and V. Vijayaraghavan. "Automated tool for Resume classification using Sementic analysis." International Journal of Artificial Intelligence and Applications (IJAIA) 10.1 (2019).

[14] N. Khamker,. "8." International Journal Of Innovative Science And Research Technology (2021).

[15] E. Faliagka, K. Ramantas, A. Tsakalidis, G. Tzimas "Application of machine learning algorithm to an online recruitment system". ICIW 2012: The seventh international conference on internet and web applications and services

[16] N. Khamker, Y. Khamker, and M. Butwall. "Resume match system." International Journal Of Innovative Science And Research Technology (2021).

[17] S. Hira, et al. "Resume parsing framework for e-recruitment." 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM). IEEE, 2022.

[18] M. Saswat, et al. "Resumate: A Prototype to Enhance Recruitment Process with NLP based Resume Parsing." 2023 4th International Conference on Intelligent Engineering and Management (ICIEM). IEEE, 2023.

[19] B. Jeonghun, et al. "What is wrong with scene text recognition model comparisons? dataset and model analysis." Proceedings of the IEEE/CVF international conference on computer vision. 2019.

[20] S. Baoguang, X. Bai, and C.Yao. "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition." IEEE transactions on pattern analysis and machine intelligence 39.11 (2016): 2298-2304.Chang, Yuan, et al. "Chinese named entity recognition method based on BERT." 2021 IEEE International Conference on Data Science and Computer Application (ICDSCA). IEEE, 2021.

[21] J. Devlin, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[22] L. Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019)

[23] C., Yuan, et al. "Chinese named entity recognition method based on BERT." 2021 IEEE International Conference on Data Science and Computer Application (ICDSCA). IEEE, 2021.

[24] W. Kadambari, et al. "A Cosine Similarity-Based Resume Screening System For Job Recruitment."

[25] R. Devika et al. "A deep learning model based on BERT and sentence transformer for semantic keyphrase extraction on big social data." IEEE Access 9 (2021): 165252-165261.

[26] E. Kesim, Ege, and A. Deliahmetoglu. "Named entity recognition in resumes." arXiv preprint arXiv:2306.13062 (2023).

[27] S. S. Helli, S. Tanberk, and S. N. Cavsak. "Resume Information Extraction via Post-OCR Text Processing." arXiv preprint arXiv:2306.13775 (2023).

[28] Label Studio, https://labelstud.io/, Accessed on May 2023.

[29] EasyOCR, https://github.com/JaidedAI/EasyOCR, Accessed on May 2023.