ResuMatch: Revolutionizing Recruitment with Advanced Resume Parsing and Job Profile Matching

No Author Given

No Institute Given

Abstract. During the recruitment process, many companies handle large numbers of resumes employing conventional techniques, such as manual processing or giving candidates standardised resume templates. To rank and select candidates, resume parsing can be used to extract pertinent data from resumes. However, current hiring practices demand more effective techniques and processes for resume analysis. While simple methods exist for parsing organized texts, they are not effective when dealing with unstructured data, such as resumes. This paper offers an all-encompassing solution for resume parsing, job profile matching, and credibility assessment. The system labels the resume format and applies specialized heuristics to extract information efficiently. This information is then used to match the resume content with the job posting in two ways: using cosine similarity and matching based on specific skills and experience. This approach effectively handles various resume formats and designs with minimal loss of information and context.

Keywords: Resume Parsing · Job Profile Matching · Natural Language Processing · Classification · KMeans

1 Introduction

In the current job market, numerous applications are submitted for various positions, and manually sorting and analyzing them is both time-consuming and requires significant human resources. This gap between job seekers and employers has created a demand for effective job-resume matching techniques. To categorize applicants and determine their eligibility for a position, automated resume parsing is necessary. Furthermore, analyzing resumes to extract a candidate's skills and experiences, as well as assessing the credibility of a resume, requires more complex algorithmic solutions than basic keyword parsing.

With the progress of NLP and machine learning technology, current research has implemented heuristic-based or deep learning based techniques for resume parsing. Many studies have acknowledged that parsing an unstructured document is a challenging task and have utilized methods like text block segmentation Zu et al. [31] to address the issue. However, this does not fully account for the variability in the formatting of different resumes, which adds an additional layer of complexity or lack of structure to the data. In most recent research, the reliability of a resume has not been considered, which is a component that human experts normally take into account to distinguish between legitimate and potentially false statements.

In light of these challenges, we have devised a three-step solution to the job-resume matching problem:

Resume Format Clustering: Basic techniques for parsing structured documents are not effective when applied to unstructured documents such as resumes. To address this issue, we cluster resumes based on their format. This step is critical in optimizing the efficiency of the subsequent information extraction step.

Information Extraction: After generating classes of semi-structured resumes, we use NLP and regex to develop class-specific heuristics for extracting information from resumes in each class. We then create json files for the job matching task using this extracted information.

Resume Profile Matching: Once we have extracted the json files corresponding to each resume, we use BERT [5] to determine similarity between a particular resume & the job posting details. If the similarity measure surpasses a predetermined threshold, we consider the applicant with that resume to be qualified for the posting.

The written work is structured as follows: Section 2 addresses the related works on resume parsing, job profile matching techniques using DL models, Section 3 describes the proposed models for different tasks like resume format clustering 3.2, information extraction 3.3, job description 3.4, and profile matching 3.5 in detail, Section 4 presents the experimental findings from the suggested model, followed by a discussion of the findings, And lastly, the paper's feature aims are summarised in Section 5.

2 Literature Review

We quickly explore the current tools for employment resume matching and discuss their drawbacks in this section. Job resume matching has typically been handled by human professionals, but recent advancement in ML have led to the development of an increasing number of strategies for automating this process. The majority of these solutions heavily rely on heuristics-based information extraction from resumes.

2.1 Information Extraction

Chen et al. [4] presents a two-step algorithm for extracting information from resumes. The first step uses a arrangement of rule-based and machine learning techniques to identify the basic components of a resume, such as personal information, education, and work experience. The second step employs a deep neural network model to extract more detailed information from the identified components. The authors evaluate performance of their algorithm on a dataset of 1,000 resumes.

Zu et al. [31], proposes a new approach to extract information from resumes by introducing a novel text block segmentation algorithm. The authors use a combination of rule-based and machine learning techniques to identify the text blocks in a resume, such as personal information, education, and work experience. They then use a set of features, including word embeddings and position-based features, to classify the text blocks and extract relevant information.

Parkavi et al. [17] use SpaCy, a Python package for natural language processing, for instance, extract private information from the unstructured or loosely organised data.

Suresh et al. [23] proposes a contextual model for information extraction in the field of resume analytics. To develop their approach, which involves pre-processing the resume data, extracting contextual features, and developing a rule-based information extraction system. The proposed model is evaluated on a dataset of 300 resumes.

Zhang et al. [29] proposes an end-to-end deep learning model for text reading and information extraction in document understanding. The authors introduce a hierarchical architecture that consists of a feature extraction module, a text reading module, and an information extraction module. The proposed model is evaluated on various public datasets.

2.2 Resume Profile Matching

One of the most common ideas behind finding whether a resume is fit for a given job posting is to encode the two as a shared representation and then finding the similarity between them using cosine similarity. In Guo et al. [7], they add weights to the different structural segments, and find the similarity between those separately. The weighted similarity score is further considered to find an appropriate matching.

In paper [22], Shao et al. divide the task of finding similarity further by finding the similarity score between attributes (key-value pairs) extracted from the resumes and job description. Consequently, trying to capture the combination of interactions between these attributes, internally and externally, to represent the overall interaction between any two documents.

Apart from similarity matching, other works adopt a deep learning approach by using models like BERT to find suitable candidates for a job description. The self-attention mechanism captures the complex semantic relationships between sentences and is capable of resolving the word ambiguity issue. Bhatia et al. [3] proposes an method for sequence pair classification using BERT for finding the similarity score of the candidate's work experience and the job description and using the score as the degree of appropriateness.

Ayishathahira et al. [2] and Li et al. [12] deploy CRF and Bi-LSTM-CNN models for the sequence labeling section to improve the performance of extracting resume entities and text features.

A third set of job resume matching framework exists which adopts an ontology based approach to mine the skills that are not explicitly indicated in the resume by using the directly extracted information. In works like Mishra et al. [14] and Sajid et al. [19], a job skill ontology is built to tackle the lack of a standard representation of the same skill. An ontology based system also helps understand and capture the semantic relationships between different skills. The main limitation of building an ontology is that it can become too domain specific and lose transferability over other domains.

3 PROPOSED METHOD

3.1 Dataset

A .csv file contains resume links of various types, such as links to Google Drive, Google Docs, LinkedIn profiles, and some unrelated links. At first, only the drive's links remained

after some filtering was done on the resume links. We then used a Python script to download the .pdf resumes, which resulted in the download of 3000 resumes from the drive links contained in the .csv file. After that, each PDF resume was converted into an image using the Python **pdf2image** package, which was then saved in a folder. A total of 3421 photos were obtained after converting the pdf resumes into images. These PDF resumes have been transformed into photos solely for the purpose of the format classifier.

We have constructed a new dataset containing the extracted resume material, together with each job posting and their similarity score, for the purpose of matching job profiles. Each job-resume match is given a binary label based on similarity scores that are larger than or equal to a threshold, which we have set at 0.8. The dataset has 43558 rows, and there are 21124 job-resume pairs that match and 22434 that don't.

3.2 Resume Clustering

This section's goal is to group resumes based on their structure or format to facilitate a faster extraction procedure. Prior approaches often entail converting the resume's .pdf format into .txt and using various methods or heuristics to extract information [27]. This strategy works effectively when the format or style of the resume is recognised, but it could fall flat when an entirely different resume is met. The distribution of each type of resume can be seen from the Fig. 1.

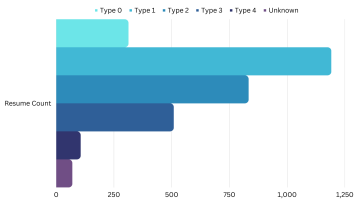

Fig. 1. Resume Formats & its Count

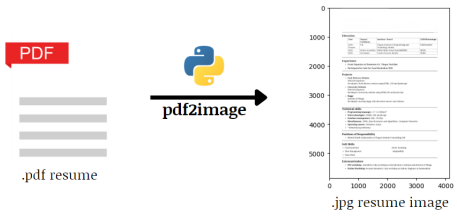

Fig. 2. Resume PDF to Image

We used the Python module **pdf2image**[1] to effectively convert PDF resumes to images with the least amount of information loss 2. The pdf2image library requires an adapter called **poppler**, via which the pdfs are transformed, in order to convert them to images. Since most resumes are A4 size and have dimensions of (4250*5500), a deep learning model would have a fairly large dataset to work with. As square photos are more effective for use with machine learning or deep learning models, we have therefore limited the size of each resume image to a set dimension of (600*600). Our suggested strategy is first turning the .pdf resume into graphics in order to allay this worry. The resume image is then given a label based on the structure that shares the most similarities with it. The information from the resume content, which is kept in a.json file, is extracted

[1] `https://pypi.org/project/pdf2image/`

using specially built algorithms that are particular to that class of resumes after the resume type label has been identified.

Unsupervised clustering is used to group resume images based on similarity after the images have been obtained. With the input image dimension of 600*600*3, EfficientNetB7 [25] was first utilized to extract image characteristics. The extracted picture features have an output dimension of $\mathbb{R}^{1x2560}$. Now, using principle component analysis, the size of the feature vectors was reduced because each feature vector has a relatively big dimension.

We used the scree plot Fig. 3, which is a plot between the main component and explained variance, to obtain the best value of n_components for the PCA model cited in [15]. The final reduced dimension of the feature vectors was created using the best value found for these data, which was 125. After that, each feature vector's labels were assigned using KMeans [16] based on Euclidean distance. The cluster of vectors with the smallest Euclidean distance. Using elbow curve approach to determine the right number of clusters to discover the ideal number. The resume dataset's n_clusters is discovered to be 5, as can be seen in the Fig. 3 as well.

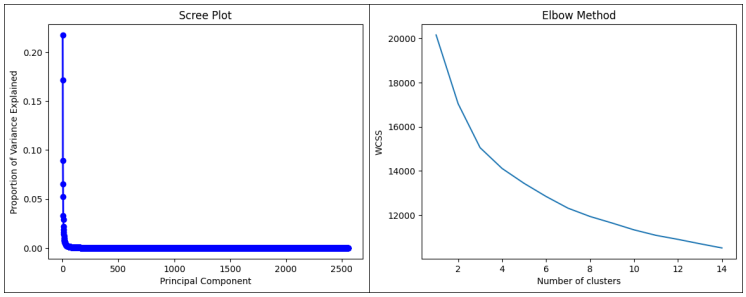

Fig. 3. Scree Plot & Elbow Curve

The classification report of the model is shown in Fig. 1 and includes the f1-score, precision, & recall for each class.

The resumes in the unknown class do not fall into any format clusters. They give such resumes the unknown label in order to handle new resume formats. By using a larger dataset to train the KMeans model, the accuracy can be improved.

Table 1. Experiment Results of Resume Clustering

Class	F1-Score	Precision	Recall
Type-0	0.53	0.54	0.51
Type-1	0.92	0.94	0.90
Type-2	0.86	0.93	0.80
Type-3	0.96	1.00	0.92
Type-4	0.51	1.00	0.34

3.3 Information Extraction

After obtaining the resume format class, the heuristics created specifically for that type of resume will be used. Regex and Python's nlp are used to generate these heuristics. It has been compared how well deep learning-based information extraction performs in comparison to heuristic-based information extraction [9]. However, it is discovered that heuristic-based extraction greatly outperforms deep learning-based extraction in terms of accuracy.

The **PyMuPDF** library is used to extract text from resumes that are in PDF format. Heuristics are used to determine typical font sizes and styles for each type of resume in order to identify headings. These heuristics are then used to build headers and text blocks, and headings that are uncommon in resumes or might be larger in a person's resume are discarded. All of a resume's headings can be recognised using this approach. Following text extraction and heading recognition, pertinent data is extracted using a variety of Python modules, including spaCy [21], regular expressions, and NLTK.

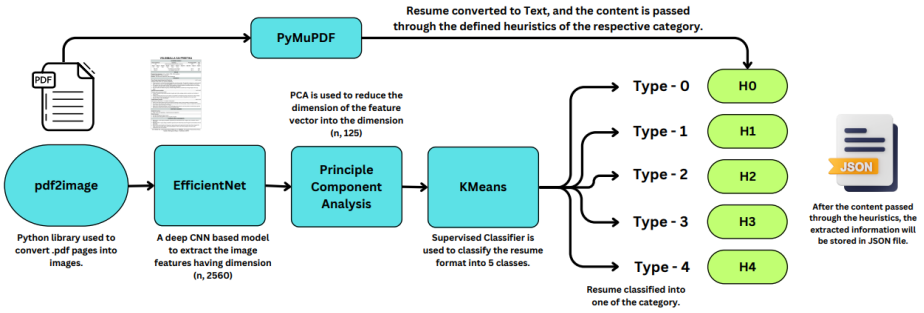

Fig. 4. Resume Format Clustering and Information Extraction Pipeline

JSON files are used to store the extracted data so that it can be processed later. The name, email address, mobile number, educational background, and other headings, such experience, abilities, and projects, are all included in the resume information extractor's final output.

Additionally, the extraction accuracy for each heading was assessed to see which headings our heuristics could accurately extract. The .json file contains the information that was extracted. Additionally, Fig. 5 displayed the extraction accuracy of one sample resume to test how well heuristics performed. When paying serious attention, it becomes clear that the extraction proportion for category's credentials, languages, and interests is lower. This is due to the fact that these sections rarely appear on most resumes, which explains why their percentage is so low.

3.4 Job Description

In this work, we want to create a classifier that can take a job description and a resume and tell us whether the resume fits the job description.

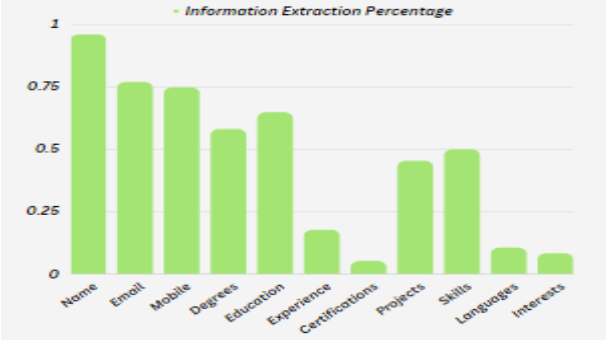

Fig. 5. Percentage of Extraction for each Heading

The previous task's extracted JSON data for each resume was then transformed into plain text using the *key: value* format. A total of fifteen job post descriptions for technical profiles including ***Cloud Engineer, Data Analyst, DevOps Engineer, Software Engineer***, etc. have been taken from ChatGPT [13]. There are 3421 resumes with simple paragraphs and 15 job descriptions to choose from. Using the transformer-based pretrained language model BERT, the current assignment entails extracting features for each resume and job posting and calculating the cosine similarity [10] between each resume and each job posting.

$$Cosine\ Similarity\ (A, B) = \frac{A.B}{||A||.||B||} \tag{1}$$

To help with completing the primary job, the next subtask entails developing and comparing the performance of DistilBERT 3.4 & BERT 3.4 binary classifiers [11]. Prior to this, each row's resume content and job description was processed using **<start>** and **<end>** tokens to denote the start & end of very sentence. Following preprocessing, the whole dataset was separated into three subsets using an 80:10:10 split ratio to create the training, testing, and validation datasets. The similarity score was used as the starting point to give labels to each pair of resume and job posting information. It was assumed that the associated resume is appropriate for the specified job position if the similarity score is more than or equal to 0.8. We conducted experiments using various thresholds and manually evaluated the results for each threshold. It was observed that a threshold of 0.8 is remarkably accurate in providing the correct answer.

DistilBERT : For training DistilBERT model, we have used DistilBERTFast tokenizer with maximum seq_length of 128. The classifier is trained using the hyper-parameters having the epochs 5, batch_size 16. Cross Entropy Loss serves the loss function, & Adaptive Moment Estimation Adam [8] is the optimizer employed in this.

DistilBERT is a compressed and lightweight variant of the popular BERT model. It retains 97% of BERT's language [20] understanding capabilities while significantly reducing its size and computational requirements. DistilBERT achieves this by distilling

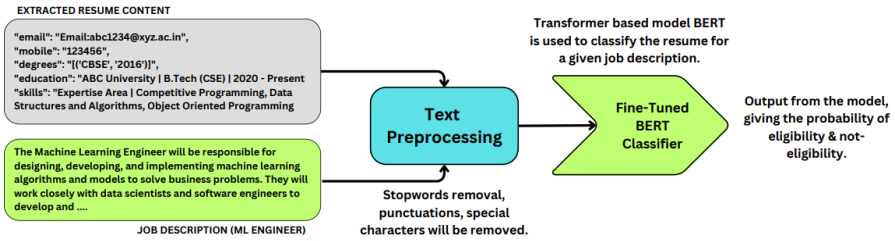

Fig. 6. Job Classifier Architecture

knowledge from a larger pre-trained BERT model into a smaller, distilled version. Despite its smaller size, DistilBERT maintains competitive performance on various natural language processing tasks, making it an efficient choice for resource-constrained environments or applications that require fast inference times without sacrificing accuracy.

BERT : Bidirectional Encoder Representations from Transformers For training, BERTTokenizer is utilized with maximum seq_length of 512. The Classifier is trained using the identical hyper-parameters configuration as DistilBERT.

$$\sigma(\overrightarrow{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2}$$

BERT is a SOTA pre-trained large language model [26] that utilizes a transformer-based architecture to generate high-quality contextualized representations of words. The model is trained on large amounts of textual data using masked language modeling & next sentence prediction objectives. During training, the model learns to predict masked words within a sentence and to determine whether two sentences are logically connected. BERT has achieved impressive results on a variety of natural language processing (NLP) tasks, including as sentiment analysis, text classification, and question answering, surpassing previous state-of-the-art models. BERTś success is attributed to its ability to capture the complex relationships between words in a sentence, as well as its ability to leverage large amounts of data for training, allowing to capture a rich and nuanced understanding of language.

For getting the classification percentage from the BERT / DistilBERT, we have used softmax 2 as the classification head which returns the probability of each class. The class having highest probability will be the final predicted class. After training both the models for 5 epochs, BERT outperforms the DistilBERT with 10.9% increase in accuracy. The results of all metrics can be seen from the table 2. The accuracy of the BERT on test set is around 88.17% and for DistilBERT it is 77.27% approximately. The training vs validation accuracy plot of best model can be seen from the fig. 7.

3.5 Profile Matching

In order to find the best candidates, profile matching [18], which involves lining up resumes and job descriptions, is an important recruitment process. In this part, we

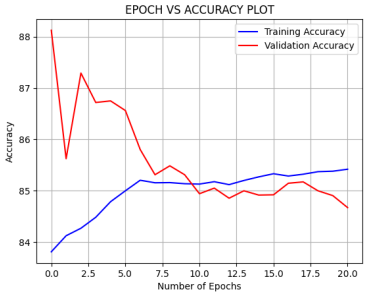

Fig. 7. Accuracy Vs Epoch Plot

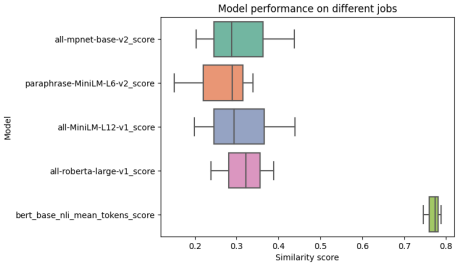

Fig. 8. Model Performance

offer a profile matching method that first establishes matches between resumes and job descriptions using semantic similarity scores, then refines those matches using degree requirement analyses.

Table 2. Job Profile Matching Experiment Results

Model	F1-Score	Precision	Recall	Accuracy
DistilBERT	0.77	0.78	0.80	77.27
BERT	**0.88**	**0.88**	**0.89**	**88.17**

Natural language processing (NLP) methods have been developed to automatically extract necessary job skills and educational qualifications from a collection of job descriptions [6]. The identification and extraction of distinctive talents indicated in job descriptions is made possible by custom entity recognition patterns [24]. The verification of a degree as the bare minimal degree need goes hand in hand with skill extraction. This approach simplifies the analysis of job descriptions and provides insightful data to both recruiting managers and researchers.

Table 3. Job Descriptions Dataset

Filename	Skills	Description	Min. Degree Level
Full-stack Developer.txt	python program...	A developer is ...	Bachelor's
Data Analyst.txt	business, r, python ...	A data analyst is ...	Bachelor's
Cloud Engineer.txt	azure, code ...	A cloud engineer is ...	Bachelor's
DevOps Engineer.txt	software, testing...	A DevOps engineer is ...	Bachelor's
Software Engineer.txt	software, design...	A software is ...	Bachelor's

A sample of the job description dataset, which was created by extracting details from job descriptions, is shown in 3 below. The reference to complete dataset taken from

here[2]. To encode resumes and job descriptions, we utilize five pre-trained models from the Sentence Transformers [1] library bert-base-nli-mean-tokens, all-mpnet-base-v2, paraphrase-MiniLM-L6-v2, all-MiniLM-L12-v1, and all-roberta-large-v1 [30]. Using these models, semantic similarity scores are computed between the job description and each resume, indicating how closely the abilities indicated in the resume correspond to those needed for the position. As a measure of overall similarity, the average score across all models is used.

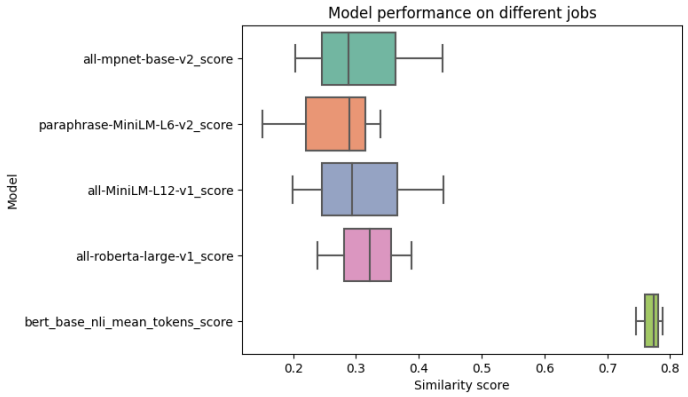

Fig. 9. Model Performance

We can see from the aforementioned graphic that bert-base-nli-mean-tokens outperforms the other models. We then established cutoffs for each similarity score based on the top 80% of scores. Potential matches are resumes that have an overall similarity score that is higher than the cutoff for a particular position. We contrast the degree level indicated in the CV with the minimal degree level required for the position in order to further refine the matches. Positive matches have a degree level that matches the criterion. Utilising a dataset of job descriptions and resumes, we apply our methodology to produce a matches dataframe that includes the degree requirement satisfaction, job ID, and resume ID. Finally, we mark matches with a 1 if they meet both the degree requirement and the threshold for similarity, and a 0 if neither criterion is met.

4 RESULTS & DISCUSSION

This section presents and analyzes the results of the proposed methods, and approaches for parsing resumes, job profile matching, and then providing the trustworthiness score to the resume. The collected resume dataset of 3000 resumes pdfs.

The pdf resumes are first converted to images for clustering based on the type of similar resumes. For that, Kmeans unsupervised algorithm is used. For this task, the

[2] `https://www.kaggle.com/datasets/xxxxx/job-descriptions`

accuracy obtained on test set is 79.0%. The number of images in each cluster is highly imbalanced, with label 4 having only 106 images while label 1 has 1191 images. This can make it difficult for the model to accurately predict the label of the images in the minority class (label 4). To address the class imbalance issue, techniques such as oversampling the minority class or undersampling the majority class can be used. This will help balance the number of images in each cluster and improve the model's performance on the minority class. The classwise f1-score, precision, and recall can be seen from the table 1. From this we can see that the average F1-Score obtained for the clustering process is 79.0% approximately.

For the job profile matching, we have used a comparison based analysis on Distil-BERT & BERT for creating contextualized embeddings [28] of the resume, and the job postings. The similarity score of each resume is taken with every job posting. The obtained classification accuracy of the best model is 88.17% on the 10% test set. The experimental results on the test set of both the models can be seen from the table 2. However, the threshold of 0.8 is arbitrary and may not be optimal for all resume-job posting pairs. It is possible that some similar text pairs have a cosine similarity value lower than 0.8, while some dissimilar text pairs have a cosine similarity value higher than 0.8. To improve the performance of the model. One can try adjusting the threshold value and see if it improves the accuracy.

5 CONCLUSION

In this paper, we present a pipeline that leverages specialized heuristics to extract information from PDF resumes in JSON format and then provides a label indicating if a given resume is eligible for a particular job description. To accomplish this, we employed BERT embeddings to find the cosine similarity between the resume content and job description, and if the similarity exceeds a predefined threshold, the resume is deemed eligible for the job. This method enabled us to create a ground truth dataset for our profile matching model using 15 technical job descriptions.

Additionally, we extracted required skills and minimum degree requirements to better match candidate profiles for profile matching, resulting in a significant improvement in results. Finally, we proposed a scoring method based on specific guidelines for evaluating the credibility of resumes, which enables organizations to save time and resources by focusing on trustworthy candidate profiles. Overall, our approach provides an efficient and effective solution for resume screening and matching.

References

1. Ahmed, A., Joorabchi, A., Hayes, M.J.: On the application of sentence transformers to automatic short answer grading in blended assessment. In: 2022 33rd Irish Signals and Systems Conference (ISSC). vol. 1, pp. 1–6. 2022 33rd Irish Signals and Systems Conference (ISSC) (2022). https://doi.org/10.1109/ISSC55427.2022.9826194
2. Ayishathahira, C.H., Sreejith, C., Raseek, C.: Combination of neural networks and conditional random fields for efficient resume parsing. In: 2018 International CET Conference on Control, Communication, and Computing (IC4). pp. 388–393 (2018). https://doi.org/10.1109/CETIC4.2018.8530883

3. Bhatia, V., Rawat, P., Kumar, A., Shah, R.R.: End-to-end resume parsing and finding candidates for a job description using BERT. CoRR **abs/1910.03089** (2019), http://arxiv.org/abs/1910.03089
4. Chen, J., Zhang, C., Niu, Z., et al.: A two-step resume information extraction algorithm. Mathematical Problems in Engineering **2018** (2018)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv **abs/1810.04805** (2019)
6. Gugnani, A., Misra, H.: Implicit skills extraction using document embedding and its use in job recommendation. Proceedings of the AAAI Conference on Artificial Intelligence **34**, 13286–13293 (04 2020). https://doi.org/10.1609/aaai.v34i08.7038
7. Guo, S., Alamudun, F., Hammond, T.: Résumatcher: A personalized résumé-job matching system. Expert Systems with Applications **60**, 169–182 (2016). https://doi.org/https://doi.org/10.1016/j.eswa.2016.04.013
8. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. International Conference on Learning Representations (12 2014)
9. Kowsher, M., Islam Sanjid, M.Z., Das, A., Ahmed, M., Hossain Sarker, M.M.: Machine learning and deep learning based information extraction from bangla names. Procedia Computer Science **178**, 224–233 (2020). https://doi.org/https://doi.org/10.1016/j.procs.2020.11.024, https://www.sciencedirect.com/science/article/pii/S187705092032398X, 9th International Young Scientists Conference in Computational Science, YSC2020, 05-12 September 2020
10. Lahitani, A.R., Permanasari, A.E., Setiawan, N.A.: Cosine similarity to determine similarity measure: Study case in online essay assessment. In: 2016 4th International Conference on Cyber and IT Service Management. pp. 1–6 (2016). https://doi.org/10.1109/CITSM.2016.7577578
11. Li, G., Kong, B., Li, J., Fan, H., Zhang, J., An, Y., Yang, Z., Danz, S., Fan, J.: A bert-based text sentiment classification algorithm through web data. In: 2022 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI). pp. 477–481 (2022). https://doi.org/10.1109/ICCEAI55464.2022.00105
12. Li, X., Shu, H., Zhai, Y., Lin, Z.: A method for resume information extraction using bert-bilstm-crf. In: 2021 IEEE 21st International Conference on Communication Technology (ICCT). pp. 1437–1442 (2021). https://doi.org/10.1109/ICCT52962.2021.9657937
13. Lund, B., Wang, T.: Chatting about chatgpt: How may ai and gpt impact academia and libraries? Library Hi Tech News **40** (01 2023). https://doi.org/10.1108/LHTN-01-2023-0009
14. Mishra, R., Rodriguez, R., Portillo, V.: An AI based talent acquisition and benchmarking for job. CoRR **abs/2009.09088** (2020), https://arxiv.org/abs/2009.09088
15. Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., Saikhom, R., Panda, S., Laishram, M.: Principal component analysis. International Journal of Livestock Research p. 1 (01 2017). https://doi.org/10.5455/ijlr.20170415115235
16. Na, S., Xumin, L., Yong, G.: Research on k-means clustering algorithm: An improved k-means clustering algorithm. In: 2010 Third International Symposium on Intelligent Information Technology and Security Informatics. pp. 63–67 (2010). https://doi.org/10.1109/IITSI.2010.74
17. Parkavi, Pandey, P., J, P., S, V.G., W, K.B.: E-recruitment system through resume parsing, psychometric test and social media analysis. International Journal Of Advanced Research in Basic Engineering Sciences and Technology **5**, 364–368 (2019)
18. Rodriguez, L.G., Chavez, E.P.: Feature selection for job matching application using profile matching model. In: 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS). pp. 263–266 (2019). https://doi.org/10.1109/CCOMS.2019.8821682
19. Sajid, H., Kanwal, J., Bhatti, S.U.R., Qureshi, S.A., Basharat, A., Hussain, S., Khan, K.U.: Resume parsing framework for e-recruitment. In: 2022 16th International Confer-

ence on Ubiquitous Information Management and Communication (IMCOM). pp. 1–8 (2022). https://doi.org/10.1109/IMCOM53663.2022.9721762
20. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter (10 2019)
21. Schmitt, X., Kubler, S., Robert, J., Papadakis, M., LeTraon, Y.: A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate. In: 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS). pp. 338–343 (2019). https://doi.org/10.1109/SNAMS.2019.8931850
22. Shao, T., Song, C., Zheng, J., Cai, F., Chen, H., et al.: Exploring internal and external interactions for semi-structured multivariate attributes in job-resume matching. International Journal of Intelligent Systems **2023** (2023)
23. Suresh, Y., Manusha Reddy, A.: A contextual model for information extraction in resume analytics using nlp's spacy. In: Inventive Computation and Information Technologies: Proceedings of ICICIT 2020, pp. 395–404. Springer (2021)
24. Takalikar, M., M.Kshirsagar, M., Singh, K.: Pattern based named entity recognition using context features. International Journal of Computer Sciences and Engineering **6**, 365–368 (04 2018). https://doi.org/10.26438/ijcse/v6i4.365368
25. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks (05 2019)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
27. Vukadin, D., Kurdija, A.S., Delač, G., Šilić, M.: Information extraction from free-form cv documents in multiple languages. IEEE Access **9**, 84559–84575 (2021). https://doi.org/10.1109/ACCESS.2021.3087913
28. Xu, H., Van Durme, B., Murray, K.: BERT, mBERT, or BiBERT? a study on contextualized embeddings for neural machine translation. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 6663–6675. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021), https://aclanthology.org/2021.emnlp-main.534
29. Zhang, P., Xu, Y., Cheng, Z., Pu, S., Lu, J., Qiao, L., Niu, Y., Wu, F.: Trie: end-to-end text reading and information extraction for document understanding pp. 1413–1422 (2020)
30. Zhuang, L., Wayne, L., Ya, S., Jun, Z.: A robustly optimized BERT pre-training approach with post-training. In: Proceedings of the 20th Chinese National Conference on Computational Linguistics. pp. 1218–1227. Chinese Information Processing Society of China, Huhhot, China (Aug 2021), https://aclanthology.org/2021.ccl-1.108
31. Zu, S., Wang, X., Darren, S.: Resume information extraction with a novel text block segmentation algorithm. Linguistics **8**, 29–48 (10 2019). https://doi.org/10.5121/ijnlc.2019.8503