Automated recruitment,resume parsing,talent acquisition tool

Alessandro Gentili
University of Bologna
alessandro.gentili@pw.edu.pl

Nigel Tao
National University of Singapore
naigel.tao@pw.edu.pl

Manon Lainaud
University Clermont Auvergne
manon.lainaud@pw.edu.pl

Marc Martinez
Polytechnic University of Catalonia
marc.martinez@pw.edu.pl

Supervisor: Anna Wróblewska,
Warsaw Universityof Technology,
anna.wroblewska1@pw.edu.pl

Abstract

This project presents a solution for optimizing the matching between resumes (CVs) and job descriptions (JDs) using advanced natural language processing (NLP) and machine learning techniques. The proposed methodology involves collecting and preprocessing CV and JD datasets, including steps like stopword removal, Named Entity Recognition (NER), and Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction. By leveraging contrastive learning and embedding generation, the system computes precise matching scores between CVs and JDs, enhancing the recruitment process. The approach ensures a meaningful alignment of candidate profiles with job requirements, leading to more effective job recommendations. This work offers a structured approach to improving the accuracy of automated job matching, providing a foundation for future advancements in this domain.

Credits

We would like to acknowledge the following individuals and resources for their contributions to this project:

- **Anna Wroblewska**, for her valuable guidance and support throughout the project.
- **The authors of the mentioned papers**, whose methodologies and insights were integral to shaping our approach.

We also wish to thank our peers and tutors for their constructive feedback and contributions to the project.

1 Introduction

Natural Language Processing (NLP) is a field of artificial intelligence (AI) that enables computers to understand, analyze, and generate human language. It combines techniques from computer science, linguistics, and statistics to process large amounts of textual or spoken data. NLP applications are vast and include chatbots, sentiment analysis, image captioning, and speech recognition. The main goal of NLP is to make human-machine interactions more natural and intuitive.In addition, NLP can be extremely useful in helping with day-to-day tasks such as recruitment.
The objective of the project is to create a tool that can extract relevant information from CV's which can be used by HR departments to classify candidates for initial recruitment phases, which are usually online. In addition, it can be used for candidates to have job recommendations. NLP techniques will be used to identify relevant CV information such as years of experience, skills and location.

In this document, we will begin by conducting a comprehensive literature review of state-of-the-art (SOTA) methods and tools in NLP, recommendation systems, and CV-JD matching. This review will explore existing research and methodologies, with a particular focus on comparison tables for datasets, methods, and pre-trained models. Following this , we will develop and document the solution concept for the extraction of features from CVs and JDs, focusing on the approach and methodology of the project. This will include a detailed explanation of the approach, covering methods for feature extraction, merging, and initial model selection.

2 Literature Review and Background

We introduce some key papers and projects relevant to this work.

2.1 Resume Parser Using Natural Language Processing Techniques

The paper aims to streamline the recruitment process by developing an automated system that parses, structures, and ranks resumes based on job requirements using Natural Language Processing (NLP) techniques. The proposed solution involves a job portal where job seekers upload their resumes, which are then converted into plain text using Optical Character Recognition (OCR).

Key NLP techniques, including lexical analysis, syntactic analysis, semantic analysis, and Named Entity Recognition (NER), are utilized to extract and structure relevant information such as education, experience, and skills. The system also incorporates social media data for comprehensive profiling and ranks candidates objectively using ElasticSearch-based dashboards. This approach enhances recruitment efficiency while reducing biases and effort for both recruiters and candidates.

2.2 A CV Parser Model Using Entity Extraction and Big Data Tools

This paper discusses the implementation of a resume parser using nlp,but taking into account the problem that CV´s are usually unstructured data and are not stored in conventional databases.The authors explore how to extract appropriately the entities of the resumes and how to optimize the process using big data tools.

This project also uses NER technique to identify the entities of the CVs and more advanced techniques such as entity relation extraction which connects relationships shared between different entities. It also reveals any conne-tions or events shared among these entities. The main aspect is it also establishes the indirect relationships amongst indirect connections.It also mentions common NLP techniques such as pos tagging or tokenization for the CV reusmer

The main goal is to extract the necessary entities from resumes ,which can be in a lot of different formats,such as XMl.In this step, the main tool used in NER.With this process,they can extract the main features for a lot of different types of inputs.However,they faced with the problem that words can have different meanings depending on the context.FOr this reason,more advanced techniques as linking or entity relation extraction.For optimizing the process of dealing with a huge amount of data they use Hadoop mapReduce to process the words in parallel.

2.3 CONFIT: Improving Resume-Job Matching Using Data Augmentation and Contrastive Learning

The paper presents CONFIT, a simple and general-purpose approach to model resume-job matching using contrastive learning and data augmentation.First,the authors formulate the problem of matching a job and a resume as a function $f(R,j) \rightarrow R$,where the function usually involves a neural network.

The datasets used on this project are AliYun dataset,which have been published by Alibaba on a job-resume intelligent matching competition.All the data is in Chinese and it provides the jobs and resumes parsed into a different columns,such as education,fields and languages in case of the CVs. The second dataset used is Intellipro dataset,which is provided by a hiring company.Sensitive fields have been removed for protecting privacy.The authors record ,for each resume-job pair if the candidate is accepted or rejected.For generalizing,they parse resumes and jobs into similar sections as AliYun dataset.

The first problem they encountered is Resume-job datasets often suffer from sparsity of interaction records, as job seekers only apply to a few jobs. CONFIT addresses this by using data augmentation techniques,which consists on creating more data using our existing data.

For applying this technique ,the authors describe the dataset as a bipartite graph,in which one group is jobs and the other CVs and the edges have labels which can be accepted or rejected. The application of data augmentation consists of taking parts of the CVs,such as experience and paraphrasing them,creating a new CV, which have the same edges as the original.Then,they do the same but with the jobs.

On second place,they use contrastive learning to create a high-quality embedding representation. Applied to this case ,the authors construct a dataset where each instance contains a pair of positive matched resume-job and a number of unsuitable resumes.Then,they train a encoder network to optimize the cross entropy loss. Finally, CONFIT

produces a matching score s between the ⟨R, J⟩ pair using inner product.

The authors based the encoder architecture on a more simplified one used on another paper,where they encode each filed(for a resume or a job) individually using a pre-trained encoder and then they use and attention layer to model the internal interactions between the fields and finally uses another attention layer to see the external interactions.The authors simplify it sung a lineal layer,which fuses the field representations into a single dense vector.

The system's ability to efficiently rank large numbers of resumes and jobs is highlighted, making it a robust solution for person-job fit systems. Ethical considerations regarding biases and privacy are discussed, emphasizing the importance of mitigating biases in such systems.

The performance is demonstrated on the experiments on two real-world person-job fit datasets ,which show that C ON F IT outperforms prior state-of-the-art methods by up to 19% and 31% absolute in nDCG@10 for ranking jobs and ranking resumes, respectively. It also remains competitive on classification tasks despite not directly optimizing for them. The authors find that the data augmentation and contrastive learning components are crucial to CONFIT's strong performance.

3 Solution concept and approach

Now that we have done a little review of the literature review and background, we are going to see the solution concept and approach.
The proposed solution aims to optimize the matching between resumes (CVs) and job descriptions (JDs) through a structured approach. The first step involves data collection. The necessary datasets include CVs, which contain candidate profiles with information such as skills, experience, education, and job history, and JDs, which outline the job requirements, responsibilities, and desired skills. Once the datasets are collected, the next phase involves data cleaning and preprocessing, which is critical for ensuring the accuracy and relevance of the analysis. This process begins with stopword removal, which eliminates common, non-informative words such as "am," "is," "are," "of," "a," and "the." These words do not contribute meaningful information to the underlying topics and are removed to enhance the dataset's semantic clarity. Next, non-English rows are discarded to ensure linguistic consistency across the dataset, which is essential for accurate natural language processing (NLP).
Following this, Named Entity Recognition (NER) is employed using SpaCy to extract meaningful entities from both CVs and job descriptions (JDs). Key focus areas for extraction include job titles, skills, certifications (e.g., programming languages, tools), as well as education and years of experience, which are crucial for aligning candidate qualifications with job requirements.
In addition, high-frequency words that appear in a significant percentage of documents (e.g., 80–90%) are removed, as these terms are often non-distinct and fail to provide valuable differentiation between various job roles or qualifications. These preprocessing steps align with methodologies proposed by Jagwani et al. (2023), who demonstrate the effectiveness of combining entity detection using SpaCy's NER and Latent Dirichlet Allocation (LDA) to extract meaningful semantic representations for resume evaluation. Their approach focuses on creating a content-driven scoring system, achieving an overall accuracy of 82% by considering attributes such as education, work experience, and skills (Jagwani et al., 2023).
Our proposed approach incorporates Term Frequency-Inverse Document Frequency (TF-IDF) as a key technique for feature extraction from resumes (CVs) and job descriptions (JDs). TF-IDF is widely used in information retrieval and natural language processing to identify and prioritize words or phrases that are not only frequent within a document but also distinctive across a corpus. By applying TF-IDF, the system ensures that terms specific to a resume or job description—such as technical skills, certifications, or unique job requirements—are given higher weight, while common words that appear across documents are deprioritized.
TF-IDF operates by calculating two key components:

- Term Frequency (TF): This measures how often a term appears in a document relative to the total number of terms in that document.

- Inverse Document Frequency (IDF): This scales the importance of a term inversely with its frequency across all documents in the corpus. Words that are common across many documents, such as "and" or "the," re-

ceive lower scores, while unique terms are weighted more heavily.

The mathematical formulation of TF-IDF can be expressed as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

where:

$$\text{TF}(t, d) = \frac{\text{occurrences of } t \text{ in document } d}{\text{Total terms in document } d}$$

$$\text{IDF}(t) = \log \frac{\text{Total number of documents}}{\text{Number of documents containing } t}$$

By leveraging TF-IDF, the proposed system extracts key features from CVs and JDs that are particularly relevant to matching candidates to job postings. This approach aligns with the methodologies discussed in Das et al. (2018), where the authors emphasize the importance of feature extraction from unstructured data for better alignment with recruitment processes. The TF-IDF model not only captures domain-specific terminology but also reduces the influence of generic terms, enabling a more precise and meaningful representation of resumes and descriptions. This feature extraction forms the foundation for subsequent processes in the pipeline, such as embedding generation and similarity computation, ensuring the system focuses on critical aspects of the resume-job matching task.

In order to maintain clarity and avoid confusion, CV and JD datasets will be processed separately. This independent preprocessing ensures that the unique attributes of each dataset are preserved, enabling the model to effectively differentiate between the qualifications presented in CVs and the requirements outlined in JDs. Specifically, CVs capture information such as experiences, skills, and educational qualifications, while JDs emphasize job-specific requirements, responsibilities, and desired expertise. By processing these datasets independently, the model can accurately represent the distinct characteristics of each, leading to more effective resume-job matching.

This methodology aligns with the approach described in ConFit: Improving Resume-Job Matching using Data Augmentation and Contrastive Learning by Yu et al. (2024). The authors emphasize the importance of representing CVs and JDs as dense embeddings to encapsulate their unique information and improve matching accuracy. Their work demonstrates that independent preprocessing and representation of these datasets enhance the system's ability to rank jobs and resumes effectively, achieving significant improvements in performance metrics such as nDCG@10 (Yu et al., 2024).

Contrastive learning, a machine learning technique, is particularly effective in representation learning. By training the model to distinguish between similar and dissimilar data pairs, it ensures that similar items are closely clustered in the embedding space while dissimilar ones are positioned further apart. This approach is also used in the methodology described in ConFit: Improving Resume-Job Matching using Data Augmentation and Contrastive Learning, where contrastive learning is leveraged to address data sparsity and improve matching accuracy. By embedding resumes and job descriptions into a shared vector space, the model enables efficient similarity measurement, achieving significant improvements in ranking performance over traditional approaches such as BM25 and neural embeddings. The use of inner product similarity to compute matching scores further demonstrates the effectiveness of this strategy, which serves as a basis for our implementation plan (Yu et al., 2024).

To improve the accuracy and relevance of resume-job matching, we will employ contrastive learning as a core technique for embedding generation and similarity measurement. This approach enables the system to learn dense embeddings for both CVs and JDs within a shared vector space, facilitating a more precise computation of matching scores. The inner product of these embeddings will then be used to calculate a matching score, ranking jobs for a CV by their matching percentage. This methodology ensures that resumes and job descriptions are not only compared on textual similarity but also on contextual alignment.

4 Reproducible Proof of Concept

We are now going to demonstrate the feasibility of the previous solution by carrying out a proof of concept (PoC). In this section, we'll present the main steps we took to implement the above solution.

We began by preprocessing the dataset containing the job descriptions. This dataset contains 31

columns, but in this project we'll be concentrating on the following 3 columns: title, description and skills_desc. Once we've extracted these columns, we'll group the description and skills_desc in the same column, so that we have everything expected of a candidate in one place. We'll save this result as a csv file called postings_combined_desc, which we can reuse later. Lines containing missing values are deleted. We then created 2 functions that extract all the skills and create an array containing all these skills. These functions will later be applied to our dataset, creating a new feature called skill.

We then cleaned up the dataset using nltk, removing URLs, special characters and stopwords, converting words to lowercase and lemmatizing them. These changes are saved in a new column, and the resulting dataframe is saved as a csv file entitled cleaned_JD. We then apply the 2 functions we created earlier to retrieve a list of skills for each job description, and create another csv file called cleaned_JD_with_skills.

We then performed the data processing for the dataset containing the CVs. In this dataset, only the resume_str and category columns will be kept. To get a better idea of our dataset, we counted the number of CVs we had in each category. As with the job descriptions, we clean up the dataset by removing special characters such as line breaks and quotation marks, replacing missing values with empty strings and deleting rows where all values are missing. We retrieve a csv of this dataset: Resume_Clean. Once again, we'll create functions to extract the skills and create an array containing all the skills.

We'll clean up the dataset once again using nltk. This time, we'll remove special characters and numbers, as well as stopswords.

We'll then retrieve a list of skills using the functions created earlier, and we'll also retrieve the role. We'll retrieve this in a csv entitled Resume_With_Skills_And_Roles.

Finally, we implemented the solution. To do this, we used the Resume_With_Skills and cleaned_JD_with_skills datasets created earlier. We begin by grouping the columns clean_resume and the skills retrieved previously in the resume dataset, and clean_JD and the list of skills in the job dataset.

We then download a tokenizer and a pre-trained model and create a get_embeddings_in_batches function that calculates the embeddings for a list of texts using a transform model like BERT.

Then we create rank_jobs, which allows us to associate a resume with job offers using text embeddings and cosine similarity measures.

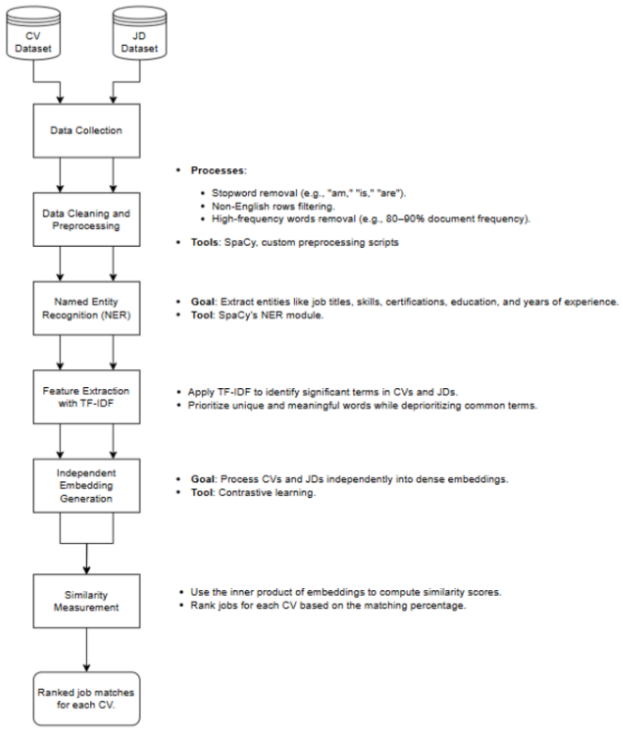

Figure 1: Pipeline Diagram for the Proposed Resume-Job Matching System: This flow illustrates the sequential steps involved, starting from data collection of CVs and JDs, through preprocessing, entity extraction, feature extraction using TF-IDF, embedding generation with contrastive learning, and similarity computation, leading to ranked job matches for each CV.

References

[Das, P., Pandey, M., & Rautaray, S. S.2018] Das, P., Pandey, M., & Rautaray, S. S. 2018. *A CV Parser Model using Entity Extraction Process and Big Data Tools. International Journal of Information Technology and Computer Science, 10(9), 21-31*, Link

[Jagwani, V., Meghani, S., Pai, K., & Dhage, S.2023] Jagwani, V., Meghani, S., Pai, K., & Dhage, S.. 2023. *Resume Evaluation through Latent Dirichlet Allocation and Natural Language Processing for Effective Candidate Selection. arXiv.* . Link

[Yu, X., Zhang, J., & Yu, Z.2024] Yu, X., Zhang, J., & Yu, Z.. 2024. *ConFit: Improving Resume-Job*

Matching using Data Augmentation and Contrastive Learning. arXiv . Link