Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 6, July, 2021: 10048 - 10057

Research Article

A Novel Approach to Rank Jobs for a Jobseeker

Dr. Shailender Kumar [a], Vaibhav Jain[b]

[a]Associate Professor, Department of Computer Science, Delhi Technological University (Formerly known as DCE), Shahbad Daulatpur, Main Bawana Road, Delhi, 110042, India
shailenderkumar@dce.ac.in
[b] Department of Computer Science, Delhi Technological University (Formerly known as DCE), Shahbad Daulatpur, Main Bawana Road, Delhi, 110042, India
vaibhavjain_2k18cse502@dtu.ac.in

Abstract

Recruitment is one of the major tasks for the HR industry. As it offers numerous job opportunities at various locations, it is therefore very time taking and difficult to determine which user-job mapping is the most relevant one for a given user, on the basis of their skills and interests. In order to serve consumers, we propose an information recommendation engine that utilises an algorithm to match a user's interests and abilities with available jobs. To form recommendations, the system uses different text filtration and similarity measures. Testing the model using data from a major job offering agencies shows that topic models are relevant and effective. Therefore, this technique can be applied to several other industries.

Keywords: recommender; e-Recruitment; Human resources; Information retrieval

1. Introduction

Talent acquisition has always been very complex, time taking and an important role of Human Resources (HR). Nowadays, with increasing Job Applications for a particular Job it has become very Challenging for the Human Resources(HR) to deal with it, lots of resources get wasted and a lot of time is spent on processing these job applications and finding a suitable candidate. The most problematic thing is that the CV of the candidates/ JobSeeker has no Standard format that makes the shortlisting of preferred candidates a lot more difficult. Also, to do so it requires a lot of domain knowledge but the roles in the industry are so diversified now that it has become near to impossible for a particular HR to have such a base of knowledge to shortlist candidates of all domains.

Lots of research has been carried out to solve this problem as already it has been said that HR does not do good to the candidates who have applied for the Job and have given their ample time for preparing an application. In contrast, there is an appreciable danger of losing out on relevant and qualified jobs or candidates because of the manual approach, and the process takes time. Also, there is a problem with not providing proper feedback on why the candidature was rejected for that Job. On the other hand, Organisations also are not very interested to provide feedback to the candidates as it is of no profit to them and extra effort is also required from their side to do so.

However, if there exists any mechanism that automatically screen the applicants candidature, it will be helpful for both recruiter and the applicant as recruiters can save their branding reputation and applicants would know the proper reason for the rejection and now they can focus on the areas they lag and prepare good for the next time on the same. Advances in technology and consumer expectations have created new and more tailored services for job seekers and recruiters, leading to an increased demand for predictive analytics that can pull the most appropriate information from various resources. Additionally, the same approach could be very useful for many educational organizations and employment agencies which could auto analyze the profiles of the unemployed people and suggest a suitable course or training required for them to improve their candidature which could give them an upper hand against the other people applying for the Job of the same domain.

Nonetheless, recommender systems have existed since the dawn of computing, but interest in the topic has grown in recent years as a result of increased exposure to social networks and e-commerce sites. To help people find suitable jobs, recruiters often utilise recommendation systems to recommend job information to various users[4]. This subject has received very little research because researchers are interested in employing content-based recommendation systems to help with recruitment agencies. Many algorithms/approaches already exist but the problem is they are mostly syntactic matching i.e. they will match the keywords of the job offer to that of the candidate's profile. The only problem with this kind of technique is that it fails to take into consideration the meaning that resides behind the text that intends to label the requirements. To improve this approach, there may be required some sort of expert knowledge and background into account.

In this paper, there is a solution to this problem, there is a mechanism that maps the Job with the JobSeeker profiles based on the skills of the candidate that he has mentioned in the CV and the skills mentioned in the Job Description and uses an approach that automatically matches the job offers to the suitable candidates profile. The information retrieval algorithms mentioned in this paper are used to augment the various recruitment strategies of job search, testing, and short listing of the right talent. It reduces the amount of time and effort that goes into the entire process, at the same time, providing interesting and precise information filtration. Latent features are applied to the text to increase the accuracy of outcomes.

The paper goes on to elaborate, and describes the rest of the process. The very next section is concerned with the background and related work in the field, it aims to brief what other researchers have done so far in this area. The followed section aims to define the actual problem followed by defining the methodology that has been used to carry out research, post that results are discussed and finally the paper in concluded with discussions

2. Background and Related Work

To extract the appropriate data from web databases has become increasingly time-consuming in today's recruitment sector, Martinez-Gil et al.[1]. There is a heavy reliance on precise data filtering systems within the market. Job postings and employment portals have witnessed a decrease in quality and performance as a result of this data surge. Now that everyone has access to more job information and job objectives , both job identifiers and recruitment teams must complete a massive pool of job postings and jobs advertised information in order to arrive at the result, Tinelli et al.[2]. Every year there is an increase in the number of people who apply for a job, with this for a particular job a lot of applicants are there and many of the profiles are deserving ones. It is a big challenge for the HR person to shortlist the most suitable candidate’s profile from the bag of applicants. The process of matching a Job with the desired Job Seeker profile is the same as a recommendation system as we are recommending candidates for a particular Job Posting.

The below figure, Fig.1 shows a general flow of how recommendation works. It’s typically a model that takes item representation (i.e products/News Articles/Jobs etc) as input along with the user feature vector, it does the profile item matching/ filtering and finally returns the recommended items. These returned items are generally ranked on the scale of 10, and the same are recommended accordingly.

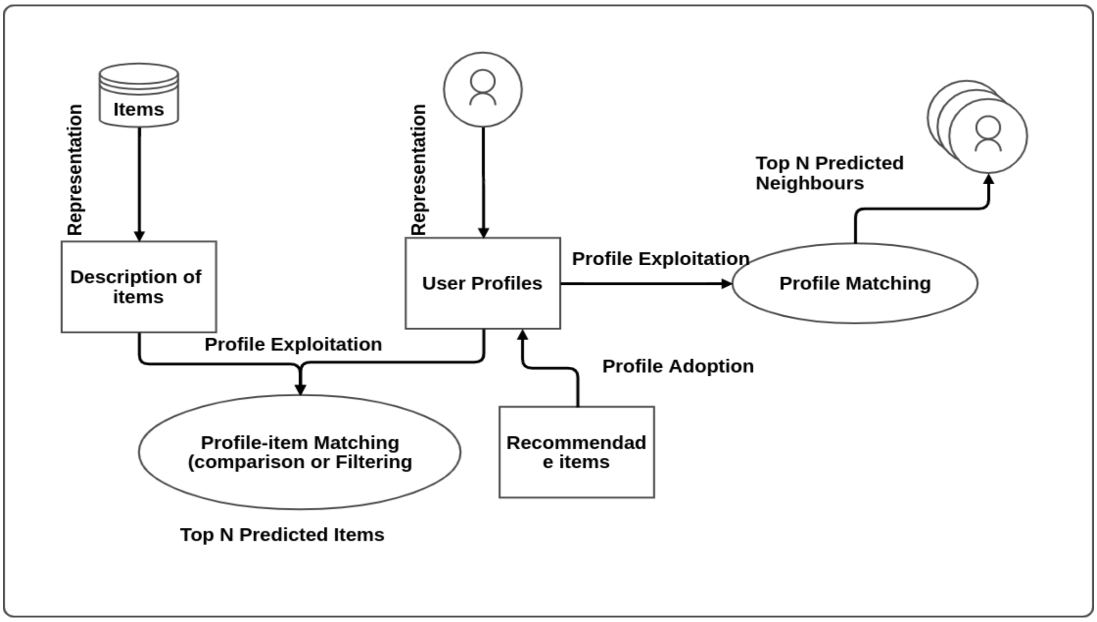

Fig:1 General Model of Recommendation Engines

Lu et al., [4] published a survey detailing the applications that are using the recommender systems and the approaches adopted by them to build the recommendations. Recommendation systems are broadly classified into four types i.e. Collaborative filtering, Content-based filtering, Knowledge-based and Hybrid as per various researchers. Al-Otaibi et al., [8] published a survey detailing recommendations systems of Job. They detailed out all the steps involved in the recruitment process like the factors that may lead to the selection of candidates to how e-recruitment is helping the companies to ease down the recruitment process.

Verma et al. [25] proposed a recommendation system that recommends personalized career paths using similarities across education and experience. Skill extraction has also become a very popular topic and has grabbed much attention. Kivimaki et al. [26] proposed a mechanism that extracts skills from the documents focusing on hiring candidates for an organization. The initial work computes the similarities between a document that is provided and the Wikipedia pages text and then uses an algorithm to associate the input document with the skills Wang et al.[27] propose a system to extract skills from user profiles in a social network context.

Most of the approaches till now are based on Information retrieval method i.e keyword based search is performed on both Job offer and the Job seeker profiles. The same kind of approach is used by Faliagka et al., [7,6], and Kessler et al.,[12]. There is also a lot of work that has been conducted on job recommender systems [20] [21][23] [24] [25] which match job descriptions with that of user profile. These strategies undoubtedly work pretty well also they are easy to implement as raw data is not required to be converted into any kind of structure and then to process it further but the problem with this method is that it is error prone as manual entry of the similar keys is required to be done. As the job market is very dynamic and flexible, therefore the skills and thus the keywords keep on changing. Alternatively there are approaches that are sector specific which helps in screening the profile of applicant and the job specification. For example, StackOverflow website contains huge information about the IT sector skills that a programmer may possess.

The mentioned frameworks such as ICED, ESCO, and DISCO, are there to capture concepts of skill and do not use the advantages of ontologies and the description logics (Colucci et al.,[9]), these frameworks currently support taxonomies. So the challenge is to use the capabilities of description logics and to get some useful facts, For ex. Experience and last use of the particular skill. In general, any approach that tries the problem of Job matching address one of the below research problem :-

– Ranking the Applicant for a certain job Posting.

– Ranking the jobs for a particular candidate.

– Identify the Gaps between the candidate's profile and the Job Posting.

– Identify the proper skill/training needed for a Job Seeker to be eligible to apply for a particular Job.

3. Problem Definition

The actual problem lies in ranking the jobs according to profiles of the Job Seekers according to the most relevant/ fit according to the profile. So there has to be a system that takes the profile of the candidates and Job description as an input and provides the rank of the Job as an output. Also, there has to be a knowledge base that acts as a database which is basically a super set of all the skills that a Job Seeker may possess and that holds all the skill base that is mentioned in the Job Description. Our job is to map the skills mentioned in the Job description with the Job Seeker skills using our knowledge base and define a function such that the ranking of the Job Seekers profile who applied for the Job is generated automatically by the inputs provided. Also, the other factors like function of the job and industry of both jobseeker and job, the experience of the jobseeker etc are to be considered to provide a relevant pool of jobs to jobseekers. These factors could be used as a filter to the recommendations made but for matching there must exist a function that does the job.

Say, We consider jd as job description and js as JobSeeker profiles list for which we should have a score list say $y \in R$ where each job is ranked according to the jobseeker profile. Now our goal is to determine a function f (jd, js) $\approx$ y that fits for every Job applicant. It is also necessary that a JobSeeker profile has to be a set which holds components that are there in a knowledge base (e.g. ICED, ESCO or DISCO etc)[Martinez-Gil et al.[1]]. Therefore, there is a job to score the eligibility of a number of possible jobs for a particular candidate, taking the following facts into consideration:

1. The elements that appear in either job offered or the applicant's profile may be of any order, as ordering does not play any key role and different order may not hamper anything and is not relevant in our case.

2. The size of the set of elements of the Job offer and applicant's profile may differ as the applicant may possess more or less skills required for the Job offered and it may affect the ranking later.

3. The elements present in the set jd can be replaced by the elements present in js at a certain cost to it. For example:- a person who possesses SVN as a skill may not face much difficulty in working on GIT but there is a certain cost that is attached to it as it can take time to switch to a familiar technology as well.

4. The Job function mentioned in the jd must match the mapped function list of js.

5. The Job sector/ industry/ domain plays a vital role and if the jd and js sector matches then it's an additional benefit.

4. Methodology

There is no single solution to this problem as every domain expert has different opinions and a different way to deal with the problem and that may give different results altogether. A model has to be trained first for each organization or user who wants to use the model to know their preferences [Martinez-Gil et al.[1]]. We have closely looked into the various parameters that affect the matching of jobs to the Candidates profile, validated the same manually and talked to many HR domain experts and considered many factors affecting the candidates opinion to opt for a particular job.

The Architecture

Fig.2 shows the overall process and the architecture of the approach adopted for the whole process. There are many data preprocessing that has to be done before the mentioned process which is also mentioned in the section below.

As shown in Fig.2 for finally ranking the jobs and to find suitable matched jobs, the profile of the jobseeker goes through various stages listed below.

1. Tika :- here Apache tika is used to parse the job seeker profile and to provide a unified form of data. It basically consumes multiple formats i.e. doc, pdf and provides the profile as a text file.

2. LT-TTT2 :- It takes the text file as an input along with the skills database and tags all the skills present in the jobseeker profile

3. Gate Pipeline :- it's combination Jape rules and other machine leaning library that identifies the various aspects from jobseeker profile. For eg. experience of a particular skill, number of projects done by applicant, number of organisations he has worked with etc.

4. Feature Extractor model :- It's a model which extracts all the identified features of the Applicant/Job seeker.

5. Enicher :- it's a model that enriches the extracted features from the Jobeseeker profile. Like adding synonyms, broadening the skills horizon, identifying and expanding the Job function etc.For this a enricher DB is required to be ingested along with the profile.

6. Weighing Model :- This is the model that gives the weighted feature matrix from the jobseeker profile.

7. Scoring Model :- Finally after the weighting matrix is obtained the scoring model gets the weighted feature matrix and the job pool as the input and provides the matched ranked jobs that are best suited according to the jobseeker profile.

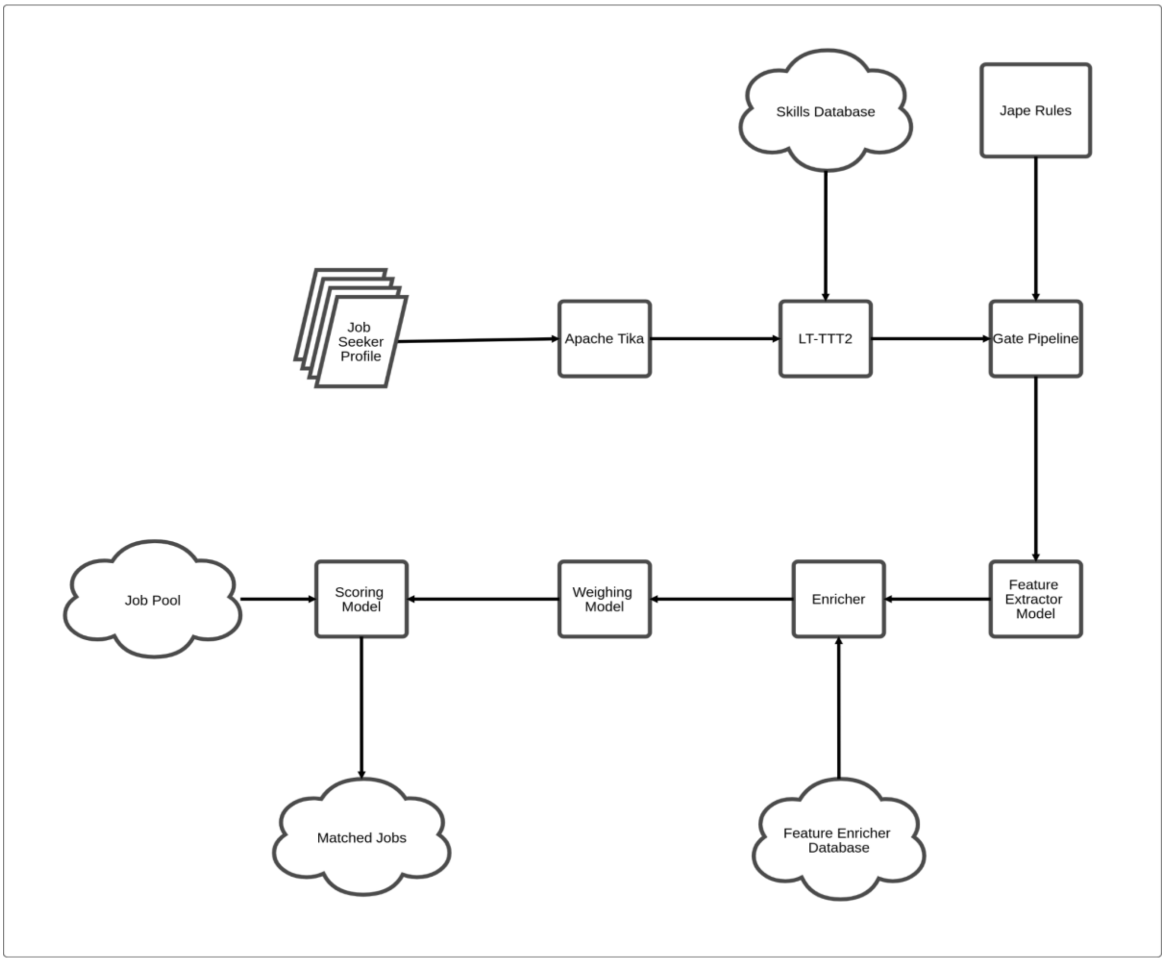

Fig:2 Proposed System Architecture

Dataset Description

For Skills enrichment DBpedia database has been used. The data is extracted and loaded as a graphical Dataset and the connected nodes are considered for the skill cloud. After several attempts and tuning we have reached to a conclusion that the nodes that are connected to maximum five level hoppings are the only ones which are useful, post that the data gets diluted and the results are not so good. The rest of the data was downloaded from Kaggle and through a classified job portal. The data is converted to an excel file having five columns, i.e Id, Function, Category, location and Resume.The number of instances for the different domain can be seen from Figure 3

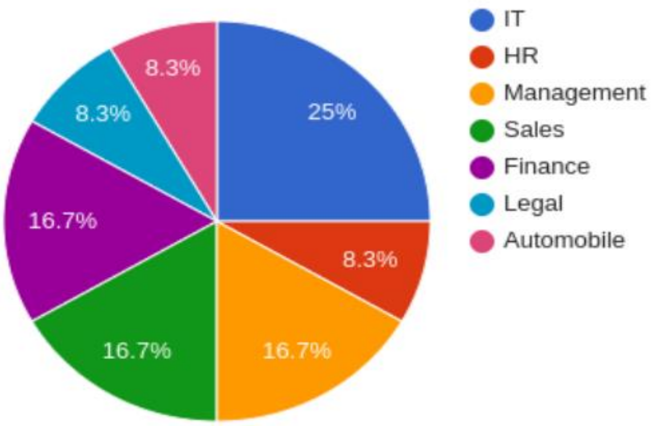

Fig:3 Data Description from different domains

Preprocessing

1. Skills Database :- Skills database is prepared by parsing data from DBpedia and storing the same in graph, the inter-relation between the skills were identified and finally stored as a flat structure to be used for the further processing purpose

2. Jape Rules :- Lot of profiles were examined and different profile structure were studied to write Jape rule to identify various Jobseeker Information

3. Feature Enricher Database :- Function prediction model was used and Function Cloud was obtained after studying the Jobseeker data. Skill cloud was also used to enrich Skills. Various synonym dictionaries were used to map synonyms as well.

4. Job Pool :- Job data was already predefined and well structured like Job Title, Skills to be required, location, function, industry etc.

5. Results

In this section, we will present the actual results obtained while working with the real data from the recruitment agency. Also, we have only considered the data that was most relevant to us and personal data like photo, reference etc was not taken into consideration. We have worked on 7 major domains that we got data for : IT, HR, Sale, Marketing, Finance, Legal,Automobile..

We are interested in forecasting the contextual factors for an employee who is presently working in one organization .If the precision factor is high enough, we may be able to utilise our model for employees looking for a new job profile. We created a prediction model using supervised machine learning. In our learning model, an instance is equivalent to a member who is employed in an institution.

S.No.	Feature	Range
1.	Skills	List<String>
2.	Designation	String
3.	Total Experience	Int
4.	Job Function	List<String>
5.	Period of Working	Date

Table 1 : Featured Enhanced by Predictive Model

The system is more of a ranking system and ranks all the jobs present in the system according to the Job Seeker profile. So it may be considered that the last job recommended might be the job that is most irrelevant to the job seeker. So for our experiment purpose we considered the top 50 recommendations for a particular domain that the system has recommended. Also, various factors mentioned have been used to filter the recommended Jobs post recommendation was made. Fig: 4, shows the importance vs feature characteristics graph to show how relevant the factor is for the recommendation the system is generating.

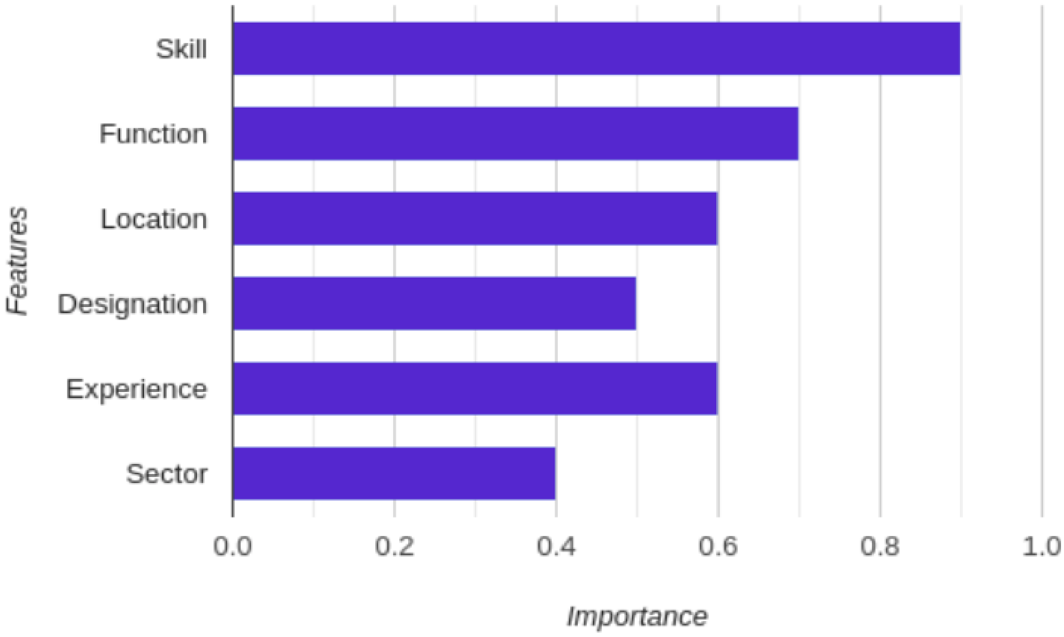

Fig:4 Importance Vs Feature Characteristics

The below Table shows different result accuracy rate on adding the different features. The experiment is performed for each domain on successively adding different feature vectors and the same was recorded as shown in Table 2.

	Skills	Function	Sector	Designation	Experience
IT	75%	85%	90%	92%	95%
HR	60%	75%	79%	80%	83%
Management	55%	72%	75%	80%	82%
Sales	58%	70%	74%	76%	78%
Finance	65%	75%	78%	81%	85%
Legal	60%	69%	77%	78%	80%
Automobile	60%	65%	72%	74%	77%

Table 2: Factors affecting accuracy in different domains

The model design is best suited for the first level of screening of the job by the Jobseeker. This would help the Jobseeker to classify the Jobs as per the requirements and easily identify the Jobs that are the best match to their profile. Also, it would help many recruitment agencies to recommend better jobs to the candidates and save a lot of time in manually screening the jobs for a particular candidate, it would in turn also increase their reliability for the Job seeker. The recommendations made by the model are currently for the varied industries but the model can be further enhanced to target specific industries which would make it more effective, and give better recommendations.

Fig: 5 Confusion Matrix using the proposed methodology

6. Conclusion

A recruitment consultancy/ agency works really hard to match jobs with the candidates profile and increasing both jobseekers and jobs makes it very difficult to do the matching process manually, thus there is a need for a system which can do the matching task automatically. The proposed system does the same and helps finding the relevant jobs for the Jobseekers of different domains. The proposed approach gave the accuracy of upto 95% for the IT domain. Though the results for other domains could not match the results of the IT domain, they are equally good. The proposed solution heavily relies on the skills database and we were able to get maximum data for the IT domain and hence the results validates our proposed approach. The model could grow if the taxonomy of the other domains could be collected from other data sources which has to be curated well with the help of HR/ Domain experts.

The experimental results showed that the proposed model showed high accuracy except for the jobs where the taxonomy database was weak. In future we plan to improve the system by extending the taxonomy, which is presently inclined more towards the IT domain.

References

[1] Martinez-Gil, Jorge, Alejandra Lorena Paoletti, and Mario Pichler. "A novel approach for learning how to automatically match job offers and candidate profiles." Information Systems Frontiers (2019): 1-10.

[2] Tinelli, Eufemia, Simona Colucci, Francesco M. Donini, Eugenio Di Sciascio, and Silvia Giannini. "Embedding semantics in human resources management automation via SQL." Applied Intelligence 46, no. 4 (2017): 952-982.

[3] Rácz, Gábor, Attila Sali, and Klaus-Dieter Schewe. "Semantic matching strategies for job recruitment: A comparison of new and known approaches." In FoIKS, pp. 149-168. Springer, Cham, 2016.

[4] Lu, Jie, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. "Recommender system application developments: a survey." Decision Support Systems 74 (2015): 12-32.

[5] Hauff, Claudia, and Georgios Gousios. "Matching GitHub developer profiles to job advertisements." In 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories, pp. 362-366. IEEE, 2015.

[6] Faliagka, Evanthia, Lazaros Iliadis, Ioannis Karydis, Maria Rigou, Spyros Sioutas, Athanasios Tsakalidis, and Giannis Tzimas. "On-line consistent ranking on e-recruitment: seeking the truth behind a well-formed CV." Artificial Intelligence Review 42, no. 3 (2014): 515-528.

[7] Faliagka, Evanthia, Athanasios Tsakalidis, and Giannis Tzimas. "An integrated e-recruitment system for automated personality mining and applicant ranking." Internet research (2012).

[8] Al-Otaibi, Shaha T., and Mourad Ykhlef. "A survey of job recommender systems." International Journal of Physical Sciences 7, no. 29 (2012): 5127-5142.

[9] Colucci, Simona, Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, Marina Mongiello, and Marco Mottola. "A formal approach to ontology-based semantic match of skills descriptions." J. UCS 9, no. 12 (2003): 1437-1454.

[10] Martinez-Gil, Jorge, Alejandra Lorena Paoletti, and Klaus-Dieter Schewe. "A smart approach for matching, learning and querying information from the human resources domain." In East European Conference on Advances in Databases and Information Systems, pp. 157-167. Springer, Cham, 2016.

[11] Hauff, Claudia, and Georgios Gousios. "Matching GitHub developer profiles to job advertisements." In 2015 IEEE/ACM 12th Working Conference on Mining Software Repositories, pp. 362-366. IEEE, 2015.

[12] Kessler, Rémy, Nicolas Béchet, Mathieu Roche, Juan-Manuel Torres-Moreno, and Marc El-Bèze. "A hybrid approach to managing job offers and candidates." Information processing & management 48, no. 6 (2012): 1124-1135.

[13] Martinez-Gil, Jorge. "An overview of knowledge management techniques for e-recruitment." Journal of Information & Knowledge Management 13, no. 02 (2014): 1450014.

[14] Hossain, Md Sabir, and Mohammad Shamsul Arefin. "Development of an Intelligent Job Recommender System for Freelancers using Client's Feedback Classification and Association Rule Mining Techniques." JSW 14, no. 7 (2019): 312-339.

[15] Gugnani, Akshay, and Hemant Misra. "Implicit Skills Extraction Using Document Embedding and Its Use in Job Recommendation." In AAAI, pp. 13286-13293. 2020.

[16] Rimitha, S. R., Vedasamhitha Abburu, Annem Kiranmai, and K. Chandrasekaran. "Ontologies to model user profiles in personalized job recommendation." In 2018 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), pp. 98-103. IEEE, 2018.

[17] Gugnani, Akshay, Vinay Kumar Reddy Kasireddy, and Karthikeyan Ponnalagu. "Generating unified candidate skill graph for career path recommendation." In 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 328-333. IEEE, 2018.

[18] Javed, Faizan, Phuong Hoang, Thomas Mahoney, and Matt McNair. "Large-scale occupational skills normalization for online recruitment." In Twenty-ninth IAAI conference. 2017.

[19] Liu, Kuan, Xing Shi, Anoop Kumar, Linhong Zhu, and Prem Natarajan. "Temporal learning and sequence modeling for a job recommender system." In Proceedings of the Recommender Systems Challenge, pp. 1-4. 2016.

[20] Hong, Wenxing, Siting Zheng, Huan Wang, and Jianchao Shi. "A job recommender system based on user clustering." J. Comput. 8, no. 8 (2013): 1960-1967.

[21] Malherbe, Emmanuel, Mamadou Diaby, Mario Cataldi, Emmanuel Viennet, and Marie-Aude Aufaure. "Field selection for job categorization and recommendation to social network users." In 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), pp. 588-595. IEEE, 2014.

[22] Patel, Bharat, Varun Kakuste, and Magdalini Eirinaki. "CaPaR: a career path recommendation framework." In 2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService), pp. 23-30. IEEE, 2017.

[23] Koh, May Fern, and Yew Choong Chew. "Intelligent job matching with self-learning recommendation engine." Procedia Manufacturing 3 (2015): 1959-1965.

[24] Al-Otaibi, Shaha T., and Mourad Ykhlef. "A survey of job recommender systems." International Journal of Physical Sciences 7, no. 29 (2012): 5127-5142.

[25] Kivimäki, Ilkka, Alexander Panchenko, Adrien Dessy, Dries Verdegem, Pascal Francq, Hugues Bersini, and Marco Saerens. "A graph-based approach to skill extraction from text." In Proceedings of TextGraphs-8 graph-based methods for natural language processing, pp. 79-87. 2013.

[26] Wang, Zhongqing, Shoushan Li, Hanxiao Shi, and Guodong Zhou. "Skill inference with personal and skill connections." In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 520-529. 2014.