ABSTRACT

The Purpose of this study is to gather some ideas of people skills and recommends the job according to their skills. An excellent job recommender system not only enables to recommend a higher paying job which is maximally aligned with the skill-set of the current job, but also suggests to acquire few additional skills which are required to assume the new position. In our experiments, we show that by jointly learning the representation for the jobs and skills, our model provides better recommendation for both jobs and skills.

1.INTRODUCTION

From the start of the commercialization of the internet in the late 1980s, the question was raised of how this technology could be leveraged in employee recruitment to enhance job seeker - vacancy matching. Even before the start of the world wide web, Vega already proposed a system to match job seekers and jobs, which could “be consulted by Minitel, using telephone number 3615 and selecting the LM/EMPLOI service”. i.e., the service allowed job seekers to send text messages in the form of search queries or their digital resume, over the telephone line, using a computer terminal called Minitel1. The service would compare words in the query/resume to a knowledge base, which used a fixed job taxonomy to return a set of potentially interesting vacancies for the job seeker. Although more than 30 years have passed since this early contribution, the usage of a fixed job taxonomy to extract information from a resume including “branch of industry” (industry) and “qualification” (skill)using “(a) dictionary specialized in the universe of employment”, seems vaguely similar to LinkedIn’s query processing method, which can be queried using your mobile phone2. Of course, this is a very simplified view on reality: Vega’s 200 simultaneous Minitel connections could not have served the currently close to 750 million LinkedIn users worldwide. Nonetheless, the problem of recommending the right job to job seekers remains as pressing as it was more than 30 years ago.

In this paper, we will provide an overview of the literature on job recommender systems (JRS) from the past decade (2011-2021). We will consider the different methods used in these systems, and consider these from a reciprocal, temporal, and ethical perspective. Furthermore, we will consider the influence of data availability on the choice of method and validation, and put extra emphasis on branching the large class of hybrid recommender systems used in this application domain. Our results suggest that JRS could benefit from a more application-oriented view: the reciprocal and temporal nature of JRS are infrequently discussed in the literature, while contributions that do consider these show considerable benefits. Furthermore, fairness is rarely considered in job recommender systems, and if it is considered, authors too often conclude that removing discriminatory features from the data is sufficient. However, recent scientific attention on fairness, in particular in candidate search engines, has introduced various metrics and algorithms, which we believe have the opportunity to be applied in the job recommender

domain as well. In accordance with recommender system literature, deep language models have also been more frequently applied in the domain of job recommender systems.

2.PRELIMINARIES

We will use the terms vacancy, job posting, and job somewhat interchangeably throughout this paper to represent the item in the classical recommender system setting, whereas job seekers are considered as users. Although, as in the early paper by Vega, job seekers are still often described by their resumes, some current e-recruitment systems allow for descriptions that move beyond self-descriptions of one's professional self. Here one should think of the social connections one can observe on (professional) social networks. When we speak of resumes, CVs, user profiles, or job seeker profiles, we assume these are synonyms and may contain additional information (such as social relations) beyond the self-description.

3.METHODS

As is common in the literature ,these are split into Content-Based Recommender Systems (CBR), Collaborative Filtering (CF), Knowledge based Recommender Systems (KB), and Hybrid Recommender Systems (HRS).

3.1 Content-Based Recommender Systems

Content-based recommender systems (CBRs) in the context of JRS are models which, to construct a recommendation, only use a semantic similarity measure between the user profile and the set of available vacancies. I.e., the semantic similarity is used as a proxy for estimating the relevance of each vacancy to the job seeker. In CBRs, one creates vector representations of the vacancy and user profile in an unsupervised way, i.e., the dimensions of these representations may not have an intuitive interpretation.

3.2 Collaborative Filtering

In collaborative filtering (CF), recommendations are based solely on behavioural data, typically stored in a user × items rating matrix. In the e-recruitment setting, like in the e-commerce setting, this matrix is usually filled with click behaviour (e.g., the rating matrix equals 1 if the job seeker (row) clicked on a vacancy (column) to see the vacancy details, 0 otherwise). Though, in case such interaction data is missing, also the sequence of previous job occupations can be used to fill the rating matrix .The latter case does require jobs to be categorized, such that they can serve as discrete items in the rating matrix. For simplicity, we will refer to the entries in the rating matrix as ratings, irrespective of the exact type of behavioural feedback.

3.3 Knowledge based Recommender Systems

Knowledge-based recommender systems as recommender systems having the conceptual goal to “Give me recommendations based on my explicit specifications of the kind of content (attributes) I want”, we will rather use the definition given by Freire and de Castro, who define it as “[Recommender systems] which rely on deep knowledge about the product domain (job) to figure out the best items (job vacancies) to recommend to a user (candidate)". In job recommender systems, this often implies that both job and candidate profiles are mapped to some predefined job ontology (i.e., the knowledge base), after which the two are matched.

3.4 Hybrid Recommender Systems

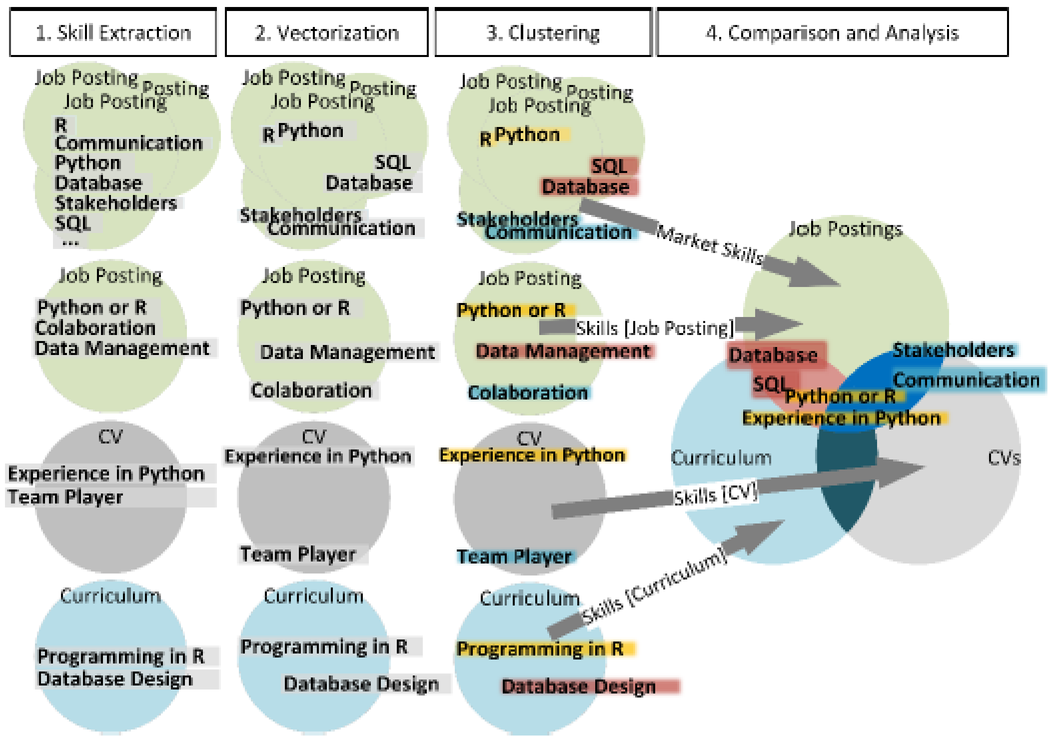
Hybrid recommender systems combine several models into one recommender system. Our aim here is to split this group into smaller, more homogeneous methods. To do so, the hybrid recommender systems split into those having a monolithic, or ensemble design. A monolithic design refers to hybrid recommender systems in which it is not possible to extract one component of the hybrid, and build recommendations based on this component independent of the other components in the hybrid, without altering the algorithm. On the contrary, ensemble methods do allow splitting the hybrid recommender system into at least two components that can (independently) be used for recommendation. I.e., the ensemble may consist of some models which by itself may be monolithic.

4. Extracting, Vectorizing, Clustering and Comparing Skills

Our goal was to help employers, job seekers, and educational institutions adapt to the market skills. Our recommendation system Skill Scanner determines which skills in the job postings are covered and which skills are missing. In this section we will de-scribe our pipeline to extract, vectorize, cluster, and compare skills.

4.1 Our Pipeline to Extract, Vectorize, Cluster and Compare Skills

For a certain job position, (1) Skill Scanner takes a CV, a job posting or a learning curriculum as input, (2) extracts the skills of the provided document, (3) compares the document's extracted skills to a skill set which represents the market's needs (market skills) and (4) returns information of which market skills are covered or missing in the provided document compared to the market's needs. To be able to compare the skills in the provided document to the market skills, we need to cope with the challenges of different levels of abstraction and synonyms among the skills in the uploaded document and the market skills. Consequently, we apply the following 4 steps when we gather the market skills and when we upload a document to be analyzed which are visualized as



1. Skill Extraction: Extract skill requirements.

2. Vectorization: Map skill requirements to a semantic vector space, where skills with similar meanings are closer together and skills with different meanings are farther apart.

3. Clustering: Cluster skill requirements to cope with the challenges of differ-ent levels of abstraction and synonyms.

4. Comparison and Analysis: Compute intersections among the skill sets of the provided documents and the market skills and visualize recommendations based on covered and missing market skills.

5.VALIDATION

Validation when lacking job seeker - vacancy interaction Many authors state in their introduction that there is a vast amount of online data available, commonly followed by a reference to the current LinkedIn user count . Although this may be true in terms of vacancies and job seeker profiles, researchers do not always have access to interactions between the two. Naturally, this also impacts the type of methods and validation that is used. In case interaction data is lacking, authors propose several strategies to still be able to evaluate their recommendations. One of these is to use one of the competition datasets for validation and training. As already discussed, approximately 32% of all contributions use one of the competition datasets, we also find that these were used after the competitions had finished. Expert validation is also used frequently for validation. During such expert validation, the quality of recommendations is inquired by a group of ‘experts’, which may be the researchers themselves, HR/recruitment experts, or sometimes students .Although the choice for expert validation is rarely discussed, we do find that for CBR and KB job recommender systems, approximately half of the contributions use expert validation

As mentioned by, job seekers may have different interpretations of a “job". That is, if the definition is too specific, each job would have too few observations for inference. Whereas if the definition is too broad, the recommendation may be accurate, but not precise, and therefore not useful. Furthermore, many details about the job may be missing from the job history, and job seekers tend to indicate their jobs at different levels of granularity . (e.g., are multiple related positions at the same firm one or multiple“jobs"?).

6. ETHICAL ASPECTS

The recent increased scientific attention on algorithm fairness will hopefully turn this tide, as online job portals are more likely to come under scrutiny by algorithm audits. Chen et al perform such an audit on the candidate search engines of Monster, Indeed, and CareerBuilder, testing these search engines on gender bias. Although all three search engines are found to not allow for direct discrimination on inappropriate demographics (e.g., by allowing to filter on gender), the researchers do find indirect discrimination both in terms of individual and group fairness. Geyik et al., compare several algorithms for ensuring group fairness in candidate recommender systems/search engines, with a focus on measures for equal opportunity and demographic parity, one of which has been implemented in LinkedIn.

Another concern is the usage of so-called “People aggregators", in which researchers or people from industry use a web crawler to obtain a substantial number of professional or personal profiles. Although this is sometimes with the user’s consent and also in many cases this remains unclear. We did not find any work in which user profiles are explicitly anonymized before processing.

7. LINKEDIN ASPECTS

Although many job recommender systems have been proposed in the literature, and the internet holds a considerable number of websites where job seekers can search for jobs, we do find it relevant to put some emphasis on contributions published by LinkedIn employees for a number of reasons. Even though many JRS have been proposed in the literature, few of these dominate the (global) recruitment market. Hence, we consider it likely that job seekers' perception will be biased towards the JRS these larger players are using.

Furthermore, although many job seekers also use other well-known general-purpose search engines/social networks, for which the large-scale argument also holds, LinkedIn includes algorithms specifically designed for establishing professional matches. Also, LinkedIn has been considerably transparent about the algorithms they use, and their respective performance, and is capable of testing these algorithms online via their A/B testing platform XLNT.

To facilitate different information needs, LinkedIn encompasses multiple search indices, which are combined into one to facilitate federated search. The job search engine itself is composed of a linear combination of several features, which can be grouped into four classes. First, a query-document semantic matcher. Second, an estimator for the searcher's skill with respect to the set of potential jobs. Third, an estimator for the quality of the document, independent of the query and/or searcher. And fourth, a collection of miscellaneous features such as textual, geographical and social features. LinkedIn's job recommender system consists of a GLMix model, including a large number of profile features from both the candidate, the job, their interactions, and the cosine similarity between user and job features in the same job domain. Additionally, the job ranking is adjusted based on the expected number of applications for each job in such a manner that the job seekers are better spread over the set of potential jobs, without reducing the relevance of the high ranked jobs in the recommendation too much.

8.CONCLUSION

In this paper, we have considered the job recommender system (JRS) literature from several perspectives. These include the influence of data science competitions, the effect of data availability on the choice of method and validation, and ethical considerations in job recommender systems. Furthermore, we branched the large class of hybrid recommender systems to obtain a better view on how these hybrid recommender systems differ.

REFERENCES

[1] Fabian Abel, Andras Benczur, Daniel Kohlsdorf, Martha Larson, and Robert Palovics. RecSys challenge
2016: Job recommendations. In *Proceedings of the 10th ACM conference on Recommender Systems*, pages 425–426, 2016.

[2] Fabian Abel, Yashar Deldjoo, Mehdi Elahi, and Daniel Kohlsdorf. RecSys challenge 2017: Offline and
online evaluation. In *Proceedings of the eleventh ACM Conference on Recommender Systems*, pages
372–373, 2017.

[3] Charu C. Aggarwal. *Recommender systems*. Springer, 2016.

[4] Shibbir Ahmed, Mahamudul Hasan, Md. Nazmul Hoq, and Muhammad Abdullah Adnan. User interaction
analysis to recommend suitable jobs in career-oriented social networking sites. In *2016 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–6. IEEE, 2016.

[5] Shaha T. Al-Otaibi and Mourad Ykhlef. A survey of job recommender systems. *International Journal
of Physical Sciences*, 7(29):5127–5142, 2012.

[6] Nikolaos D Almalis, George A Tsihrintzis, and Evangelos Kyritsis. A constraint-based job recommender
system integrating FoDRA. *International Journal of Computational Intelligence Studies*, 7(2):103–123,
2018.

[7] Dhruv Arya, Viet Ha-Thuc, and Shakti Sinha. Personalized federated search at linkedin. In *Proceedings
of the 24th ACM International on Conference on Information and Knowledge Management*, pages
1699–1702, 2015.

.