

# Job Recommender Systems: A Review

Corné de Ruijt <sup>1</sup> Sandjai Bhulai <sup>1</sup>  
c.a.m.de.ruijt@vu.nl s.bhulai@vu.nl

<sup>1</sup>Faculty of Science  
Vrije Universiteit Amsterdam  
Amsterdam, the Netherlands

November 29, 2021

## Abstract

This paper provides a review of the job recommender system (JRS) literature published in the past decade (2011-2021). Compared to previous literature reviews, we put more emphasis on contributions that incorporate the temporal and reciprocal nature of job recommendations. Previous studies on JRS suggest that taking such views into account in the design of the JRS can lead to improved model performance. Also, it may lead to a more uniform distribution of candidates over a set of similar jobs. We also consider the literature from the perspective of algorithm fairness. Here we find that this is rarely discussed in the literature, and if it is discussed, many authors wrongly assume that removing the discriminatory feature would be sufficient. With respect to the type of models used in JRS, authors frequently label their method as ‘hybrid’. Unfortunately, they thereby obscure what these methods entail. Using existing recommender taxonomies, we split this large class of hybrids into subcategories that are easier to analyse. We further find that data availability, and in particular the availability of click data, has a large impact on the choice of method and validation. Last, although the generalizability of JRS across different datasets is infrequently considered, results suggest that error scores may vary across these datasets.

## 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 [116] 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 Minitel<sup>1</sup>. 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 [75], which can be queried using your mobile phone<sup>2</sup>. Of course, this is a very simplified

---

<sup>1</sup><https://en.wikipedia.org/wiki/Minitel>

<sup>2</sup>Parts between brackets are job attributes in the job taxonomy used by LinkedIn, to which queries are matched using dictionaries [75]

view on reality: Vega’s 200 simultaneous Minitel connections could not have served the currently close to 750 million LinkedIn users worldwide [78]. 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. What remains somewhat unknown, is how well results on JRS generalize across datasets: the only study that considers this question shows that error metrics may vary greatly over different datasets.

This paper has the following structure. Section 2 discusses some earlier literature surveys, thereby providing an argument for our interest in writing this review. Also, it discusses the method used for obtaining and selecting literature, and provides a brief discussion on some datasets and terminology we will frequently mention in this paper. Section 3 discusses our findings, in which Section 3.1 classifies the job literature into the familiar recommender system taxonomy, while at the same time branching the large class of hybrid contributions. The remainder of Section 3 considers auxiliary topics such as the influence of data science competitions (Section 3.2), validation of JRSs (Section 3.3), JRSs taking a temporal and/or reciprocal view (Section 3.4), ethical considerations in JRS (Section 3.5), and considers contributions discussing job search and recommendation at LinkedIn (Section 3.6). Last, Section 4 draws a conclusion and discusses directions for further research.

## 2 Method and preliminaries

### 2.1 A brief discussion of previous surveys

Before we discuss the literature, it is useful to observe that in recent surveys on applications of recommender systems, job recommender systems and (more general) recommender systems in e-recruitment, are frequently not included. I.e., in the well-cited review on applications of recommender systems, Lu et al. [81] do not mention the application area of e-recruitment, the same holds for the earlier review by Felfernig et al. [41]. Also, although most papers on neural networks in job recommender systems were published after 2018, the survey on (deep) neural networks in recommender systems (including a section on application areas) also neglects this application [11]. From the HR perspective, job search and recommendation are also not always mentioned as an application area, as opposed to candidate selection, while in the end these systems do determine who will be in the applicant pool in the first place [111].

One possible explanation could be that, from a technical perspective, the problem of job search and job recommendation is little different from a general information retrieval/recommendation task. Job seekers frequently use ‘general-purpose’ search engines and online social networks to search for jobs (e.g., [56, 32, 66]). Furthermore, many job recommender systems we will discuss in this paper could very well be used in other application areas (and vice versa). Nonetheless, we will argue that factors such as the large amount of textual data, the reciprocal and temporal nature of vacancies, and the fact that these systems deal with personal data does require a tailored approach, and the sheer volume of contributions make it clear that this

application area should not be neglected.

Previous surveys on job recommender systems, which consider JRS contributions before 2012, include Al-Otaibi and Ykhlef [5] and Siting et al. [110], though especially the latter survey is very limited in scope. More recent is the survey on recommender systems in e-recruitment by Freire and de Castro [43]. Although our work has some overlap, we especially wish to address some of the limitations of the work by Freire and de Castro in this paper.

Even though the work by Freire succeeds in collecting a substantial number of contributions in the JRS application domain, they seem to fail to properly classify these contributions, making it difficult to see patterns in this literature. A clear example of this is that approximately 20% of the contributions discussed in their paper is labeled as hybrid, whereas another 33% is being labeled as “other”. Although the reader would later find that the “other” category includes for 25% contributions using (deep) neural networks, this still leaves a large number of contributions with an unsatisfying label. Furthermore, as shown by Batmaz et al. [11], there is a considerable development within the class of (deep) neural networks applied to recommender systems, which we also find in job recommender systems. This aspect is neglected by Freire and de Castro.

The classification given by Freire and de Castro is understandable, given that so many contributions use mixtures of collaborative filtering and content-based techniques, and given that these are presented by the contributions themselves as hybrids. However, these labels do not provide much insight into what these contributions actually entail. Furthermore, Freire and de Castro [43] focus solely on methods and validation, whereas we, among other subjects, will also take into consideration ethical considerations. We will also put special emphasis on job recommender systems which, often successfully, take into account the reciprocal and temporal nature of job recommendations.

## 2.2 Preliminaries

As we will discuss in Section 3.2, many contributions use one of the data sources made available through data science competitions for training and validating job recommender systems. Most noticeably, these include the *RecSys 2016* and *RecSys 2017* competitions ([1], and [2] respectively), using a dataset from the job board Xing[121], and the *CareerBuilder 2012* Job Recommendation Challenge [60], which was hosted by CareerBuilder on Kaggle[21, 61]. All three datasets contain data with respect to candidate profiles, vacancies and online interaction between the two. Another resource often used is the Occupation Information Network (*O\*NET*)[93], an English-based job ontology that is frequently used in knowledge-based job recommender systems (see Section 3.1.4).

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[116], 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. Last, we will sometimes speak of “textbook” or “off-the-shelf”, by which we mean methods one can find in popular machine learning/pattern recognition textbooks such as by Bishop [16] and Flach [42], or Aggarwal [3] in the case of commonly used recommender systems.

## 2.3 Method

To obtain the set of contributions we will discuss in this paper, the following method was employed. English contributions published between January 1st 2011 and January 1st 2021, published as an article in a scientific journal or in conference proceedings were searched using Google Scholar, using key phrases *job recommender systems*, *job recommendation*, and *job matching*, where additionally we replaced “job” by “occupation” in

Table 1: Contributions per year per JRS category

Year	Dataset	Method		Ensemble hybrid Cascade	Feature aug.	Switching	Weighted	Monolithic hybrid		KB
		CBR	CF					MM-SE	DNN	
2011	O	[63]		[82]	[35] [36, 34, 40, 49, 86]	[19, 53] [52]		[118] [98]		
2012	O									[48]
2013	O									[64, 65]
2014	O									[75]
	CB12									
2015	O	[27]								
2016	O	[37]								
	RS16									
2017	O	[9]	[4, 80] [102, 72]	[46] [17]	[71]	[106] [129]	[77, 130] [99, 73, 133, 22] [114]	[118] [79]	[127, 128]	
	CB12									
	RS17									
2018	O	[108]		[76, 117, 124, 105]	[47]		[74, 15]	[85, 29, 59, 122]	[100, 132]	[83] [103] [6]
	CB12									
	RS16									
2019	O	[67]								
2020	O	[45, 28]	[90]							
	CB12									
	CB12, RS17, O			[68]						[84, 13, 70, 91, 125] [14, 58, 101]

Dataset abbreviations: Career Builder 2012 (CB12), RecSys 2016 (RS16), RecSys 2017 (RS17). ‘O’ implies another dataset was used in these papers.

these phrases. Furthermore, a forward and backward search was applied, which after reviewing the titles and abstracts resulted in 192 contributions.

Next, contributions on career trajectory recommendation were omitted, that is, recommender systems that only recommend a job type (such as “data scientist”), but not a specific occupation at a given organization at a specific time. Furthermore, also papers on employee selection, recommender systems specifically for freelancers, and candidate recommender systems (i.e., recommending candidates for open vacancies) were filtered out. We further removed contributions without (some form of) empirical validation of their recommender system. The resulting set contained 87 contributions, which we will discuss in this paper.

## 3 Results

### 3.1 Methods in job recommender systems

As discussed in Section 2.1, although the work by Freire and de Castro [43] is considerable, our critique lies in how they classify the different job recommender systems. As is common in the literature (e.g., [3]), these are split into Content-Based Recommender Systems (CBR), Collaborative Filtering (CF), Knowledge-based Recommender Systems (KB), and Hybrid Recommender Systems (HRS). However, given that so many contributions are hybrid, we will further make a distinction between *monolithic* and *ensemble* hybrid recommender systems [3, p. 200]<sup>3</sup>, and split the class of ensemble hybrids using the classification introduced by [20]. The interpretation of monolithic and ensemble hybrids, and the different ensemble hybrids, will be discussed in Section 3.1.3.

#### 3.1.1 Content-based JRS

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. Many authors use Bag of Words (BoW) with TF-IDF weighting [90, 63, 28, 37, 27], though also Latent Dirichlet Allocation is used [9], and the more recent word2vec [45, 115, 57]. Interestingly, CBR contributions have been relatively stable in the past 10 years, but also, they were not part of the top contributions during the 2016 and 2017 RecSys competitions (see Table 1).

Perhaps the main challenge in content-based JRS is that, in the words of Schmitt et al. [106], “*Job seekers and recruiters do not [always] speak the same language*”. I.e., job seekers and recruiters may use a different terminology to describe jobs, knowledge, or skills. As a result, two entities having the same meaning may end

<sup>3</sup>[3] also considers the mixed design, which we did not find in the JRS literature, and is therefore excluded

up in different vector representations when described by the recruiter or by the job seeker. Unfortunately, this discrepancy is often neglected in content-based JRS.

### 3.1.2 Collaborate filtering JRS

In collaborative filtering (CF), recommendations are based solely on behavioral data, typically stored in a user  $\times$  items rating matrix. In the e-recruitment setting, like in the e-commerce setting, this matrix is usually filled with click behavior (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 (e.g., [72]). 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 behavioral feedback.

The literature commonly distinguishes between two types of CFs: memory-based CF and model-based CF [3]. In memory-based CF, recommendations are created using a  $K$ -nearest neighbor (KNN) type of approach. That is, for some user  $u$ , one tries to find either  $K$  users similar to  $u$  (user-based CF), or  $K$  items similar to the items  $u$  has already ranked (item-based CF). Here, the similarity is always based on the entries of the rating matrix. Contributions using textbook CF methods include Lee et al. [72], Reusens et al. [102], Ahmed et al. [4], Liu et al. [79], where the latter two applied it to the RecSys 2016 dataset. Lacic et al. [68] compare several auto-encoders to encode user interactions, based on which similar users are determined.

Model-based CF attempts to fill the missing values in the rating matrix using regression-based models, based solely on the rating matrix. However, likely due to the large amount of textual data, which can ideally function as features in such regression models, we are not aware of any studies using model-based CF in job recommender systems. On the other hand, although we classified regression models using textual data as hybrid, it should be noted that these models have a strong CF flavor. Even though textual features are used, their weights in the regression model are determined by behavioral feedback. Hence, even though not only behavioral data is used, still the resulting recommendation is largely determined by what vacancies similar job seekers have interacted with. Only the definition of ‘similar’ has changed, as it now partly includes features based on content.

### 3.1.3 Hybrid JRS

As discussed earlier, 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, we follow Aggarwal [3, pp. 199-204], who split the hybrid recommender systems 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.

**Model-based methods on shallow embeddings** In the class of monolithic designs, we again consider two classes. We will refer to the first class as *model-based methods on shallow embeddings* (MM-SE). These approaches commonly use off-the-shelf machine learning methods such as support vector machines [98], naive Bayes [95], or gradient boosting [126, 122], though also convolutional neural networks (CNNs) [85] are used. What unifies these contributions, is that the textual data is mapped to a vector space using linear or small order transformations. Although we will leave the definition of shallow or small order somewhat vague on purpose, these embeddings do have in common that this order is explicit in the method, whereas in deep embeddings this could be arbitrarily large. These shallow embeddings include TF-IDF weighted document representations [122], representations based on a predefined job classification [126, 95], word2vec [85], and a probabilistic stacked denoising auto-encoder [25].

Also graphical models are included in the MM-SE class of monolithic hybrids. Dave et al. [29], map jobs and skills to two matrices sharing the same latent space. Although the mapping is trained independently from O\*NET, the resulting vector representations showed considerable resemblance to the O\*NET ontology. Jiang et al. [59], extract a large number of features from their dataset of the vacancies, job seeker profiles, and their interactions, to train a graphical model in which clicks are predicted based on estimates of whether a (job seeker, vacancy) pair is of some latent type  $z$ . They also make use of O\*NET and use bin-counting (i.e., [131, Ch. 5]) to obtain a numeric representation of some categorical features. Although the model provides an explicit click probability, as opposed to only predicting the appropriate ranking of items, the paper does mainly compare its approach with learning to rank models such as AdaRank and RankBoost, where the model is found to be favorable.

Three contributions model the job recommendation problem with a somewhat different objective function, though, we still label these as MM-SE. Dong et al. [38] and subsequent work [26] propose an MM-SE monolithic hybrid. Contrary to the approaches discussed so far, they consider the problem as a reinforcement learning problem. Given a successful or failed recommendation, a reward function is updated, taking as input the representation of a job seeker and a vacancy. To speed up computations, vacancies are represented in a binary tree. Wang et al. [118] not only consider what job should be recommended, but also when this recommendation should be made, based on estimations of when a job seeker will switch jobs.

**Deep neural networks** Apart from shallow embeddings, also deep representations have become common strategies, accounting for approximately 50% of all contributions in 2019 and 2020 (see Table 1). Since this class consists of only deep neural networks (DNNs), we will also refer to this class by DNN. Many studies follow a similar approach as language models such as BERT [33] or ELMo [97], and extend on these embeddings by adding additional hierarchical attention networks [91, 100]. Also embeddings based on CNNs are common [70], or mixtures of the above approaches [84, 13, 14, 58, 125]. Siamese neural networks have also been used for this purpose [107].

The increasing usage of DNNs in recommender systems is not limited to recommending jobs, but holds for recommender systems in general [11]. Although it is too soon to draw firm conclusions, the absence of MM-SE contributions in the last two years is somewhat striking, giving that these were quite common in 2017 and 2018. Furthermore, some contributions did benchmark their proposed DNN with models that we may label MM-SE, e.g., using word2vec or BoW to obtain vector representations, and found the DNN to be favorable [100, 91, 132]. Also, in some DNN contributions, the neural network on top of deep job/user/session embeddings is compared to other commonly used machine learning algorithms, such as gradient boosted decision trees (GBDT), with the latter using the same job/user/session embedding as the proposed DNN. In such comparisons, DNNs are also found to outperform other machine learning models on the same embeddings [84, 58].

To what extend this argument is completely satisfying remains somewhat unknown. If we view the architecture on top of initial job/user/session embeddings as automated feature engineering, then comparing these models with models without additional feature engineering is perhaps not completely fair. Moving away from the application of job recommender systems, DNNs also have not always been outperforming gradient boosting or ensemble techniques in recent data science competitions, in which participants put more effort on feature engineering for these boosting/ensemble models [55].

**Ensemble hybrids** Following Burke [20], we split ensemble hybrids into four classes. *Cascade hybrids* refine the recommendation given by another recommender system. These commonly include (gradient) boosting models [89, 76, 94, 119, 120, 117, 105], where in particular XGBoost[24] is popular. It should be noted that all these contributions were competitors in either the 2016 or 2017 RecSys competitions, with [120] and [117] being the winning contributions to the 2016 and 2017 competitions respectively. The problem has also been addressed from a learning to rank perspective, using LambdaMART [54]. Besides boosting, also refinements using common CBR or CF methods have been proposed [30, 124, 82].

Gui et al. [46] propose the so-called benefit-cost model, a re-ranking procedure for recommender systems that takes into account downside in recommender systems, and which is validated on a job recommender problem. To our knowledge, this contribution is the only contribution that raises the question of whether in some scenarios a recommendation should be given at all. Ill fitted recommendations could hurt the trust job seekers have in the recommender system, making them decide not to use it [69].

Borisuk et al. [17] refine an earlier job recommendation by predicting the number of applications to each vacancy, and pushing less popular vacancies to the top of the recommendation list. The idea behind this strategy is that, contrary to the e-commerce recommender setting, it can be hurtful to have too many applicants on a single vacancy. Recruiters will have to spend at least some time on each application to evaluate the fit and communicate the result of this evaluation to the job seeker (e.g., communicate a rejection or invite the job seeker for an interview). Furthermore, Borisuk et al. [17] also find that pushing less popular vacancies does increase the number of applications to those vacancies, hence causing a better spread of the applicants over the vacancy portfolio, though this is conditioned on whether the less popular vacancies are sufficiently relevant.

Similar to cascade hybrids is the *feature augmentation* class, in which case the result of the previous recommender system in the sequence is not so much refined, but simply used as an input for the next model. The somewhat similar approaches by Diaby et al. [35, 36], Diaby and Viennet [34], and Malherbe et al. [86], Faliagka et al. [40], Gupta and Garg [49], Guo et al. [47] use such an approach, in all cases applying off-the-shelf machine learning methods on top of an initial CBR.

In *weighted hybrids*, the output of the separate models is combined using some (possibly non-linear) combination of the predicted scores. Commonly, CBR and model-based CF are in this way combined [99, 73, 74, 114, 52, 22, 15]. Others combine a large number of textual and behavioral features [133, 130, 19], which we consider an ensemble hybrid as some of these features may be used for job recommendation by themselves. The output of several off-the-shelf machine learning methods has also been used for this purpose [77].

A common problem in CF is the cold-start problem: new users/items have not given/received any behavioral feedback yet, making CF-based recommendations difficult. Although multiple hybrid approaches can be used to resolve this problem, perhaps the most direct approach is to use *switching hybrids*. In JRS, this most often implies that the recommender system uses CF by default. However, if an item or a user has insufficient data, the recommender system switches to CBR. Contributions using such approach with off-the-shelf CF and CBR methods include [129], and [106].

### 3.1.4 Knowledge-based JRS

Although Aggarwal [3] define 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 [43], 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.

A common strategy to generate job recommendations is then to compute the similarity between the candidate profile and vacancy in the ontology space [104, 64, 88, 65, 127, 128, 48, 109, 6, 83], where the overlap can be computed by for example the Jaccard index ([48]). Although one can imagine that the construction of such ontologies can take considerable effort, they have been used successfully in practice by, for example, LinkedIn [75] or Textkernel [113]. Nudging users to complete their profile by recommending skills from the ontology has also been shown to be a successful strategy to improve such recommendations [10]. Mapping jobs and candidate profiles to a shared ontology also provides a solution to the discrepancy between how job seekers and recruiters define jobs without requiring behavioral feedback.

An advantage of using job ontologies is that this also simplifies the implementation of keyword-based search

engines and simplifies filtering. Gutiérrez et al. [50] consider such an approach, in which an interactive recommender system was built on top of the knowledge-based recommender ELISE [39], where users are able to filter jobs based on travel distance or the type of contract. The recommender system also explains why the recommendation is (not) given, e.g., by indicating that a required skill is not mastered by the job seeker. Participants in the study indicated that the tool allowed (citing the authors) “*greater autonomy, a better understanding of needed competencies and potential location-based job mobility*”.

Reusens et al. [103] compare several CF approaches with keyword search in terms of recall and reciprocity, both for job recommendation and job seeker recommendation. With respect to job recommendations, although traditional CF approaches performed acceptably in terms of recall, they were inferior to keyword search in terms of reciprocity. The authors did find that using a ‘reversed’ rating matrix for job recommendation (that is, building a rating matrix on whether recruiters saved job seeker profiles for specific vacancies) improved reciprocity even beyond keyword search, though with a trade-off in terms of recall.

### 3.2 Competitions

Competitions have played an important role in the job recommendation literature. These competitions are settings in which organizations share a dataset, an objective, and an error measure with respect to this objective, typically via a competition platform such as Kaggle[61]. Teams can enroll in these competitions, and in the case of job recommender competitions, the goal for each team is to construct a job recommendation for a set of hold-out users. As already hinted by Table 1, to our knowledge three competitions have had a considerable impact on the job recommender literature: 1) The 2012 Careerbuilder Kaggle competition [60], 2) the RecSys 2016 competition [1], and 3) the RecSys 2017 competition [2]. The latter two both used a dataset from the German job board Xing [121].

Besides being used for contributions to the competitions themselves, the datasets are also commonly used after completion of the competition, e.g., to train and validate job recommender systems when no dataset is available. Given that approximately 32% of all job recommender system contributions use a dataset originating from one of these contributions, these datasets have had a considerable influence on the job recommender literature. Although many models proposed for the RecSys 2016 and 2017 competitions were cascade or weighted hybrids (and in particular gradient boosted trees were quite successful [120, 117]), there is no free lunch: all contributions show that a considerable time and effort was spent on constructing useful features to represent job seekers, jobs, and their interaction.

We will emphasize two more observations with respect to competitions. First, the 2012 CareerBuilder dataset contains user interactions, hence could be used to evaluate a collaborative or hybrid recommender system. However, it is frequently used to evaluate content-based recommender systems (see Table 1). Second, to our knowledge, only Lacic et al. [68] compare their proposed recommender system over multiple datasets: the Careerbuilder dataset, the 2017 RecSys dataset, and a private dataset originating from the student job portal Studio Jobs[112]. What is striking, is that the performance of the different models differs considerably across datasets, with no unanimous agreement across data sets and across error measures which model should be preferred. I.e., the results are rather dataset dependent. However, the authors do find that from the set of models they consider, using variational auto-encoders to embed user sessions gave overall the best performance.

### 3.3 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 [78]. 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 in Section 3.2, approximately 32% of all contributions use one of the competition datasets, but from Table 1, 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 (e.g., [85]). 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 [63, 64, 27, 65, 9, 127, 128, 57, 115, 88, 109, 45].

Another way to obtain behavioral feedback when interaction data is lacking, is by using previous  $N$  jobs in the job detail section of a resume to predict the  $N + 1$ -th job. This does come with the challenge of rightfully defining a job. As mentioned by [31], 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 [31] (e.g., are multiple related positions at the same firm one or multiple “jobs”?).

A somewhat interesting observation with respect to the best performing contributions during the RecSys 2016 and 2017 competitions is the fact that none of the best performing contributions were CBRs. This may be because interaction data was given, hence, why not use it? However, also in other hybrid recommender systems, incorporating behavioral feedback is found to improve the recommender system, even if one has access to a well-defined and validated job ontology [75].

**Choices in negative sampling** In case interaction data is used for validation, one mostly only observes those jobs that the job seeker has positively interacted with. I.e., the job recommender system problem is most often modeled as an unary classification problem, and therefore training requires defining a set of negative samples. We consider two types of behavioral feedback, which commonly lead to somewhat different approaches in negative sampling. We consider using previously held jobs in the resume as positives as Type 1 behavioral feedback, whereas Type 2 behavioral feedback is obtained from observed online job seeker - vacancy interaction.

For Type 2 behavioral feedback, common strategies for defining negatives include using shown but skipped items [54, 59, 70, 79, 89, 91, 94, 99, 101, 120, 126, 130, 133, 47], picking negative samples at random (not per se uniform) [13, 124, 125, 73, 29], replacing the job (but not the candidate and further context) at random [132], using vacancies of which the vacancy details were shown, but did not lead to an application [101, 100], or if the method allows for sparse matrices (such as in some matrix factorization methods): using all possible vacancy-user interactions [74, 98, 71, 68, 19, 102, 4, 72, 80, 22, 15, 83, 105]. Others incorporate negative sampling into the estimation method itself [76, 14]. Some datasets include even more user actions besides those of Type 2, such as deletion of stored vacancies from a profile [119, 117], or whether the candidate was hired/rejected [58], which can be used to define positive/negative instances.

In case the behavioral data is of Type 1, also all possible pairwise job seeker-vacancy interactions are used. As discussed in Section 3.3, this is somewhat less memory demanding as jobs, in this case, are defined at a lower level of granularity [95, 77, 52].

### 3.4 On the temporal and reciprocal nature of job recommendations

In Borisyuk et al. [17], the authors note on interaction with vacancies posted at LinkedIn that “*Since users like to apply for new jobs, our [previous] job recommendation system tends to always show new jobs more often than old jobs*”, a relationship that is also found by [67]. Furthermore, what is interesting about the winning RecSys 2016 contribution is that the model explicitly takes into account the recency of items,

something which contributions with similar methods (i.e., using XGBoost) did not.

Whether the claim by Borisyuk et al. (users like to apply to new jobs) holds the question. The relationship could also be reversed: recent vacancies are shown more on top of the recommendation list, and therefore more likely to be clicked. Furthermore, given that vacancies with high demand will find a candidate faster, there is also possibly a survival bias: vacancies with longer lead times are likely to be vacancies with low demand, as those with high demand were already removed from the platform (i.e., as they found a suitable candidate). Nonetheless, irrespectively of the causal chain, we would argue that the vacancy lead time should be taken more into account in job recommender systems, whereas we find that in most literature this aspect is currently neglected.

In one early contribution of job recommender systems [87], the authors represent the job recommendation problem as a reciprocal recommendation problem. They state that “*Theory shows that a good match between persons and jobs needs to consider both, the preferences of the recruiter and the preferences of the candidate*”. Although some contributions do take this reciprocal nature into account, and usually with success (e.g., [17, 103, 70]), most contributions consider the one-dimensional job seeker perspective, neglecting possible harm to the employer and/or job portal.

For example, the long tail problem in recommender systems [3, p. 33] also plays a role in job recommender systems and job search engines (e.g., [32]), which in extreme cases may lead to receiving over 1500 applications per vacancy, as was the case for office jobs at the United Nations office in New York [18, p. 62]. However, the probability of hiring a candidate seems to be concave increasing in the size of the applicant pool [17]. I.e., growing the size of the applicant pool is likely to lead to diminishing returns. In case there is a large applicant pool for a single vacancy, this not only implies that the recruiter will have to sift through more resumes, but also has to reject more (possibly also well-suited) candidates, which may damage the employer brand. Furthermore, as also shown by Borisyuk et al. [17], job seekers do apply to less popular but still relevant vacancies, if these are pushed to the top of the recommendation list.

### 3.5 Ethical aspects in job recommender systems

In their literature review on applied machine learning in human resource management, Strohmeier and Piazza [111] state that “*Ethical and legal aspects are not broadly considered in current research, only equality of treatment and protection of privacy issues are discussed*”. Furthermore: “*A few of the research contributions address equality of treatment [...], mostly by excluding potential discriminatory features from mining*”. Unfortunately, in job recommender literature, equality of treatment is also rarely considered, and if it is considered, the authors indeed come to the conclusion that simply excluding the discriminatory feature would be sufficient (e.g., [100]).

Excluding discriminatory features may seem logical from the perspective of preserving privacy, or when trying to eliminate human bias, but at the same time this may actually hurt equality of treatment. To check whether the algorithm is discriminating against a certain group, one would require these discriminatory features. Following the argument made by Perez [96, Ch. 3-6]<sup>4</sup>, the problem of (gender) inequality is perhaps caused by the fact that the impact of HR/recruitment policies, and as we argue here, the choice for implementing a certain job recommender system, are not tested against specific groups. As a result, the chosen policy is often (in Perez’ argument male-) biased.

One can easily find examples of how sensible features extracted from a resume or vacancy may lead to inequality of treatment, which is also referred to in the literature as *fairness*. Trivial ones include work experience (age-biased), or first name (gender-biased), but many are also less trivial. These may be the type of words used in a vacancy text [12, Ch. 4], or examples from O’Neil’s well-known *weapons of math destruction*[92, Ch. 6], of how online (professional) social activity may be gender-biased, or how commuting time may be related to welfare (especially if the office is located in, say, the city center of London). Hence,

---

<sup>4</sup>Perez actually provides an argument that this holds more general, Chapters 3 to 6 discuss gender inequality at the workplace.

these should make us aware that it is not an option to assume that if a (discriminatory) feature is not observed, it does not exist.

The recent increased scientific attention on algorithm fairness [8] will hopefully turn this tide, as online job portals are more likely to come under scrutiny by algorithm audits. Chen et al. [23] 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. [44], 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 (e.g., [35, 34, 40]), also in many cases this remains unclear. We did not find any work in which user profiles are explicitly anonymized before processing.

### 3.6 JRS at scale: notes from LinkedIn

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 [51]. 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 [123].

LinkedIn’s job recommender system/search engine is largely described in [62, 75], respectively. To facilitate different information needs, LinkedIn encompasses multiple search indices, which are combined into one to facilitate federated search [7]. 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 [130]. 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 [17].

From considering the different algorithms and their validation, a few observations can be made. Both the JRS and JSE quite heavily rely on the set of skills users indicate on their profile. The algorithms thereby benefit from LinkedIn’s policy to nudge users to complete their list of skills, or to provide endorsements to other users, partly by recommending skills to add to their profile [10]. Also, of all individual classes of features, the estimate of one’s expertise seemed to lead to the largest improvement in model performance. Given that endorsements from other users have a considerable influence on this estimate, the JRS/JSE seems to benefit from LinkedIn being a social network. Although LinkedIn strives towards personalized results, the algorithms are required to be scalable to meet latency requirements [62], which leads to some algorithms being competitive to, but not outperform, other models [130].

## 4 Conclusion

### 4.1 Conclusions and directions for further research

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. Both this multi-perspective view, and the new taxonomy of hybrid job recommender systems has not been discussed by previous reviews on job recommender systems.

Application-oriented challenges in JRS were already highlighted in early JRS contributions, though, still most literature does not take these into account. Contributions that do take different views on the JRS problem, however, do show that such views can have considerable benefits. These benefits may include improved model performance (temporal perspective), improved distribution of candidates over a set of homogeneous vacancies (reciprocal perspective), or ensuring algorithm fairness (ethical perspective). Currently, most attention goes out to how to represent the substantial amount of textual data from both candidate profiles and vacancies to create job recommendations, for which recently especially deep representations have shown promising results. However, this focus may also create the illusion that this is the only perspective that is relevant.

Especially in terms of fairness, such a single perspective can be considerably harmful. Although we are not aware of algorithm audits on job recommender systems, an audit on the candidate search engines of Indeed, Careerbuilder, and Monster, did show significant results for both individual and group unfairness in terms of gender. The increased scientific attention towards algorithm fairness, however, does provide algorithms and metrics that can be applied to measure and ensure algorithm fairness. Hence, there is a research opportunity to study how these can be transferred to the job recommender system domain.

Many authors state in the introduction of their contribution that there is a vast amount of data available in the form of vacancies and job seeker profiles. However, there is a clear split in the literature with regards to contributions having also access to interaction data between these two, in particular in the form of clicks/skips on the recommendation list. Interaction data can resolve the language inconsistency between job seekers and recruiters, which is especially troublesome in content-based and some knowledge-based JRS. In case interaction data is missing, one common resort is to use one of the available datasets originating from JRS competitions, in particular the CareerBuilder 2012, RecSys 2016, and RecSys 2017 competitions, which therefore have had a considerable influence on the JRS literature.

An interesting aspect with respect to the usage of these competition datasets, beyond the contributions to the competitions themselves, is that these datasets are mostly used for training, but rarely for validation. This is unfortunate, as the (to our knowledge) only contribution that compares JRS on different competition datasets shows that error metrics may differ substantially across different datasets. I.e., this raises questions with respect to the generalizability of JRS trained on one dataset. Another interesting question why (online) interaction data is sometimes not taken into account, or along the same line, why researchers often resort to the competition datasets, beyond the motives of contributing to the competition or for validation. Although there may be many valid reasons, we would like to hypothesize from anecdotal experience that it can be difficult to obtain such interaction datasets, as recruitment organizations are not always part of research communities, or given that these recruitment organizations have not always considered the implications of sharing data for research, either from a technical or legal point of view, making it difficult to use such datasets on a short term.

### 4.2 Limitations

Although we believe to have given a broad overview of contributions on job recommender systems, we do like to address some limitations of this review. First, although we managed to further split JRS hybrids into smaller categories, still some classes comprise similar methods. One particular example is that some

methods, currently classified as MM-SE, are quite similar to cascade hybrids. I.e., if a cascade hybrid would have used the semantic representation of jobs and job seekers as features in a boosting model, instead of using the similarity between the two as feature, it would have been classified as MM-SE. A similar argument holds for the feature augmentation and cascade hybrid classes.

Second, to limit the scope of this literature review, we only considered literature on job recommender systems. However, such systems do not exist in isolation: they are commonly part of an e-recruitment platform also comprising candidate recommendations, or even employee selection, which are sparsely included in this review.

## References

- [1] Fabian Abel, András Benczúr, Daniel Kohlsdorf, Martha Larson, and Róbert Pálovics. 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 Tsirhrintzis, 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 Conference on Information and Knowledge Management*, pages 1699–1702, 2015.
- [8] Jack Bandy. Problematic machine behavior: A systematic literature review of algorithm audits. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–34, 2021.
- [9] Shivam Bansal, Aman Srivastava, and Anuja Arora. Topic modeling driven content based jobs recommendation engine for recruitment industry. *Procedia computer science*, 122:865–872, 2017.
- [10] Mathieu Bastian, Matthew Hayes, William Vaughan, Sam Shah, Peter Skomoroch, Hyungjin Kim, Sal Uryasev, and Christopher Lloyd. LinkedIn skills: large-scale topic extraction and inference. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 1–8, 2014.
- [11] Zeynep Batmaz, Ali Yurekli, Alper Bilge, and Cihan Kaleli. A review on deep learning for recommender systems: challenges and remedies. *Artificial Intelligence Review*, 52(1):1–37, 2019.
- [12] Alan Mark Berg. The learning analytics architectural lifecycle. 2018.
- [13] Shuqing Bian, Wayne Xin Zhao, Yang Song, Tao Zhang, and Ji-Rong Wen. Domain adaptation for person-job fit with transferable deep global match network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4810–4820, 2019.

- [14] Shuqing Bian, Xu Chen, Wayne Xin Zhao, Kun Zhou, Yupeng Hou, Yang Song, Tao Zhang, and Ji-Rong Wen. Learning to match jobs with resumes from sparse interaction data using multi-view co-teaching network. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 65–74, 2020.
- [15] Mattia Bianchi, Federico Cesaro, Filippo Ciciri, Mattia Dagradra, Alberto Gasparin, Daniele Grattarola, Ilyas Inajjar, Alberto Maria Metelli, and Leonardo Cella. Content-based approaches for cold-start job recommendations. In *RecSys Challenge '17: Proceedings of the Recommender Systems Challenge 2017*, pages 1–5. 2017.
- [16] Christopher M. Bishop. *Pattern recognition and machine learning*. Springer, 2006.
- [17] Fedor Borisyuk, Liang Zhang, and Krishnaram Kenthapadi. LiJAR: A system for job application redistribution towards efficient career marketplace. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1397–1406, 2017.
- [18] Paul Boselie. *Strategic human resource management: A balanced approach*. McGraw-Hill Education, 2010.
- [19] Svetlin Bostandjiev, John O’Donovan, and Tobias Höllerer. LinkedVis: exploring social and semantic career recommendations. In *Proceedings of the 2013 international conference on Intelligent user interfaces*, pages 107–116, 2013.
- [20] Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
- [21] CareerBuilder. *Website CareerBuilder*, 2021. Retrieved from: <https://www.careerbuilder.com>, accessed April 26, 2021.
- [22] Tommaso Carpi, Marco Edemanti, Ervin Kamberoski, Elena Sacchi, Paolo Cremonesi, Roberto Pagano, and Massimo Quadrana. Multi-stack ensemble for job recommendation. In *RecSys Challenge '16: Proceedings of the Recommender Systems Challenge*, pages 1–4. 2016.
- [23] Le Chen, Ruijun Ma, Anikó Hannák, and Christo Wilson. Investigating the impact of gender on rank in resume search engines. In *Proceedings of the 2018 chi conference on human factors in computing systems*, pages 1–14, 2018.
- [24] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.
- [25] Weijian Chen, Xingming Zhang, Haoxiang Wang, and Hongjie Xu. Hybrid deep collaborative filtering for job recommendation. In *2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA)*, pages 275–280. IEEE, 2017.
- [26] Wenbo Chen, Pan Zhou, Shaokang Dong, Shimin Gong, Menglan Hu, Kehao Wang, and Dapeng Wu. Tree-based contextual learning for online job or candidate recommendation with big data support in professional social networks. *IEEE Access*, 6:77725–77739, 2018.
- [27] Oualid Chenni, Yanis Bouda, Hamid Benachour, and Chahnez Zakaria. A content-based recommendation approach using semantic user profile in e-recruitment. In *International Conference on Theory and Practice of Natural Computing*, pages 23–32. Springer, 2015.
- [28] Yi-Chi Chou and Han-Yen Yu. Based on the application of AI technology in resume analysis and job recommendation. In *2020 IEEE International Conference on Computational Electromagnetics (ICCEM)*, pages 291–296. IEEE, 2020.
- [29] Vachik S. Dave, Baichuan Zhang, Mohammad Al Hasan, Khalifeh AlJadda, and Mohammed Korayem. A combined representation learning approach for better job and skill recommendation. In *Proceedings*

*of the 27th ACM International Conference on Information and Knowledge Management*, pages 1997–2005, 2018.

- [30] Toon De Pessemier, Kris Vanhecke, and Luc Martens. A scalable, high-performance algorithm for hybrid job recommendations. In *RecSys Challenge '16: Proceedings of the Recommender Systems Challenge*, pages 1–4. 2016.
- [31] Corné de Ruijt and Sandjai Bhulai. A comparison of machine-learned survival models for predicting tenure from unstructured resumes. 2021.
- [32] Corné de Ruijt, Sandjai Bhulai, Han Rusman, and Leon Willemsens. Predicting candidate uptake for online vacancies. *DATA ANALYTICS 2016*, pages 63–68, 2016.
- [33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [34] Mamadou Diaby and Emmanuel Viennet. Taxonomy-based job recommender systems on Facebook and LinkedIn profiles. In *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*, pages 1–6. IEEE, 2014.
- [35] Mamadou Diaby, Emmanuel Viennet, and Tristan Launay. Toward the next generation of recruitment tools: an online social network-based job recommender system. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*, pages 821–828. IEEE, 2013.
- [36] Mamadou Diaby, Emmanuel Viennet, and Tristan Launay. Exploration of methodologies to improve job recommender systems on social networks. *Social Network Analysis and Mining*, 4(1):227, 2014.
- [37] Giacomo Domeniconi, Gianluca Moro, Andrea Pagliarani, Karin Pasini, and Roberto Pasolini. Job recommendation from semantic similarity of LinkedIn users’ skills. In *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, pages 270–277, 2016.
- [38] Shaokang Dong, Zijian Lei, Pan Zhou, Kaigui Bian, and Guanghui Liu. Job and candidate recommendation with big data support: A contextual online learning approach. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–7. IEEE, 2017.
- [39] ELISE. Website *ELISE Job Matching Search and Match Platform*, 2021. Retrieved from: <https://www.wcc-group.com/employment/products/elise-job-matching-search-and-match-platform/>, accessed April 26, 2021.
- [40] Evangelia Faliagka, 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(3):515–528, 2014.
- [41] Alexander Felfernig, Michael Jeran, Gerald Ninas, Florian Reinfrank, and Stefan Reiterer. Toward the next generation of recommender systems: applications and research challenges. In *Multimedia services in intelligent environments*, pages 81–98. Springer, 2013.
- [42] Peter Flach. *Machine learning: the art and science of algorithms that make sense of data*. Cambridge University Press, 2012.
- [43] Mauricio Noris Freire and Leandro Nunes de Castro. e-Recruitment recommender systems: a systematic review. *Knowledge and Information Systems*, pages 1–20, 2020.
- [44] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2221–2231, 2019.

- [45] Akshay Gugnani and Hemant Misra. Implicit skills extraction using document embedding and its use in job recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13286–13293, 2020.
- [46] Huan Gui, Haishan Liu, Xiangrui Meng, Anmol Bhavin, and Jiawei Han. Downside management in recommender systems. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 394–401. IEEE, 2016.
- [47] Cheng Guo, Hongyu Lu, Shaoyun Shi, Bin Hao, Bin Liu, Min Zhang, Yiqun Liu, and Shaoping Ma. How integration helps on cold-start recommendations. In *RecSys Challenge ’17: Proceedings of the Recommender Systems Challenge 2017*, pages 1–6. 2017.
- [48] Xingsheng Guo, Houssem Jerbi, and Michael P O’Mahony. An analysis framework for content-based job recommendation. In *22nd International Conference on Case-Based Reasoning (ICCBR)*, 2014.
- [49] Anika Gupta and Deepak Garg. Applying data mining techniques in job recommender system for considering candidate job preferences. In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1458–1465. IEEE, 2014.
- [50] Francisco Gutierrez, Sven Charleer, Robin De Croon, Nyi Nyi Htun, Gerd Goetschalckx, and Katrien Verbert. Explaining and exploring job recommendations: a user-driven approach for interacting with knowledge-based job recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 60–68, 2019.
- [51] Christopher J Hartwell. *The use of social media in employee selection: Prevalence, content, perceived usefulness, and influence on hiring decisions*. PhD thesis, Purdue University, 2015.
- [52] Bradford Heap, Alfred Krzywicki, Wayne Wobcke, Mike Bain, and Paul Compton. Combining career progression and profile matching in a job recommender system. In *Pacific Rim International Conference on Artificial Intelligence*, pages 396–408. Springer, 2014.
- [53] Wenxing Hong, Siting Zheng, Huan Wang, and Jianchao Shi. A job recommender system based on user clustering. *Journal of Computers (JCP)*, 8(8):1960–1967, 2013.
- [54] Jose Ignacio Honrado, Oscar Huarte, Cesar Jimenez, Sebastian Ortega, Jose R. Perez-Aguera, Joaquin Perez-Iglesias, Alvaro Polo, and Gabriel Rodriguez. Jobandtalent at RecSys challenge 2016. In *RecSys Challenge ’16: Proceedings of the Recommender Systems Challenge*, pages 1–5. 2016.
- [55] Dietmar Jannach, Gabriel de Souza P. Moreira, and Even Oldridge. Why are deep learning models not consistently winning recommender systems competitions yet? a position paper. *RecSys Challenge’20, Proceedings of the Recommender Systems Challenge 2020*, page 44–49, 2020.
- [56] Bernard J Jansen, Karen J Jansen, and Amanda Spink. Using the web to look for work: Implications for online job seeking and recruiting. *Internet research*, 15(1):49–66, 2005.
- [57] Andrzej Janusz, Sebastian Stawicki, Michał Drewniak, Krzysztof Ciebiera, Dominik Ślęzak, and Krzysztof Stencel. How to match jobs and candidates - a recruitment support system based on feature engineering and advanced analytics. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 503–514. Springer, 2018.
- [58] Junshu Jiang, Songyun Ye, Wei Wang, Jingran Xu, and Xiaosheng Luo. Learning effective representations for person-job fit by feature fusion. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2549–2556, 2020.
- [59] Miao Jiang, Yi Fang, Huangming Xie, Jike Chong, and Meng Meng. User click prediction for personalized job recommendation. *World Wide Web*, 22(1):325–345, 2019.
- [60] Kaggle. *Careerbuilder Job Recommendation Challenge*, 2012. Retrieved from: <https://www.kaggle.com/c/job-recommendation>, accessed February 15, 2021.

- [61] Kaggle. *Kaggle website*, 2021. Retrieved from: <https://www.kaggle.com>, accessed February 15, 2021.
- [62] Krishnaram Kenthapadi, Benjamin Le, and Ganesh Venkataraman. Personalized job recommendation system at linkedin: Practical challenges and lessons learned. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 346–347, 2017.
- [63] Rémy Kessler, 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(6):1124–1135, 2012.
- [64] Aseel B. Kmail, Mohammed Maree, and Mohammed Belkhatir. MatchingSem: Online recruitment system based on multiple semantic resources. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 2654–2659. IEEE, 2015.
- [65] Aseel B. Kmail, Mohammed Maree, Mohammed Belkhatir, and Saadat M. Alhashmi. An automatic online recruitment system based on exploiting multiple semantic resources and concept-relatedness measures. In *27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 620–627. IEEE, 2015.
- [66] Tanja Koch, Charlene Gerber, and Jeremias J. De Clerk. The impact of social media on recruitment: Are you linkedin? *SA Journal of Human Resource Management*, 16(1), 2018.
- [67] Emanuel Lacic, Markus Reiter-Haas, Tomislav Duricic, Valentin Slawicek, and Elisabeth Lex. Should we embed? A study on the online performance of utilizing embeddings for real-time job recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 496–500, 2019.
- [68] Emanuel Lacic, Markus Reiter-Haas, Dominik Kowald, Manoj Reddy Dareddy, Junghoo Cho, and Elisabeth Lex. Using autoencoders for session-based job recommendations. *User Modeling and User-Adapted Interaction*, 30(4):617–658, 2020.
- [69] Sven Laumer, Fabian Gubler, Christian Maier, and Tim Weitzel. Job seekers’ acceptance of job recommender systems: Results of an empirical study. In *Proceedings of the 51st Hawaii International Conference on System Sciences*, pages 3914–3923, 2018.
- [70] Ran Le, Wenpeng Hu, Yang Song, Tao Zhang, Dongyan Zhao, and Rui Yan. Towards effective and interpretable person-job fitting. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1883–1892, 2019.
- [71] Yeon-Chang Lee, Jiwon Hong, and Sang-Wook Kim. Job recommendation in askstory: experiences, methods, and evaluation. In *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pages 780–786, 2016.
- [72] Yujin Lee, Yeon-Chang Lee, Jiwon Hong, and Sang-Wook Kim. Exploiting job transition patterns for effective job recommendation. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2414–2419. IEEE, 2017.
- [73] Vasily Leksin and Andrey Ostapets. Job recommendation based on factorization machine and topic modelling. In *RecSys Challenge ’16: Proceedings of the Recommender Systems Challenge*, pages 1–4. 2016.
- [74] Vasily Leksin, Andrey Ostapets, Mikhail Kamenshikov, Dmitry Khodakov, and Vasily Rubtsov. Combination of content-based user profiling and local collective embeddings for job recommendation. In *CEUR Workshop Proceeding, Vol. 1968, No. Experimental Economics and Machine Learning*, pages 9–17, 2017.
- [75] Jia Li, Dhruv Arya, Viet Ha-Thuc, and Shakti Sinha. How to get them a dream job? entity-aware features for personalized job search ranking. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 501–510, 2016.

- [76] Jianxun Lian, Fuzheng Zhang, Min Hou, Hongwei Wang, Xing Xie, and Guangzhong Sun. Practical lessons for job recommendations in the cold-start scenario. In *RecSys Challenge '17: Proceedings of the Recommender Systems Challenge 2017*, pages 1–6. 2017.
- [77] Yiou Lin, Hang Lei, Prince Clement Addo, and Xiaoyu Li. Machine learned resume-job matching solution. *arXiv preprint arXiv:1607.07657*, 2016.
- [78] LinkedIn. *About LinkedIn*, 2021. Retrieved from: <https://about.linkedin.com>, accessed March 23, 2021.
- [79] Kuan Liu, Xing Shi, Anoop Kumar, Linhong Zhu, and Prem Natarajan. Temporal learning and sequence modeling for a job recommender system. In *RecSys Challenge '16: Proceedings of the Recommender Systems Challenge*, pages 1–4. 2016.
- [80] Miao Liu, Zijie Zeng, Weike Pan, Xiaogang Peng, Zhiguang Shan, and Zhong Ming. Hybrid one-class collaborative filtering for job recommendation. In *International Conference on Smart Computing and Communication*, pages 267–276. Springer, 2016.
- [81] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. Recommender system application developments: a survey. *Decision Support Systems*, 74:12–32, 2015.
- [82] Yao Lu, Sandy El Helou, and Denis Gillet. A recommender system for job seeking and recruiting website. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 963–966, 2013.
- [83] Malte Ludewig, Michael Jugovac, and Dietmar Jannach. A light-weight approach to recipient determination when recommending new items. In *RecSys Challenge '17: Proceedings of the Recommender Systems Challenge 2017*, pages 1–6. 2017.
- [84] Yong Luo, Huaizheng Zhang, Yonggang Wen, and Xinwen Zhang. Resumegan: An optimized deep representation learning framework for talent-job fit via adversarial learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1101–1110, 2019.
- [85] Saket Maheshwary and Hemant Misra. Matching resumes to jobs via deep siamese network. In *Companion Proceedings of the The Web Conference 2018*, pages 87–88, 2018.
- [86] Emmanuel Malherbe, 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)*, pages 588–595. IEEE, 2014.
- [87] Jochen Malinowski, Tobias Keim, Oliver Wendt, and Tim Weitzel. Matching people and jobs: A bilateral recommendation approach. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 6, page 137.3. IEEE, 2006.
- [88] Jorge Martinez-Gil, Alejandra Lorena Paoletti, and Mario Pichler. A novel approach for learning how to automatically match job offers and candidate profiles. *Information Systems Frontiers*, pages 1–10, 2019.
- [89] Sonu K Mishra and Manoj Reddy. A bottom-up approach to job recommendation system. In *RecSys Challenge '16: Proceedings of the Recommender Systems Challenge*, pages 1–4. 2016.
- [90] Motebang Daniel Mpela and Tranos Zuva. A mobile proximity job employment recommender system. In *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–6. IEEE, 2020.
- [91] Amber Nigam, Aakash Roy, Hartaran Singh, and Harsimran Waila. Job recommendation through progression of job selection. In *2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pages 212–216. IEEE, 2019.

- [92] Cathy O’Neil. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown, 2016.
- [93] O\*Net. *O\*Net*, 2021. Retrieved from: <https://www.onetonline.org>, accessed March 27, 2021.
- [94] Andrzej Pacuk, Piotr Sankowski, Karol Węgrzycki, Adam Witkowski, and Piotr Wygocki. RecSys challenge 2016: job recommendations based on preselection of offers and gradient boosting. In *RecSys Challenge ’16: Proceedings of the Recommender Systems Challenge*, pages 1–4. 2016.
- [95] Ioannis Paparrizos, B. Barla Cambazoglu, and Aristides Gionis. Machine learned job recommendation. In *Proceedings of the fifth ACM Conference on Recommender Systems*, pages 325–328, 2011.
- [96] Caroline Criado Perez. *Invisible women: Exposing data bias in a world designed for men*. Random House, 2019.
- [97] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (long papers)*, pages 2227–2237, 2018.
- [98] Marc Poch, Núria Bel Rafecas, Sergio Espeja, and Felipe Navio. Ranking job offers for candidates: learning hidden knowledge from big data. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. ACL (Association for Computational Linguistics), 2014.
- [99] Mirko Polato and Fabio Aiolfi. A preliminary study on a recommender system for the job recommendation challenge. In *RecSys Challenge ’16: Proceedings of the Recommender Systems Challenge*, pages 1–4. 2016.
- [100] Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Liang Jiang, Enhong Chen, and Hui Xiong. Enhancing person-job fit for talent recruitment: An ability-aware neural network approach. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 25–34, 2018.
- [101] Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Chao Ma, Enhong Chen, and Hui Xiong. An enhanced neural network approach to person-job fit in talent recruitment. *ACM Transactions on Information Systems (TOIS)*, 38(2):1–33, 2020.
- [102] Michael Reusens, Wilfried Lemahieu, Bart Baesens, and Luc Sels. A note on explicit versus implicit information for job recommendation. *Decision Support Systems*, 98:26–35, 2017.
- [103] Michael Reusens, Wilfried Lemahieu, Bart Baesens, and Luc Sels. Evaluating recommendation and search in the labor market. *Knowledge-Based Systems*, 152:62–69, 2018.
- [104] Alberto Rivas, Pablo Chamoso, Alfonso González-Briones, Roberto Casado-Vara, and Juan Manuel Corchado. Hybrid job offer recommender system in a social network. *Expert Systems*, 36(4):e12416, 2019.
- [105] Masahiro Sato, Koki Nagatani, and Takuji Tahara. Exploring an optimal online model for new job recommendation: Solution for RecSys challenge 2017. In *RecSys Challenge ’17: Proceedings of the Recommender Systems Challenge 2017*, pages 1–5. 2017.
- [106] Thomas Schmitt, Philippe Caillou, and Michèle Sebag. Matching jobs and resumes: a deep collaborative filtering task. In *GCAI 2016. 2nd Global Conference on Artificial Intelligence*, volume 41, 2016.
- [107] Thomas Schmitt, François Gonard, Philippe Caillou, and Michèle Sebag. Language modelling for collaborative filtering: Application to job applicant matching. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1226–1233. IEEE, 2017.

- [108] Walid Shalaby, BahaaEddin AlAila, Mohammed Korayem, Layla Pournajaf, Khalifeh AlJadda, Shannon Quinn, and Wlodek Zadrozny. Help me find a job: A graph-based approach for job recommendation at scale. In *2017 IEEE international conference on big data (big data)*, pages 1544–1553. IEEE, 2017.
- [109] Saman Shishehchi and Seyed Yashar Banihashem. Jrdp: a job recommender system based on ontology for disabled people. *International Journal of Technology and Human Interaction (IJTHI)*, 15(1):85–99, 2019.
- [110] Zheng Siting, Hong Wenxing, Zhang Ning, and Yang Fan. Job recommender systems: a survey. In *2012 7th International Conference on Computer Science & Education (ICCSE)*, pages 920–924. IEEE, 2012.
- [111] Stefan Strohmeier and Franca Piazza. Domain driven data mining in human resource management: A review of current research. *Expert Systems with Applications*, 40(7):2410–2420, 2013.
- [112] Studio. *Website Studio*, 2021. Retrieved from: <https://stud'o.com/at>, accessed March 2, 2021. Redirected from <https://stud'o.co>.
- [113] textkernel. *Website Textkernel*, 2021. Retrieved from: <https://www.textkernel.com>, accessed March 15, 2021.
- [114] Minh-Luan Tran, Anh-Tuyen Nguyen, Quoc-Dung Nguyen, and Tin Huynh. A comparison study for job recommendation. In *2017 International Conference on Information and Communications (ICIC)*, pages 199–204. IEEE, 2017.
- [115] Jorge Carlos Valverde-Rebaza, Ricardo Puma, Paul Bustios, and Nathalia C. Silva. Job recommendation based on job seeker skills: An empirical study. In *Text2Story@ ECIR*, pages 47–51, 2018.
- [116] José Vega. Semantic matching between job offers and job search requests. In *COLING 1990 Volume 1: Papers presented to the 13th International Conference on Computational Linguistics*, 1990.
- [117] Maksims Volkovs, Guang Wei Yu, and Tomi Poutanen. Content-based neighbor models for cold start in recommender systems. In *RecSys Challenge '17: Proceedings of the Recommender Systems Challenge 2017*, pages 1–6. 2017.
- [118] Jian Wang, Yi Zhang, Christian Posse, and Anmol Bhasin. Is it time for a career switch? In *Proceedings of the 22nd international conference on World Wide Web*, pages 1377–1388, 2013.
- [119] Pengyang Wang, Yingtong Dou, and Yang Xin. The analysis and design of the job recommendation model based on GBRT and time factors. In *2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA)*, pages 29–35. IEEE, 2016.
- [120] Wenming Xiao, Xiao Xu, Kang Liang, Junkang Mao, and Jun Wang. Job recommendation with hawkes process: an effective solution for RecSys challenge 2016. In *Proceedings of the recommender systems challenge*, pages 1–4. 2016.
- [121] Xing. *Website Xing*, 2021. Retrieved from: <https://www.xing.com>, accessed March 27, 2021.
- [122] Peng Xu and Denilson Barbosa. Matching résumés to job descriptions with stacked models. In *Canadian Conference on Artificial Intelligence*, pages 304–309. Springer, 2018.
- [123] Ya Xu, Nanyu Chen, Addrian Fernandez, Omar Sinno, and Anmol Bhasin. From infrastructure to culture: A/b testing challenges in large scale social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2227–2236, 2015.
- [124] Murat Yagci and Fikret Gurgen. A ranker ensemble for multi-objective job recommendation in an item cold start setting. In *Proceedings of the Recommender Systems Challenge*, pages 1–4. 2017.
- [125] Rui Yan, Ran Le, Yang Song, Tao Zhang, Xiangliang Zhang, and Dongyan Zhao. Interview choice reveals your preference on the market: To improve job-resume matching through profiling memories.

In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 914–922, 2019.

- [126] Shuo Yang, Mohammed Korayem, Khalifeh AlJadda, Trey Grainger, and Sriraam Natarajan. Combining content-based and collaborative filtering for job recommendation system: A cost-sensitive statistical relational learning approach. *Knowledge-Based Systems*, 136:37–45, 2017.
- [127] Abeer Zaroor, Mohammed Maree, and Muath Sabha. A hybrid approach to conceptual classification and ranking of resumes and their corresponding job posts. In *International Conference on Intelligent Decision Technologies*, pages 107–119. Springer, 2017.
- [128] Abeer Zaroor, Mohammed Maree, and Muath Sabha. Jrc: a job post and resume classification system for online recruitment. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 780–787. IEEE, 2017.
- [129] Chenrui Zhang and Xueqi Cheng. An ensemble method for job recommender systems. In *RecSys Challenge '16: Proceedings of the Recommender Systems Challenge*, pages 1–4. 2016.
- [130] XianXing Zhang, Yitong Zhou, Yiming Ma, Bee-Chung Chen, Liang Zhang, and Deepak Agarwal. Glmix: Generalized linear mixed models for large-scale response prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 363–372, 2016.
- [131] Alice Zheng and Amanda Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc., 2018.
- [132] Chen Zhu, Hengshu Zhu, Hui Xiong, Chao Ma, Fang Xie, Pengliang Ding, and Pan Li. Person-job fit: Adapting the right talent for the right job with joint representation learning. *ACM Transactions on Management Information Systems (TMIS)*, 9(3):1–17, 2018.
- [133] Dávid Zibriczky. A combination of simple models by forward predictor selection for job recommendation. In *RecSys Challenge '16: Proceedings of the Recommender Systems Challenge*, pages 1–4. 2016.