An Exploration of Sentence-Pair Classification for Algorithmic Recruiting

Mesut Kaya
mkaya@ikp.aau.dk
Aalborg University Copenhagen
Copenhagen, Denmark

Toine Bogers
tobo@itu.dk
IT University of Copenhagen
Copenhagen, Denmark

ABSTRACT

Recent years have seen a rapid increase in the application of computational approaches to different HR tasks, such as algorithmic hiring, skill extraction, and monitoring of employee satisfaction. Much of the recent work on estimating the fit between a person and a job has used representation learning to represent both resumes and job vacancies computationally and determine the degree to which they match. A common approach to this task is Sentence-BERT, which uses a Siamese network to encode resumes and job descriptions into fixed-length vectors and estimates how well they match based on the similarity between those vectors. In our paper, we adapt BERT's next-sentence prediction task—predicting whether one sentence is likely to follow another in a given context—to the task of matching resumes with job descriptions. Using historical data on past (mis)matches between job-resume pairs, we fine-tune BERT for this downstream task. Through a combination of offline and online experiments on data from a large Scandinavian job portal, we show that this approach performs significantly better than Sentence-BERT and other state-of-the-art approaches for determining person-job fit.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Job recommendation, algorithmic recruiting, computational HR, person-job fit, algorithmic hiring

ACM Reference Format:
Mesut Kaya and Toine Bogers. 2018. An Exploration of Sentence-Pair Classification for Algorithmic Recruiting. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Recent years have seen a rapid increase in the popularity of online recruitment platforms, such as LinkedIn, Indeed, Glassdoor, and Seek. These platforms have made it increasingly easier for companies and job seekers to make job postings and resumes available online and match them algorithmically [3, 8, 14]. On the LinkedIn platform alone, 117 job applications were submitted per second and 8 people were hired every minute from its 930M+ members and 63M+ companies in July 2023[1]. However, the availability of a larger number positions and candidate resumes also places a greater assessment burden on job seekers and recruiters [14]. As a result, there has been a commensurate increase in the popularity of algorithmic hiring approaches to support job seekers and recruiters [7, 9, 12, 15, 19]. This line of research is aimed at matching job seekers with relevant jobs by assessing the match between a job's requirements and the candidates' knowledge, skills, and abilities, education level, work experience and interests [4]. A successful matching algorithm can then be used to either recommend relevant jobs to job seekers or to support recruiters by identifying relevant candidates for a specific job.

With the rising popularity of deep learning approaches and large language models, much of the recent research on person-job fit algorithms has focused on generating representations of both the job seekers' resumes and and the job postings using different representation learning techniques [7, 15, 16, 18, 19]. Some of these approaches require resumes and job postings to be available in a structured format. For instance, job postings and resumes should consist of minimum number of clearly separated requirements and work experiences respectively. However, this assumption can be problematic given the abundance of noisy, heterogeneous and (at most) semi-structured data in the HR domain.

To deal with such heterogenous and semi-structured data, a recently proposed approach by Reimers and Gurevych [17] was based on Sentence-BERT (SBERT), a modification of the BERT network that uses Siamese networks to derive semantically meaningful sentence embeddings. SBERT was originally used for the task of pairwise semantic similarity [17], but it has recently been adapted to the person-job fit scenario in matching two heterogeneous types of data, and was shown to work better than approaches that creates representations of both resumes and job postings using BERT or TF·IDF [12]. Although both resumes and job postings are documents consisting of multiple sentences, Lavi et al. [12] showed that the best approach is to take the first N tokens of both resumes and job postings, treat them as sentences, and then create separate representations for them to predict the final matching probability (as shown in Figure 1a) [12]. However, creating separate representations on heterogenous data could result in sub-optimal performance due to a language gap between job postings and resumes. A more fruitful approach to closing this language gap might instead be to jointly encode pairs of job postings and resumes into single, unified embeddings, as visualized in Figure 1b.

[1] https://news.linkedin.com/about-us#Statistics, last accessed March 26, 2024

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

We explore this approach in this paper by adapting BERT's next-sentence prediction task to the person-job fit scenario. In the next-sentence prediction task (also known as sentence-pair classification) a model is trained to predict whether a given sentence follows another in a given context. We formulate the person-job fit problem as a next-sentence prediction task where the goal is to predict, given a job posting and resume pair, whether a resume follows the job posting. We refer to our model as *SPBERT* (or Sentence-Pair BERT). This approach is analogous to concatenating a job posting and a resume into a single document and fine-tuning a base BERT model to learn the semantic ordering of the tokens being used. Using offline experiments, we show that SPBERT outperforms not only SBERT but also other state-of-the-art person-job fit algorithms. Finally, we also show the effectiveness of SPBERT using online experiments.

2 RELATED WORK

Like in many other recommendation domains, the main types of person-job fit algorithms fall under content-based filtering, collaborative filtering, or hybrid approaches [1]. Due to the textual nature of both job postings and resumes, most approaches have at least a content-based component where job postings and resumes are embedded in a latent space. These resume and job posting embeddings are then used in various ways to estimate the degree to which they match [7, 9, 11, 15, 16, 19, 21]. Some approaches vary in the type of textual and social information that is embedded in the latent space, while other approaches differ in the embedding techniques used, such as Word2vec and Doc2vec to create representations [9, 11], deep learning approaches using CNNs, RNNs and LSTMs [7, 15, 21] or graph representation learning [19].

A hybrid model called JTIH (which stands for Job Title-based recommendation with Item History) was proposed by Kaya and Bogers [9] and uses only job titles as the textual representation of both job posting and resumes. All job titles are embedded into a latent space using a Word2vec model pre-trained on a large dataset of job postings and resumes. A new job posting is embedded directly using this model, while a resume embedding is created by using historical interaction data: each job seeker is represented by the mean of all job title embeddings for all the jobs they have expressed a positive interest in in the past. Figure 1c visualizes the JTIH architecture. While JTIH is a simple yet effective approach, a potential problem is that by using only job title information, the algorithm may miss crucial information in assessing person-job fit. It also disregards a job seekers actual CV in favor of past positions they have liked but not necessarily held. Another approach called PJFNN (Person-Job Fit Neural Network) by Zhu et al. [21] uses a hierarchical Convolutional Neural Network (CNN) to encode resumes and job postings independently into embeddings based on historical job application data, along with requirements from jobs and work experience from resumes. PJFNN, shown in Figure 1d, then computes the matching degree between resume and job posting by using their embeddings.

BPJFNN (or Basic Person-Job Fit Neural Network) leverages bidirectional LSTMs (BiLSTM) to derive the resume and job description representations of job requirements and resume work experiences [15], as shown in Figure 1e. BPJFNN is based on APJFNN [15], and treats all requirements in one job posting as a long sentence instead of separate requirements. Similary, all work experiences in a resume are considered as a single long sentence. Unlike BPJFNN, APJFNN is fed with separate requirements for jobs and work experiences for resumes.

PJFNN, BPJFNN and APJFNN all assume that all job requirements and work experiences are extracted from the job postings and resumes respectively prior to recommendation and that they are represented in a structured format. They also require a minimum number of requirements and work experiences to be available for each job posting and resume, which means it works less well on people new to the labor market. While these algorithms may provide competitive performance under such controlled conditions, they lack generalizability to situations where such structured data is not available or in the required amounts.

Finally, SBERT has also been applied successfully to the problem of person-job fit by Lavi et al. [12]. Sentence-BERT (SBERT) is commonly used to determine the degree of semantic similarity between pair of input sentences, in this case job postings and resumes. As shown in Figure 1a, each job posting and resume is independently embedded using BERT, after which the model is optimized by looking at the similarity of the two representations [12, 17].

Sentence-pair BERT (SPBERT) is a variant of SBERT and creates a single representation for a pair of input sentences. We expect that such a sentence-pair classification task would be more suitable for the task of matching two heteregoneous types of data that could potentially suffer from the vocabulary problem, i.e., resumes and job postings. To the best of our knowledge, the only work that has SPBERT in the job domain was to predict whether two job experiences belong to one job seeker or not [2]. While they use textual job description data of two different job postings as their sentence pairs, in our work we instead use job posting and resume pairs to assess whether they match.

3 SENTENCE-PAIR CLASSIFICATION FOR PERSON-JOB FIT

3.1 Problem Formulation

In this paper, we focus on Person-Job Fit problem or job recommendation task, which measures the matching degree between a given job posting j and a resume r, to find suitable job seekers for a given job posting. We define the dataset for person-job fit as $\mathcal{D} = \{\langle r, j, y_{r,j}\rangle | r \in \mathcal{R}, j \in \mathcal{J}\}$, where $y_{r,j}$ is the binary label indicating the matching result (1 for match and 0 for not match) for resume r and job j, $\mathcal{R}$ and $\mathcal{J}$ are the set of resumes and jobs respectively. Based on the observed set $\mathcal{D}$ the task is to learn a predictive function $f(r, j) \in [0, 1]$ to predict the confidence score that a resume $r \in \mathcal{R}$ is relevant for job posting $j \in \mathcal{J}$. For the binary classification tasks that we use, for all available resumes $r \in \mathcal{R}$ we compute a confidence score $f(r, j)$ that a given resume r is relevant for a job j. For the top-N recommendation task, we sort the confidence scores for all candidate resumes and recommend top-N resumes with highest scores for j.

3.2 Model architecture

As shown in Figure 2, SPBERT uses BERT's off-the-shelf sentence-pair classification architecture [5]. To do so, SPBERT takes a concatenated job-resume pair (j, r) for tokenization. BERT's tokenizer

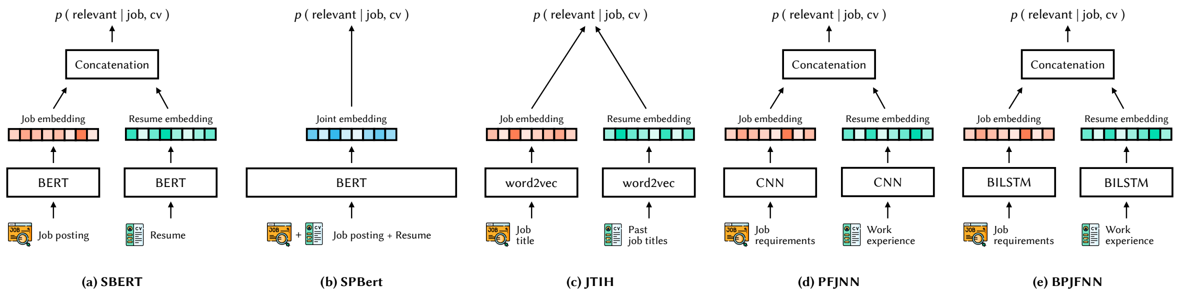

Figure 1: Visual representations of the five approaches compared in this paper.

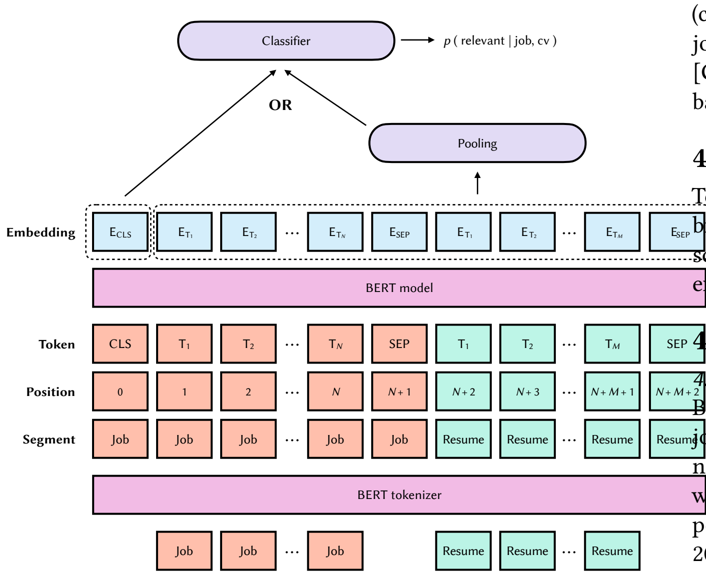

Figure 2: Overview of the SPBERT architecture.

outputs (a maximum of) 512 tokens—divided equally over the job posting and resume—from the (j, r) pair, which includes special [CLS] and [SEP] tokens that respectively indicate the sentence-pair classification task and separate the job and resume tokens. Then, those tokens are embedded into embeddings. To further separate the job j from resume r, token segments [JOB] and [CV] are added to the token embeddings. To capture the order of the tokens, position embeddings are added. The tokens go through several layers of transformers ('BERT model' in Figure 2). Finally, the output embedding of the [CLS] token *or* a pooling of the rest of the tokens' embeddings is used as a representation for the (j, r) pair. A multilayer perceptron (MLP) is then used to predict the degree to which j and r match. SPBERT is initialized with a pre-trained BERT model to leverage the pre-trained language model, while the last MLP layer is learned from scratch. During training, the entire BERT model in SPBERT is fine-tuned to learn (j, r) pair representations.

We fine-tune SPBERT with a softmax classifier objective function for one epoch by optimizing the binary cross-entropy classification loss. We used a batch size of 32, Adam optimizer with learning rate 2E-5. Note that we fine-tune our model using only the training set (cf. see Section 4.1.1). We used a maximum total of 512 tokens from job posting and resume text combined (including special tokens [CLS] and [SEP]) to fine-tune the pre-trained BERT model, as the base model limits the maximum number of tokens to 512.

4 EXPERIMENTS

To evaluate SPBERT's performance against baselines, we use combine offline and online evaluation [13]. In Section 4.1, we first describe the dataset used and baselines used as part of the offline experiments, followed by the online experiments in Section 4.2.

4.1 Offline Experiment

4.1.1 Dataset. The data used in the experimental validation of SPBERT for person-job fit estimation was provided by a Scandinavian job portal called NordicJobs. To protect the privacy of candidates, no personal information of job seekers was included in ths real-world dataset[2]. We used historical response data for 138,000 job postings with 120,941 resume from a four-year period from April 2018 until May 2022. After NordicJobs recruiters have shortlisted job seekers as relevant candidates for a job, they contact them by sending them a message. On average, 17.6 job seekers are contacted per job posting with a total of 2,427,288 data points (= job-resume pairs). Since all of the contacted candidates are assessed as relevant by recruiters—regardless of applying for the job or not—we treat them as positive data samples (as we have no data on whether or not they actually apply for or get the job). To create negative samples, we randomly sample a resume from the pool of available resumes for each of 2,427,288 job-resume pairs. We leave experimentation with different negative sampling strategies [6] for future work.

For the offline experiments, we further split our dataset into train, validation and test sets by looking at the timestamp of the job postings, i.e., the most recent 5% of the job postings are used as our test set, the 5% of job postings preceding our test set serves as our validation set and the remaining 90% is used as our training set. Each job posting contains the job title, an appetizer text that summarizes the job posting, a list of job categories and a list of of location IDs corresponding to where the company is located. Each resume contains a list of past and/or desired future job titles, a list of education history, a list of previous work experience, a list of keywords representing skills and occupations, a list of job categories, a list of location IDs detailing where the job seekers

[2]Due to privacy issues, we will not be able to make the datasets used publicly available, though we are planning to make our code available.

would be willing to commute to. Apart from JTIH, which only uses job title information, we use the same textual data source from job postings and resumes to feed to the models. Namely, for a resume r we take the most recent work experience, the most recent education completed, a list of job titles and keywords, which are then concatenated to create a single resume document. For job posting j, we take job title and appetizer text and concatenate both to create a single job posting document.

4.1.2 Methodology. We use the following state-of-the-art baselines in our offline experiments to compare to SPBERT (as shown in Figure 1): (a) SBERT [12], (c) JTIH [9], (d) PJFNN [21], and (e) BPJFNN [15]. As explained earlier in Section 2 section, JTIH is a simple approach that only encodes job postings and resumes using their job title information.

Similar to SPBERT (see Section 3.2), for SBERT, PJFNN and BPJFNN we also use a softmax classifier objective function for varying epochs with number of epochs tuned using the validation set by optimizing BCE loss. We used a batch size of 32 and the Adam optimizer with learning rate 2E-5. Again, all models' parameters were tuned using the trained models' performances on validation sets. We used the Recbole [20] implementation of PJFNN and BPJFNN and implemented all other algorithms ourselves using PyTorch.

In order to fairly compare SPBERT to the baselines that are being optimized using classification objectives, we use AUC, a standard evaluation metric for classification problems [12]. Our goal in the NordicJobs scenario of supporting recruiters is to rank relevant candidates—in other words: a top-N recommendation task. We therefore also need to compare the different algorithms on such a ranking task. We choose NDCG@20 as our main metric for this task. We set the cut-off at 20 for a reason: recruiters are instructed to contact between 15-20 relevant candidates per job posting and indeed, on average, they contact 17.6 candidates in our data set. For this task, following Kaya and Bogers [9], we employ pre-filtering the candidate set of resumes by applying location and category filters to all models.[3]

4.1.3 Results. The reported values in Table 1 are based on performance on test set. It can be seen that our adaptation of SPBERT outperforms all baseline algorithms. SBERT, BPJFNN and PJFNN do not perform well on either classification or ranking compared to SPBERT and with JTIH (for ranking). The relatively poor performance of PJFNN and BPJFNN could be explained by the strong requirements these algorithms have for their input data: both job postings and resumes have to be formatted in a structured way with minimum number of clearly separated job requirements and work experiences. Even though SBERT was shown to work better than other approaches that create resume and job posting representations using BERT [12], our offline experiments show that it is not the optimal approach to person-job fit estimation when matching two heterogeneous types of data. Perhaps surprisingly, although JTIH is not using deep neural networks or large language models, it significantly outperforms SBERT, PJFNN and BPJFNN. This lends further evidence to the claim by Kaya and Bogers [9] that a simple baseline that uses only job title information from resumes and job postings can be competitive. Our method achieves the best performance and improves over the best baseline JTIH by 87.9%. These results shows the value of using BERT's sentence pair classification task for person-job fit or job recommendation tasks.

[3]We do not show the results for the experiments without applying pre-filtering since for all cases applying pre-filtering always improves the performance of the recommendation models.

Table 1: Results of the offline classification and ranking experiments. The difference between SPBERT and second best performing baseline is statistically significant for CFR (p-value < 2.2e-16) according to an independent-samples t-test. We do not report AUC value for JTIH, since, unlike other approaches, it is not trained with a classification objective task.

Algorithm	AUC	NDCG@20
JTIH	n/a	0.099
PJFNN	0.620	0.011
BPJFNN	0.695	0.009
SBERT	0.703	0.017
SPBERT	**0.946**	**0.186**

4.2 Online Experiment

4.2.1 Methodology. To validate the results from our offline experiment, we conduct an online field experiment, as it is the most reliable way to measure the results from our experiments on real recruiters [10]. Because of the way candidate recommendations have to be integrated into NordicJobs's systems and the availability of recruiters, we were limited to running two algorithms against each other. Based on the results of offline experiments, we picked the best-performing baseline (JTIH) and compared it against SPBERT. After selecting an open job posting, recruiters were presented with a list of candidate recommendations from one of these two algorithms, which was randomly assigned. After assessing this list and possibly shortlisting candidates from it, recruiters could then conduct their own searches in NordicJobs's resume database. We ran our online field experiment for a 25-day period from March 6-31, 2023. During this period, 1,334 different job postings were completed by 38 different recruiters who took part in the experiment. For 676 of those jobs, recruiters were shown recommendations from JTIH and the remaining 658 were shown recommendations from SPBERT. In order to evaluate the performances of SPBERT and JTIH we use two different evaluation metrics: (1) the **P**ositive **R**esponse **R**ate (PRR) is the share of contacted candidates that responded positively to being contacted; and (2) the share of **C**ontacted job seekers that originated **F**rom the **R**ecommendations (CFR) as opposed to from their own searches. CFR is a proxy for how much work the recommendation algorithm can take over from the recruiter.

4.2.2 Results. The results of the online experiments are inline with the offline experiment results. Both shows the superiority of the SPBERT over the best performing baseline, JTIH. Due to its success, SPBERT has been deployed in NordicJobs's production environment and is actively being used to support the recruiters in their daily recruiting workflow.

Table 2: Online experiment results. The difference between SPBERT and JTIH is statistically significant for CFR (p-value < 2.2e-16), but not for PRR (p-value = 0.2093). Two sample t-test is used.

Algorithm	PRR	CFR
JTIH	12.7	24.75
SPBERT	**13.5**	**41.1**
% increase	+10.44%	+66.06%

5 DISCUSSION & CONCLUSIONS

In this paper, we have presented the results of findings of our initial investigation of adapting BERT's sentence-pair classification to the problem of estimating person-job fit. We evaluated the performance of our proposed approach SPBERT through both offline and online experiments on data from a NordicJobs, a large Scandinavian job portal, which demonstrated the superiority of SPBERT across different metrics over competitive baselines. We believe that SPBERT's superior performance stems from its ability to better address the vocabulary gap between job postings and resumes by fine-tuning the base BERT model on concatenated job-resume pairs. By generating joint single representations for these pairs, SPBERT is better able to estimate person-job fit than fine-tuned SBERT-like approaches, which fine-tune job posting and resumes separately.

Despite the promising results, we are aware that our study represents only a first step in investigating the suitability of SPBERT for the person-job fit problem. We only included a small number of key elements of job postings and resumes in our representations, and future work should study the contributions of additional elements to person-job fit estimation. In addition, we wish to perform a more detailed error analysis on collaboration with recruiters to identify the specific cases where SPBERT does and does not perform well. In future work, we would also like to compare SPBERT to additional state-of-the-art approaches, and apply it to multiple datasets. Finally, improving the scalability of SPBERT is another point of interest: SBERT embeddings can be pre-computed, but due to its use of joint job-resume representations, this is more problematic and time-consuming for SPBERT.

REFERENCES

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

[2] Vedant Bhatia, Prateek Rawat, Ajit Kumar, and Rajiv Ratn Shah. 2019. End-to-end resume parsing and finding candidates for a job description using bert. *arXiv preprint arXiv:1910.03089* (2019).

[3] Tanya Bondarouk and Chris Brewster. 2016. Conceptualising the Future of HRM and Technology Research. *The International Journal of Human Resource Management* 27, 21 (2016), 2652–2671.

[4] James A. Breaugh. 2008. Employee Recruitment: Current Knowledge and Important Areas for Future Research. *Human Resource Management Review* 18, 3 (2008), 103–118.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. *Advances in Neural Information Processing Systems* 33 (2020), 1094–1105.

[7] Junshu Jiang, Songyun Ye, Wei Wang, Jingran Xu, and Xiaosheng Luo. 2020. Learning effective representations for person-job fit by feature fusion. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2549–2556.

[8] Ugur Karaboga and Pelin Vardarlier. 2020. Examining the Use of Artificial Intelligence in Recruitment Processes. *Bussecon Review of Social Sciences (2687-2285)* 2, 44 (Dec 2020), 1–17. https://doi.org/10.36096/brss.v2i4.234

[9] Mesut Kaya and Toine Bogers. 2021. Effectiveness of job title-based embeddings on résumé-to-job-ad recommendation. In *Proceedings of the RecSys in HR 2021 workshop*. 35–41.

[10] Ron Kohavi and Roger Longbotham. 2015. Online Controlled Experiments and A/B Tests. *Encyclopedia of Machine Learning and Data Mining* (2015), 1–11.

[11] Emanuel Lacic, Markus Reiter-Haas, Tomislav Duricic, Valentin Slawicek, and Elisabeth Lex. 2019. 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*. 496–500.

[12] Dor Lavi, Volodymyr Medentsiy, and David Graus. 2021. conSultantBERT: Fine-tuned Siamese Sentence-BERT for Matching Jobs and Job Seekers. In *Proceedings of the RecSys in HR 2021 workshop*.

[13] Adrien Mogenet, Tuan Anh Nguyen Pham, Masahiro Kazama, and Jialin Kong. 2019. Predicting online performance of job recommender systems with offline evaluation. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 477–480.

[14] Paolo Montuschi, Valentina Gatteschi, Fabrizio Lamberti, Andrea Sanna, and Claudio Demartini. 2014. Job Recruitment and Job Seeking Processes: How Technology Can Help. *IT Professional* 16, 5 (Sep 2014), 41–49. https://doi.org/10.1109/MITP.2013.62

[15] Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Liang Jiang, Enhong Chen, and Hui Xiong. 2018. 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*. 25–34.

[16] Rohan Ramanath, Hakan Inan, Gungor Polatkan, Bo Hu, Qi Guo, Cagri Ozcaglar, Xianren Wu, Krishnaram Kenthapadi, and Sahin Cem Geyik. 2018. Towards deep and representation learning for talent search at linkedin. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2253–2261.

[17] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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)*. 3982–3992.

[18] Rui Yan, Ran Le, Yang Song, Tao Zhang, Xiangliang Zhang, and Dongyan Zhao. 2019. 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*. 914–922.

[19] Chen Yang, Yupeng Hou, Yang Song, Tao Zhang, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Modeling Two-Way Selection Preference for Person-Job Fit. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 102–112.

[20] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Kaiyuan Li, Yushuo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *CIKM*.

[21] Chen Zhu, Hengshu Zhu, Hui Xiong, Chao Ma, Fang Xie, Pengliang Ding, and Pan Li. 2018. Person-job fit: Adapting the right talent for the right job with joint representation learning. *ACM Transactions on Management Information Systems (TMIS)* 9, 3 (2018), 1–17.