Embedding-based Recommender System for Job to Candidate Matching on Scale

Jing Zhao, Jingya Wang, Madhav Sigdel, Bopeng Zhang, Phuong Hoang,
Mengshu Liu and Mohammed Korayem
CareerBuilder
Peachtree Corners, Georgia, US
{jing.zhao,jingya.wang,madhav.sigdel,bopeng.zhang,phuong.hoang,mengshu.liu,mohammed.korayem}@careerbuilder.com

ABSTRACT

Large scale matching between job posts and candidates is a major opportunity and challenge in online recruitment domains. In this paper, we discuss the techniques for applying an embedding-based recommendation system at large scale for job to candidates matching at CareerBuilder. To learn the comprehensive and effective embedding for job posts and candidates, we construct a fused-embedding via different features. First, we get embeddings from the raw text in resumes and job posts using a deep learning model. Second, we extract the semantic embeddings of job categories and skills from job and skill information graphs, which include implicit information of job transitions and job-to-skill co-occurrences. Third, location embeddings are extracted from job and resume geo-locations. The clusters of fused-embeddings of jobs and candidates are then used to build and train Faiss index that supports a real time approximate nearest neighbor search. We further add other contextual information to generate the final job to candidate matching. Both the offline and online evaluation results indicate a significant improvement of our proposed two-stage embedding-based system compared to the baseline system based on Apache Solr document matching. The overall improvement of our job to candidate matching system demonstrates feasibility and scalability in a major online recruitment platform.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

recommender system, job to candidate matching, deep learning, text embedding, approximate nearest neighbor index.

ACM Reference Format:

Jing Zhao, Jingya Wang, Madhav Sigdel, Bopeng Zhang, Phuong Hoang, Mengshu Liu and Mohammed Korayem. 2021. Embedding-based Recommender System for Job to Candidate Matching on Scale. In *Proceedings of The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'21 IRS Workshop)*. ACM, New York, NY, USA, 8 pages.

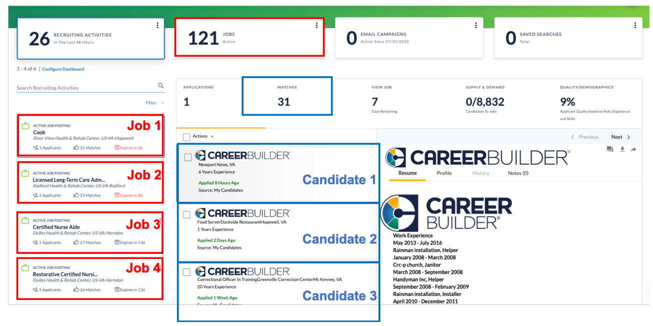

Figure 1: Snapshot of CareerBuilder's online recruitment panel and user interface.

1 INTRODUCTION

Efficient and real-time job candidate matching service is not only highly desirable for employers and job seekers, but is also beneficial to the long-term socioeconomic well-being [1]. The number of both job postings and hiring events through online recruitment platforms has grown rapidly in recent years [2]. Especially because of the impact of the COVID-19 pandemic, millions of employers and job seekers would prefer to conduct their hiring or job-seeking through the online recruitment platform [3]. Careerbuilder has the largest online job boards and provides varieties of online recruitment services in the human capital domain. Therefore, the online recruitment matching system has been one of the key services that support CareerBuilder's core business and serves millions of customers and users globally. Figure 1 illustrates a typical job to candidate recommendation scenario that takes place at Careerbuilder every day. The red boxes highlight the posted jobs from the employer and the blue boxes highlight the matched candidates recommended by the algorithm.

With millions of job postings and resumes submitted or updated at CareerBuilder every day, the most critical challenge is to build a recommender system that allows employers to target their fitting candidates and allow the job seekers to find their desired jobs in real-time. To address this challenge, we propose a two-stage recommendation system using an embedding-based approach (Figure 3). A fused embedding strategy that applies deep learning [4, 5], representation learning with job-skill information graph [6] and geolocation calculator [7] techniques are used for both jobs and candidates. We also implement the Faiss index for clustering and compressing of the embeddings, which can conduct approximate nearest neighbor search for candidate retrieval at runtime [8, 9].

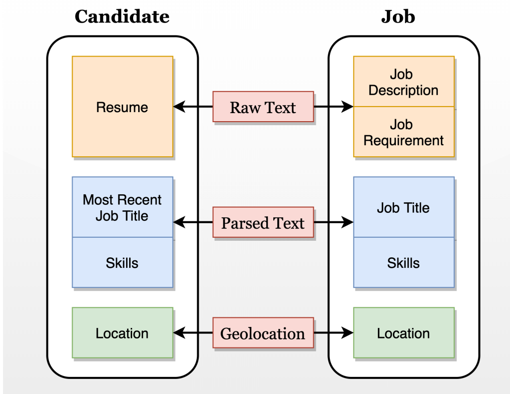

Figure 2: Contextual mapping between job and candidate.

There are several advantages of using embedding-based recommendation.

(1) Scalability: Easy to scale on industrial level with embeddings of millions to billions of items with Faiss.
(2) Sparsity/Similarity: Content-based embedding provides an alternative way to measure user-item interaction. The pairwise similarity can be easily computed using l_2 distance or cosine similarity.
(3) Cold-Start: Mitigate the cold-start issue as this content-based approach does not rely on individual user behavior data.

As for designing the recommender system for online recruitment, a major characteristic that distinguishes it from e-commerce, stream media and social network recommendation scenarios is that the contexts of user and item are likely to be symmetrical. Figure 2 illustrates such symmetric structure in terms of the context mapping between job and candidate. Active candidates have the motivation to provide full-profile information as it raises their chances to be discovered by recruiters through search and platform recommendations. At the core of our recommender system, we take advantage of such a symmetric structure of contextual mapping to construct a fused embedding using a combination of different strategies. We apply a convolution neural network (CNN) based end-to-end approach to learn the effective embedding of the raw text. This deep learning embedding model is equipped with the domain-specific vocabulary to process the text paragraphs from the resume, job description and job requirement. However, deep learning-based models are typically more effective for generalized natural language processing instead of conducting the contextual enrichment for the semantic entity extraction. Therefore, we also implement a representation learning model based on the job-skill information graph to parse job title and skills, which includes implicit information of job transition and job-skill co-occurrence that is crucial for the job to candidate matching. Moreover, a geolocation calculator that converts longitude and latitude to three-dimensional Cartesian coordinates is used to construct the location vector. With these three embeddings, we construct a fused embedded representation for both job and candidate by concatenated them together after a weight factor empirically assigned to each component.

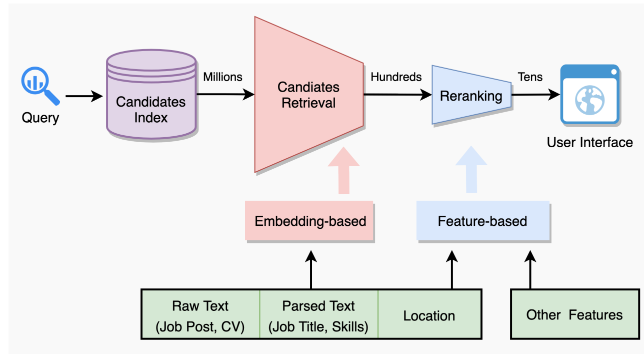

Figure 3: The Architecture of the two-stage online recruitment recommender system.

2 RELATED WORKS

2.1 Recommender System

Content-based recommender system has the inherent advantages in generalization and mitigating cold-start problems. The content-based embedding strategy allows an easy multi-feature convolution to achieve efficient and reliable item retrieval. Dating back to the classical matrix factorization framework, the content-based features have long been incorporated in the recommendation model [10]. The Factorized Machine can be used as a more generalized model for any content-based feature embeddings [11]. The rapid development of deep neural networks(DNN) in recent years has opened a new racetrack for developing recommender systems. Researchers at YouTube proposed a recommender system with a Wide & Deep neural networks architecture [12]. They *et al* proposed a neural network based collaborative filtering architecture (NCF) for modeling user-item interactions [13]., although Rendle et al. argue that simple dot-product substantially outperforms NCF learned similarities [14]. The success of recommender system in e-commerce, media and social network has promoted the development of new technologies in this field. For example, knowledge graph has been utilized to build the billion-scale commodity embedding at Alibaba [15]. Wang *et al.* also suggested propagating user preferences on the knowledge graph for the recommender system [16]. As for developing a more dynamical recommender system that also addresses the often delayed logged user feedback, researchers at Google implemented a policy-gradient-based algorithm that adopted reinforcement learning to build a recommender system [17]. On this basis, a more sophisticated off-policy learning with a two-stage recommendation system is proposed by Ma *et al* [18].

2.2 Recommender System in the Online Recruitment Domain

Job and recruitment recommendations in the human capital domain are the particular applications of recommender system that involves text mining, semantic analysis, skill/job title normalization and other NLP techniques. Diaby *et al.* proposed a content-based job recommender system along with user interaction and connection data [19]. Rafter *et al.* proposed a user-based collaborative filtering (CF) system that utilizes the overlapping of interacted jobs as the

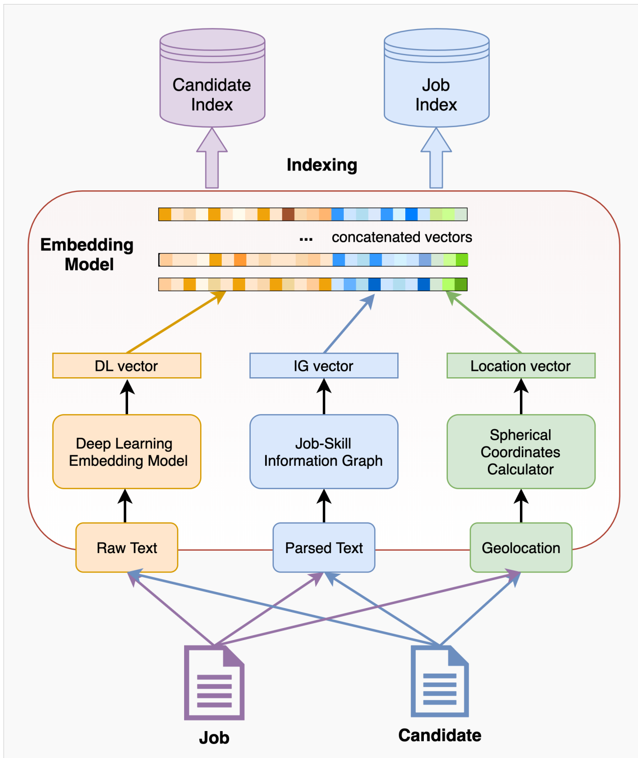

Figure 4: Fused-embedding model with vectors from deep learning embedding model (DLEM), skill-job information graph and geolocation calculator.

similarity measure between two users. They also applied a nearest neighbor search approach to generate recommendations [20]. To overcome the sparsity and cold-start problems of the classical CF method, Shalaby *et al.* built a scalable item-based recommendation system by leveraging a directed graph of job connections to represent user behaviors and contextual similarity [21]. Bian *et al.* proposed a deep global match network for capturing the global semantic interactions between job posting and candidate resume at both sentence and global levels [22]. Jiang *et al.* proposed of using deep learning and LSTM to learn the explicit and implicit interactions between job and candidate to get a more comprehensive and effective representation for the matching [23].

3 CANDIDATE MATCHING SYSTEM AND ARCHITECTURE

The proposed architecture of the two-stage recommendation system consists of two major components (Figure 4):

(1) First stage retrieval component utilizes a two-tower embedding structure to find hundreds of potential candidates from the pool of millions.
(2) Second stage rerank component takes advantage of various contextual features to allow the narrow down to a few dozen of candidates after the fine-tune scoring.

At the core of the first component, we propose a fused embedding strategy to learn the representations from raw text, parsed text and geolocation for both candidate and job.

3.1 Deep Learning Embedding Model

We train an end-to-end Deep Learning Embedding Model (DLEM) on a supervised learning task that utilizes our job application data. This allows the DLEM model to not only learn the context embedding from an NLP perspective but also be able to capture the job application behavior from the users. The DLEM consists of an input layer, a convolutional neural network (CNN) layer and an attention layer as illustrated in Figure 5. At the data generation stage, a pair of job and candidate's raw text documents (e.g. job posts and resume) are generated for the input layer. The positive pairs are particularly selected from our job application logs in which the candidate is paired with the job that he/she applied for. The negative pair is generated using random samples but the results are filtered with additional rules to remove the false negative signals. For example, the job and candidate pair that belongs to the same SOC domain are removed from the negative samples. The pairwise raw text inputs are then encoded using word2vec using a domain-specific vocabulary with a focus on the human resource and job domain. With this domain-specific encoding of the word index, we are able to construct a more space-efficient index-based representation. The input text encoding is then sent to the convolutional layer, which consists of six stacked blocks with different kernel sizes, ranged from 1 to 10. Each stacked block contains three consecutive convolutional blocks, in which a pipeline of 1D-convolution, batch normalization and max-pooling is considered as a unit processing. The stacked blocks with different kernel sizes are aimed to construct the distributed representations of the sentence instead of just the lexical features. An attention layer is built from the outputs from the stacked blocks and their saliency, inspired by the recent progress of the Transformer architecture [24]. The output context vector of the attention layer is then sent to the fully-connected layers (FC layers) with RELU activation. FC layers also determine the desired output dimension of the embedding vector based on the need. As for training the DLEM, we choose relevance-based binary cross-entropy as the loss function.

C and J represent the set of candidates and jobs. The application mapping A is defined as $A : C \longrightarrow 2^J$, and relevancy mapping R is defined as $R : C \times J \longrightarrow \{0, 1\}$

$$\text{such that} \quad R(c, j) = \begin{cases} 1, & \text{if } c \text{ is relevant to } j \\ 0, & \text{otherwise} \end{cases}.$$

$g(x)$ is defined as the embedding from DLEM model and $[g(c); g(j)]$ represents the concatenation of candidate embedding vector $g(c)$ and job embedding vector $g(j)$.

$$Loss(C, J) =$$

$$\sum_{c \in C} \sum_{j \in A(c)} -R(c, j) \log(p) + (1 - R(c, j)) \log(1 - p)$$

$$\text{where} \quad p = \sigma\left(w^T [g(c); g(j)] + b\right)$$

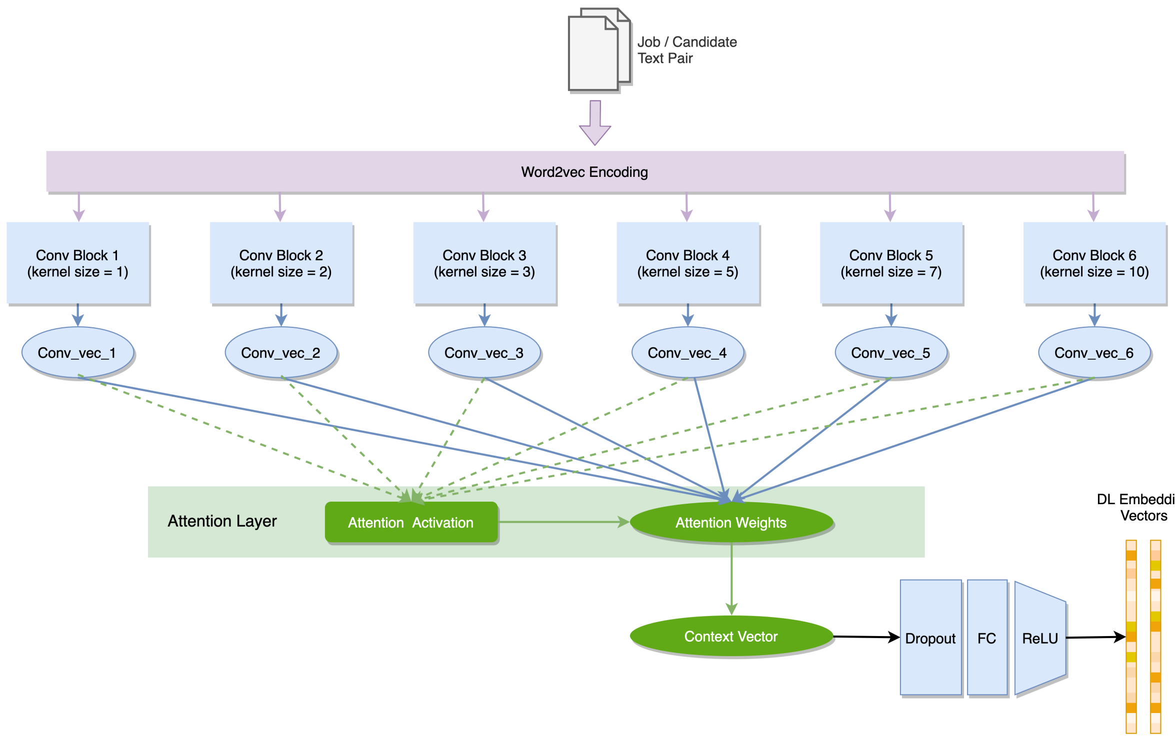

Figure 5: Architecture of the deep learning embedding model (DLEM).

Figure 6 illustrates the t-distributed stochastic neighbor embedding (t-SNE) plots of 10,000 sample jobs' embeddings obtained from (a) DLEM and (b) distilBERT pre-trained model [22]. Each job is also color labeled with 23 major job categories and one unknown category based on the Standard Occupational Classification System (SOC). The t-SNE plot shows that our DLEM is very effective in job classification as the job cohorts with different colors are clearly clustered in different regions. For example, job category: 29-0000, Healthcare Practitioners and Technical Occupations, and job category: 13-0000, Business and Financial Operation Occupations are circled on the plot, which are clearly separated. In contrast, we cannot observe a structural clustering of the embeddings obtained from the pre-trained distilBERT model. This might be caused by the lack of a specific domain dictionary and labeled training data for the distilBERT model.

3.2 Representation Learning with Job-Skill Information Graph

Job title and skills are considered the most important semantic entities as they are (semi-)structured fields and contain enriched information in the job-related documents. Traditionally, semantic matching using job title and skill entities has been the focuses for job classification and job recommendation tasks. Herein, we take advantage of a representation learning model that utilizes the information graph from job transition network, job-skill network and skill co-occurrence network [6]. The model uses both Bayesian personalized ranking and margin-based loss functions to learn the vector representation for the semantic entities and allows us to encode the local neighborhood structures captured by the information graphs. The following three objective functions are used to learn the representation for W and W', which correspond to the representation of job title and skill, respectively.

$$O^{jj} = \min_{W} - \sum_{(x,y,z)\in\mathcal{D}^{jj}} \ln\sigma\big(\langle \mathbf{w_x}\cdot\mathbf{w_y}\rangle - \langle \mathbf{w_x}\cdot\mathbf{w_z}\rangle\big)$$

$$O^{ss} = \min_{W'} - \sum_{(x,y,z)\in\mathcal{D}^{ss}} \ln\sigma\big(\langle \mathbf{w}'_\mathbf{x}\cdot\mathbf{w}'_\mathbf{y}\rangle - \langle \mathbf{w}'_\mathbf{x}\cdot\mathbf{w}'_\mathbf{z}\rangle\big)$$

$$O^{js} = \min_{W,W'} - \sum_{(x^j,y^s,z^s)\in\mathcal{D}^{js}} \ln\sigma\big(\langle \mathbf{w_{x^j}}\cdot\mathbf{w}'_{\mathbf{y}^s}\rangle - \langle \mathbf{w_{x^j}}\cdot\mathbf{w}'_{\mathbf{z}^s}\rangle\big)$$

Where $\mathcal{D}^{jj}$ represents the transition relationship of job triplets (x, y, z), $\mathcal{D}^{ss}$ represents the co-occurrence of skill triplets and $\mathcal{D}^{js}$ represents the relationship between (job, skill, skill) triplets. $\langle \mathbf{w_x}\cdot\mathbf{w_y}\rangle$ is the dot product of two embeddings, which is then used as the input for the sigmoid function $\sigma(x)$ to calculate the probability.

$$O\left(\mathbf{W},\mathbf{W}'\right) = \min_{W,W'} O^{jj} + O^{js} + O^{ss} + \lambda\left(\|\mathbf{W}\|_F^2 + \|\mathbf{W}'\|_F^2\right)$$

To unify these three types of networks between job and skill, the joint objective function with l_2 normalization is applied to avoid the over-fitting of representation W and W'.

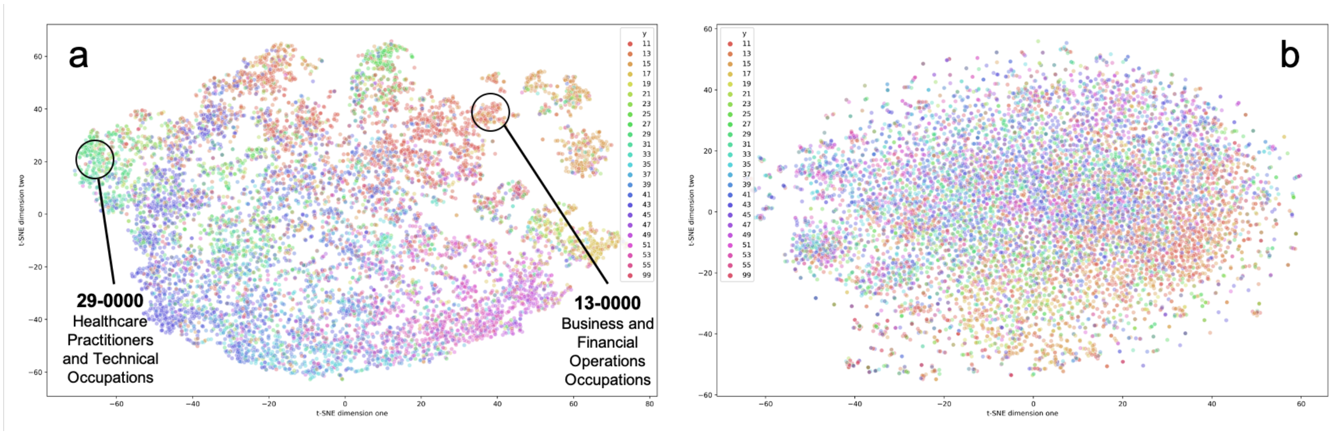

Figure 6: t-SNE plot of job embedding vectors from (a) DLEM and (b) distilBERT that are color labeled with 24 SOC categories.

3.3 Geolocation Calculator

In the geolocation part, we transforms the spherical coordinates of latitude θ and longitude ϕ to their Cartesian coordinates $[x, y, z]$ using the following equations:

$$x = \cos(\theta)\cos(\phi)$$
$$y = \cos(\theta)\sin(\phi)$$
$$z = \sin(\theta)$$

The Cartesian coordinates location vector $c = [x, y, z]$ has a straightforward advantage for conducting dot product operations between two vectors. The larger dot product $\langle \mathbf{c_1} \cdot \mathbf{c_2} \rangle$ between the location vectors, the shorter distance between these two locations. This relationship is revealed by the following equation:

$$d = 2r \cdot \arcsin\left(\frac{\sqrt{r^2 - 2\left\langle \mathbf{c_1} \cdot \mathbf{c_2} \right\rangle}}{2}\right)$$

in which r is the radius of the earth and d is the great-circle distance between two locations on the earth. Therefore, it has the same property as the content-based embeddings when compares to the pair-wise similarity using dot product operation. So the Cartesian location vector is incorporated in the fused embeddings as well.

3.4 Approximate Nearest Neighbor Search

The embeddings from DLEM $\mathbf{v_{dlem}}$, job-skill information graph $\mathbf{v_{ig}}$ and geolocation calculator $\mathbf{v_{geo}}$ are concatenated together with a set of empirically assigned weights for with each component. The concatenated embedding $\mathbf{v_{fused}}$ are defined as:

$$\left[\mathbf{w_1} \cdot \mathbf{v}_{\mathbf{dlem}}^{\top}; \mathbf{w_2} \cdot \mathbf{v}_{\mathbf{ig}}^{\top}; \mathbf{w_3} \cdot \mathbf{v}_{\mathbf{geo}}^{\top}\right] \rightarrow \mathbf{v}_{\mathbf{fused}}^{\top}$$

After constructing the fused embedding vectors, we employ the Faiss index to store all of our item embeddings for search and retrieval. This brings several advantages:

(1) Faiss index requires less space for storage due to product quantization of the embedding vectors [23], which is essential for both our offline spark pipeline and online services that possess tight memory restriction.
(2) It is easy to be integrated into the system for item retrieval. The inverted file index (IVF) allows a runtime approximate nearest neighbor search from millions or even billions of items.
(3) We can easily evaluate the similarity score between job and retrieved candidates using the inner product or l_2 metric from the index.

There are several factors we consider during the customization of the Faiss index. 1. We choose IVF algorithms and carefully tune the number of coarse clusters during the coarse quantization, which typically works through the K-means clustering; 2. As for the fine-grained quantization, we applly OPQ to transform data prior to the product quantization, which is recommended by Huang *et al* [9]; 3. We also tune the *nprobe* parameter that decides how many coarse clusters will be scanned during the query, which may affect the retrieval's performance and recall. Overall, the architecture of both job and candidate index resembles the two-tower model, which demonstrates its effectiveness in text-based information retrieval in large-scale recommender system [24, 25].

3.5 Reranking with Contextual Features

After the first stage candidate retrieval, the final ranking score for each candidate is calculated by a weighted linear equation that aggregates the scores we obtained from the first-stage relevancy score as well scores from contextual features of job and candidates. These context-based scores include skill matching, location restriction, year of experience and education level. The weights representing the importance of each score and are tuned empirically. The final ranking score is then used for reranking to generate the second-stage recommendation result. The fine-tuning in the reranking stage also allows us to implement some specialization for a certain type of job. Since the pandemic, there is a significant increase amount of Work From Home (WFH) or remote jobs that appeared in the job posts [29]. This type of job typically has very little or no location restriction, which is different from a lot of front-line occupations. To reflect such distinction in our recommendation, we adjust the location weight during reranking, which results in a more suited and robust candidate recommendation overall.

3.6 System Implementation

The job and candidate data are stored in our in-house Hadoop clusters which allows distributed processing using Spark. The deep learning model is served in the spark jobs to create document embeddings. The fused-embeddings are then used to train the Faiss index with coarse quantization and product quantization (PQ). The published inverted file (IVF) Faiss index is then served for the candidate retrieval in the batch offline mode. All the spark jobs are scheduled by the Oozie coordinator that runs periodically. At the end of the workflow, the generated recommendation results are delivered to the production database.

4 EXPERIMENT AND RESULT

The test and evaluation of our job to candidate matching system take advantage of a rich corpus of job and candidate data at CareerBuiler.com. CareerBuilder operates the largest job posting board in the U.S. and has quickly expanded its global presence in recent years. On the daily routine, millions of job postings and more than 60 million actively searchable resumes need to be processed for the online recruitment service. In this section, we describe the details of the case study, testing and evaluation of our system.

4.1 Case Study of Matching Scenarios

The two-stage job-to-candidate matching system has achieved impressive matching quality which is showcased in the table. Table 1 presents 3 cases with jobs and their top candidates. Each job has its job title, job requirement, job description and location information. The corresponding information from the candidate, such as most recent title, skills, work experience and location is provided as well. As for case 1, the database developer job, the top candidate is a matching for all four aspects. As for case 2, the licensed practical nurse (LPN) job, we notice that top candidates meet the requirement for LPN license and other required certificates. Case 3 regional sales representatives job does not have a specific location other than the north American region, therefore a broader spectrum of candidates can be selected as long as they meet the location requirement. This case also applies to work from home job scenario in which the candidate's working location is not restricted. Overall, our job-to-candidate matching system provides satisfied matching results from title, description, requirement and location perspectives, which indicates the success of our two-stage model and fused-embedding strategy.

4.2 Offline Evaluation

The DLEM cutoff parameter γ, the fused embedding weight parameters w_1, w_2, w_3 and the score aggregation parameters during the heuristic re-ranking are all tuned empirically through multiples rounds of test and evaluation. QA team and professional recruiters at CareerBuilder also participate in the qualitative evaluations for several rounds. They are asked to validate the list of recommended candidates from both jobs in specific domains and randomly sampled jobs. They give a qualitative score and leave comments for each job-candidate pair. The feedback is used as the empirical signal for us to better tweak the parameters and search for the optimal parameter combinations for our system. After the fine-tuning of the parameters, we compare the quality score and nDCG between our baseline model and our two-stage matching model. For background information, our baseline model is a solr-powered recommendation engine that utilizes hierarchical classification and a content-based approach to retrieve relevant candidate profiles. As for the offline evaluation, 150 jobs spanning over multiple job categories with 3k matching candidates are manually examined. The overall quality score of recommendation has improved ~19%, and the nDCG score has improved ~18% (Figure 7).

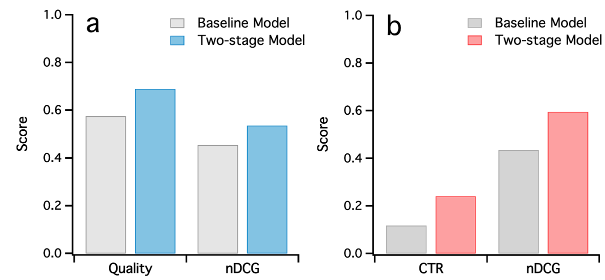

Figure 7: Offline and online evaluation results between baseline model and two-stage matching system.

4.3 Online Evaluation

As for the online evaluation, we compare the traffic over 4 months between the baseline model and two-stage matching system. Over 120k user impressions and click events are used to calculate the nDCG score and click through rate (CTR) for comparison. The CTR and nDCG score both show a significant improvement over a three months period of time. The CTR increases ~104%, and nDCG score increases ~37%. These results are summarized in Figure 7. In summary, both offline and online evaluation results suggest that our two-stage matching system significantly improves the matching quality, resulting in higher traffic and CTR from our users.

5 CONCLUSION

The online recommender system has gained considerable attention in both academia and industry in recent years, as quickly evolved technology plays a key role in bringing an enormous amount of commercial and social values. The online recruitment service at CareerBuilder takes advantage of such progress to serve millions of job applicants and employers. To bring the full potential of the recommender system for online recruitment, we propose a two-stage embedding-based recommender system for job to candidates matching. The architecture of this system consists of a two-stage recommendation procedure: a fused-embedding component for candidate retrieval and a fine-tuning reranking module. The successful deployment of embedding-based job to candidate matching system in production creates the avenue to optimize the system end to end

Table 1: Case Study of Job to Candidate Matching Scenarios

Case	Job		Candidate	
1	Title	Database Developer	Previous Title	Sr. Database Developer
	Requirement	BS in Computer Science, 5+ years of experience working with Microsoft SQL Server, database code development, data modeling with ER/Studio or ERWin, data warehouse design, SSIS	Skills	Web development and database architecture, Microsoft SQL Servers, Java, C#, Visual Basic, Microsoft Access
	Description	Design and development of database objects, populate and maintain the data in the data warehouse, creation of ETL programs in a Microsoft SQL Server environment.	Work Experience	• Sr. Database Software Developer (13 - 20) • Sr. Database Developer (09 - 13) • Sr. Web Program Developer (02 - 09)
	Location	San Diego, CA	Location	Beaumont, CA
2	Title	Licensed Practical Nurse (LPN)	Previous Title	Licensed Practice Nurse
	Requirement	Current LPN license in good standing, CPR certification, Minimum 1 year clinical experience.	Skills	LPN, CPR BLS certified, nursing care practice, physical examination, IV drug therapy management, EMR systems
	Description	Client assessment, administration of prescribed medication, treatment and therapy, clinical works, supply management, emergency Management.	Work Experience	• Licensed Practical Nurse Endoscopy (19 - 20) • LPN Charge Nurse (16 - 19) • State-tested Nurse Assistant (09 - 15)
	Location	Carnegie, PA	Location	Bulter, PA
3	Title	Regional Sales Representative	Previous Title	Regional Sales Representative
	Requirement	Require 5+ years of outside sales experience, ability to travel up to 50% of the time, interpersonal communication skills, experience with CRM platforms, proficient in Microsoft Office	Skills	Client relationship management, communication and negotiation, proficient in salesforce and other CRM platforms, analytical skills, Microsoft Office
	Description	Generate, develop, and maintain a robust pipeline of qualified opportunities, actively conduct cold and warm calling to prospective leads, manage sales process.	Work Experience	• Regional Sales Representative (19-20) • Business Development Specialist (18 - 19) • Account Manager II (14-18)
	Location	North America	Location	Bedford, TX

through user feedback. We also provide valuable experiences in architecture design, parameter tuning and later-stage optimization. Overall, our two-stage job to candidate matching system shows a significant improvement over the baseline model by measures of CTR and nDCG in real world production environment, which provides an excellent example for deploying an embedding-based recommender system for applications of job to candidate matching on the scale.

ACKNOWLEDGMENTS

The authors would like to pay special tribute to Bopeng, who sadly passed away during the drafting of this paper. We dedicate this paper to Bopeng to recognize his crucial contribution and achievement during his days at CareerBuilder.

REFERENCES

[1] Pologeorgis, N. Employability, the Labor Force, and the Economy. 2019. https://www.investopedia.com.

[2] LinkedIn Workforce Report Janurary 2021 United States.

[3] Columbus, L. Remote Recruiting In A Post COVID-19 World. 2020. https://www.forbes.com.

[4] Yuan, J., Shalaby, W., Korayem, M., Lin, D., Aljadda, K. & Luo, J. 2016. Solving cold-start problem in large-scale recommendation engines: A deep learning approach. 2016 IEEE International Conference on Big Data (Big Data).

[5] Wang, J., Abdelfatah, K., Korayem, M. & Balaji, J. 2019. DeepCarotene -Job Title Classification with Multi-stream Convolutional Neural Network. 2019 IEEE International Conference on Big Data (Big Data).

[6] Dave, V. S., Zhang, B., Hasan, M. A., Aljadda, K. & Korayem, M. 2018. A Combined Representation Learning Approach for Better Job and Skill Recommendation. Proceedings of the 27th ACM International Conference on Information and Knowledge Management.

[7] Liu, M., Wang, J., Abdelfatah, K. & Korayem, M. 2019. Tripartite Vector Representations for Better Job Recommendation. Retrieve from https://arxiv.org/abs/1907.12379v1

[8] Johnson, J., Douze, M. & Jegou, H. 2019. Billion-scale similarity search with GPUs. IEEE Transactions on Big Data.

[9] Huang, J., Sharma, A., Sun, S., Xia, L., Zhang, D., Pronin, P., ... Yang, L. 2020. Embedding-based retrieval in Facebook search. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. doi:10.1145/3394486.3403305

[10] Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In 2008 Eighth IEEE International Conference on Data Mining. 263–272.

[11] S. Rendle. 2010. Factorization Machines. In 2010 IEEE International Conference on Data Mining. 995–1000.

[12] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir, et al. Wide & deep learning for recommender systems. 2016. Technical report.

[13] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.S. Chua. 2017. Neural Collaborative Filtering. In Proceedings of the 26th International Conference on World Wide Web (WWW '17). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 173–182.

[14] Rendle, S., Krichene, W., Zhang, L. & Anderson, J. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited.

[15] Wang, J., Huang, P., Zhao, H., Zhang, Z., Zhao, B. & Lee, D. L. 2018. Billion-scale Commodity Embedding for E-commerce Recommendation in Alibaba. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.

[16] Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X. & Guo, M. 2018. RippleNet. Proceedings of the 27th ACM International Conference on Information and Knowledge.

[17] Chen, M., Beutel, A., Covington, P., Jain, S., Belletti, F., & Chi, E. H. 2019. Top-K Off-Policy Correction for a REINFORCE Recommender System. Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining.

[18] Ma, J., Zhao, Z., Yi, X., Yang, J., Chen, M., Tang, J., Chi, E. H. 2020. Off-policy Learning in Two-stage Recommender Systems. Proceedings of The Web Conference 2020.

[19] Diaby, M., Viennet, E. & Launay, T. 2013. Toward the next generation of recruitment tools. Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM '13.

[20] Rafter, R., Bradley, K. & Smyth, B. 2000. Automated Collaborative Filtering Applications for Online Recruitment Services. Lecture Notes in Computer Science Adaptive Hypermedia and Adaptive Web-Based Systems, 363-368.

[21] Shalaby, W., Alaila, B., Korayem, M., Pournajaf, L., Aljadda, K., Quinn, S. & Zadrozny, W. 2017. Help me find a job: A graph-based approach for job recommendation at scale. 2017 IEEE International Conference on Big Data (Big Data).

[22] Bian, S., Zhao, W. X., Song, Y., Zhang, T. & Wen, J. 2019. Domain Adaptation for Person-Job Fit with Transferable Deep Global Match Network. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP).

[23] Jiang, J., Ye, S., Wang, W., Xu, J. & Luo, X. 2020. Learning Effective Representations for Person-Job Fit by Feature Fusion. https://arxiv.org/abs/2006.07017v1

[24] Brain, A., Vaswani, A., Brain, G., Profile, G., Brain, N., Shazeer, N., et al, 2017, Attention is all you need. Advances in Neural Information Processing Systems 30 (NIPS 2017).

[25] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A. & Rush, A. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. https://arxiv.org/abs/1910.03771

[26] Jégou, H., Douze, M., & Schmid, C. 2011. Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(1), 117-128.

[27] Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning Text Similarity with Siamese Recurrent Networks. In Rep4NLP@ACL.

[28] Guo, M., Yan, N., Cui, X., Wu, S. H., Ahsan, U., West, R. & Jadda, K. A. 2020. Deep Learning-based Online Alternative Product Recommendations at Scale. Proceedings of The 3rd Workshop on E-Commerce and NLP.

[29] Ability to work from HOME: Evidence from two surveys and implications for the labor market in the COVID-19 PANDEMIC : Monthly Labor Review. 2020. https://www.bls.gov/opub/mlr/2020/article/ability-to-work-from-home.htm