Talent Search and Recommendation Systems at LinkedIn: Practical Challenges and Lessons Learned

Sahin Cem Geyik, Qi Guo, Bo Hu, Cagri Ozcaglar, Ketan Thakkar,
Xianren Wu, Krishnaram Kenthapadi
LinkedIn Corporation, USA

KEYWORDS

Talent Search & Recommendation, Candidate Retrieval & Ranking

1 TALENT SEARCH AND RECOMMENDATION: PRACTICAL CHALLENGES

LinkedIn Talent Solutions business contributes to around 65% of LinkedIn's annual revenue, and provides tools for job providers to reach out to potential candidates and for job seekers to find suitable career opportunities. LinkedIn's job ecosystem has been designed as a platform to connect job providers and job seekers, and to serve as a marketplace for efficient matching between potential candidates and job openings. A key mechanism to help achieve these goals is the *LinkedIn Recruiter* product, which enables recruiters to search for relevant candidates and obtain candidate recommendations for their job postings.

We highlight a few unique information retrieval, system, and modeling challenges associated with talent search and recommendation systems:

(1) The underlying query to the talent search system could be quite complex, combining several structured fields (such as canonical title(s), canonical skill(s), company name) and unstructured fields (such as free-text keywords). Depending on the application, the query could either consist of an explicitly entered query text and selected facets (talent search), or be implicit in the form of a job opening, or ideal candidate(s) for a job (talent recommendations). Our goal is to determine a ranked list of most relevant candidates in real-time among hundreds of millions of structured candidate profiles. Consequently, robust standardization, efficient indexing, candidate selection, and multi-pass scoring/ranking systems are essential [2, 6].

(2) Unlike traditional search and recommendation systems which solely focus on estimating how relevant an item is for a given

query, the talent search domain requires mutual interest between the recruiter and the candidate in the context of the job opportunity. In other words, we require not just that a candidate shown must be relevant to the recruiter's query, but also that the candidate contacted by the recruiter must show interest in the job opportunity. Hence, it is crucial to use appropriate metrics (e.g., the likelihood of a candidate receiving an inMail (message) from the recruiter and also answering with a positive response) for model optimization as well as for online A/B testing, taking into account the fact that certain ideal metrics (e.g., the likelihood of a candidate receiving a job offer and accepting it) may either be unavailable or delayed [3, 5].

(3) Quite often, the recruiter or the hiring manager may not be able to express their hiring needs in the form of a search query (or even a job posting), since this often requires deep domain knowledge, as well as significant time and manual effort to come up with the best search criteria (e.g., which skills are relevant for a specific role that the recruiter is looking to fill). To address this challenge, it is desirable to support search based on ideal candidate(s) [4], and online learning of recruiter preferences within a search session based on their instantaneous response to recommended candidates [1].

In this talk, we will present how we formulated and addressed the above problems, the overall system design and architecture, the challenges encountered in practice, and the lessons learned from the production deployment of these systems at LinkedIn. By presenting our experiences of applying techniques at the intersection of recommender systems, information retrieval, machine learning, and statistical modeling in a large-scale industrial setting and highlighting the open problems, we hope to stimulate further research and collaborations within the SIGIR community.

2 OVERVIEW OF TALENT SEARCH AND RECOMMENDATION SYSTEMS DEVELOPED AND DEPLOYED AT LINKEDIN

We next briefly describe the overall architecture of LinkedIn's talent search and recommendation engine, highlighting the key components (Figure 1). Our system can be subdivided into an online system for serving most relevant candidate results and an offline workflow for updating different machine learned models (described in greater detail in Figure 2). Our presentation covers the architecture choices, modeling design decisions, and the practical lessons learned.

Online system architecture: First, the recruiter's search request (either as explicitly entered query, or implicit in the form of job opening / ideal candidate(s)), along with the recruiter and session context, is transformed into a complex query combining structured fields (e.g., canonical title(s) / skill(s), company name, region) and
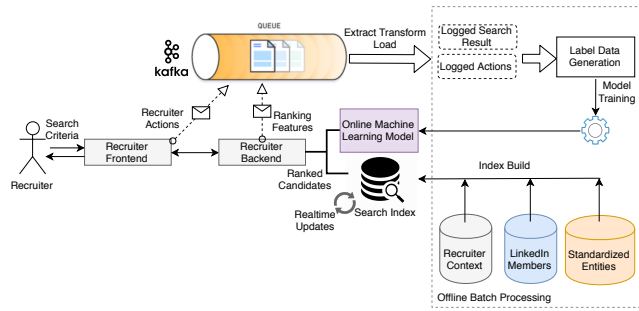
Figure 1: Architecture of LinkedIn's talent search and recommendation engine.

unstructured text keywords, and issued to LinkedIn's *Galene* search engine [6]. A candidate set of results is then retrieved from the search index based on the criteria specified, and then ranked in multiple passes using machine learned scoring models of varying complexity [1–5]. The search result set, along with the features used by the ranking model, are logged for later use for model training. Finally, front-end server gets the top ranked candidates, renders the result page, and logs recruiter interactions. The underlying search index is updated in near real-time to reflect changes in LinkedIn member data.

Offline modeling pipeline: Our offline system periodically trains the ranking models using recruiter usage logs [1–5]. The training data is generated from recruiter interactions (and candidate responses to recruiter messages) over the search results displayed. As the member data can change over time, we also log computed features along with search results, instead of generating the features during model training. The offline modeling pipeline is designed to support ease of feature engineering, incorporation of different types of machine learning models, and experimentation agility.
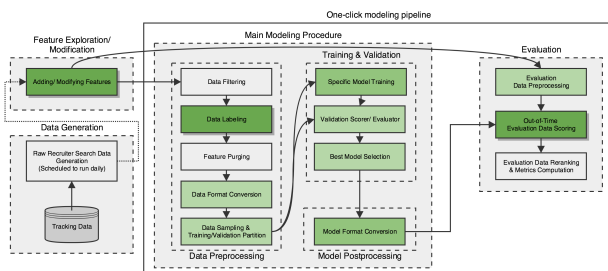

Figure 2: Offline modeling pipeline for LinkedIn's talent search and recommendation engine.

3 AUTHOR BIOS

The authors have extensive experience in the field of web-scale search and recommendation systems, and in particular, in applying data mining, machine learning, and information retrieval techniques in the talent search domain. They have built and deployed multiple generations of machine learning models and systems for real-time, low latency applications such as talent search and recommendations at LinkedIn. They have published extensively in venues such as SIGIR, KDD, WWW, WSDM, and CIKM, and also presented tutorial/industry talks about their work.

Sahin Cem Geyik is part of the AI team at LinkedIn, focusing on personalized recommendations across several LinkedIn Talent Solutions products. He received his Ph.D. degree in Computer Science from Rensselaer Polytechnic Institute in 2012, and has authored papers in top-tier conferences and journals such as KDD, INFOCOM, IEEE TMC, and IEEE TSC.

Qi Guo is part of the AI team at LinkedIn, where he applies machine learning for LinkedIn Talent Solutions products. He received his M.S. degree in Robotics from Carnegie Mellon University in 2016. He has published at IJCAI.

Bo Hu is part of the AI team at LinkedIn, where he works on relevance for LinkedIn Talent Solutions products. He received his Ph.D. degree in Computer Science from Simon Fraser University in 2014, and has authored papers in top-tier conferences and journals such as RecSys, ICDM, IEEE TKDE, and ACM TOIS.

Cagri Ozcaglar is part of the AI team at LinkedIn, where he works on relevance for LinkedIn Talent Solutions products. He received his Ph.D. degree in Computer Science from Rensselaer Polytechnic Institute in 2012, and has authored papers in top-tier conferences and journals such as IEEE BIBM, IEEE TNBS, BMC Genomics, Mathematical Biosciences.

Ketan Thakkar is a relevance engineer at LinkedIn Talent Solutions and works on improving search relevance on LinkedIn search stack. Previously, he was part of Microsoft's Bing search relevance team working on improving relevance for Bing search and ads. He received his M.S. degree in Information Technology from Bentley University in 2010.

Xianren Wu is part of the AI team at LinkedIn, leading the candidate recommendation relevance efforts within LinkedIn Talent Solutions. He previously co-founded and was the director of R&D for GageIn Inc. He received his Ph.D. degree in Electrical Engineering from U.C. Santa Cruz in 2008, and has authored papers in top-tier conferences such as WWW and CIKM.

Krishnaram Kenthapadi is part of the AI team at LinkedIn, where he leads the fairness and privacy modeling efforts across different LinkedIn applications. Previously, he was a Researcher at Microsoft Research Silicon Valley. He received his Ph.D. degree in Computer Science from Stanford University in 2006. He has published 35+ papers, filed 125+ patents, and received the CIKM best case studies paper award, the SODA best student paper award, and the WWW best paper award nomination.

REFERENCES

[1] Sahin Cem Geyik, Vijay Dialani, Meng Meng, Ryan Smith, and Xianren Wu. 2018. In-Session Personalization for Talent Search. (2018). LinkedIn Technical Report. In Submission.
[2] Viet Ha-Thuc and Shakti Sinha. 2016. Learning to Rank Personalized Search Results in Professional Networks. In *SIGIR*.
[3] Viet Ha-Thuc, Ganesh Venkataraman, Mario Rodriguez, Shakti Sinha, Senthil Sundaram, and Lin Guo. 2015. Personalized expertise search at LinkedIn. In *Big Data*.
[4] Viet Ha-Thuc, Ye Xu, Satya Pradeep Kanduri, Xianren Wu, Vijay Dialani, Yan Yan, Abhishek Gupta, and Shakti Sinha. 2016. Search by Ideal Candidates: Next Generation of Talent Search at LinkedIn. In *WWW Companion*.
[5] 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. (2018). LinkedIn Technical Report. In Submission.
[6] S. Sriram and A. Makhani. 2014. LinkedIn's Galene Search engine, https://engineering.linkedin.com/search/did-you-mean-galene.