User-Oriented Explanation for Intelligent Candidate Recommendation in Liepin.com

Roy Shan
Liepin.com
Beijing, China

Xueyan Xu
Liepin.com
Beijing, China

Zhijie Li
Liepin.com
Beijing, China

ABSTRACT

Liepin.com is one of the largest e-recruitment platforms in China. Matching millions of jobs with more than 40 million talents effectively is critical to our customer success and user experience. In this paper, we provide an overview of the architecture of our intelligent candidate recommender engine and present our user-oriented approaches to generate various types of explanations for recommendation. The experimental evaluation of our approaches shows that the explanations significantly improve customer satisfaction and business metrics.

KEYWORDS

job matching, candidate screening algorithm, recommender system, explainable recommendation

1 INTRODUCTION

With the rapid development of Internet, online e-recruitment platform has become mainstream. Liepin.com is a popular platform in China, which has more than 40 million registered talents and more than one million active job openings. Traditionally, in order to locate interesting candidates, recruiters have to consider many factors including location, title, skill, salary, education background, previous employers, personality, etc. and manually conduct search by multiple criteria. However, due to the complexity of criteria and the vast amount of talents in Liepin.com database, the manual search process is very time-consuming and cumbersome. In order to improve customer experience and platform efficiency, an automatic and accurate job candidate recommender service is needed.

In Liepin.com, recruiters post job openings and jobseekers fill out resumes online. A typical job opening has a textual job description and multiple specific requirements such as location, salary, education level, work experience, etc. On the other side, a resume usually contains information about candidate's location, job preference, work experience, education, current and expected salary.

Our candidate recommender engine for recruiters, called Bole, (1) first uses various NLP techniques to extract basic attributes from the database of job openings and resumes; (2) secondly analyzes the statistics of the behaviors of both candidates and recruiters, tries to understand their preferences and propensities; (3) thirdly applies skill/experience matching algorithms and collaborative filtering algorithms to find possible candidates; (4) and finally uses a learning-to-rank model to pick the best matches and rank them for each job opening.

The Bole engine have served our customers quite well and achieved excellent results in internal blind studies and a public contest. However, how it works seems mysterious. According to our customer study, the majority of recruiters who used the service often wanted to know why candidates are recommended and why some of the candidates are ranked higher than others. There also exist some concerns about possible biases in our machine-learned models. Therefore, we need to have the capability to answer these questions and explain the reasons behind the recommendation. We believe good explanations can improve trust, transparency, persuasiveness and customer satisfaction.

The remaining sections are organized as follows. Section 2 reviews related research on job matching or candidate screening, explainable recommender system and explanation of machine learning models. Section 3 provides an overview of the job candidate engine architecture. Section 4 describes the user-oriented approaches of automatic explanation generation for candidate recommendation. Section 5 presents the experimental study that assess the generated explanation. Section 6 concludes our work.

2 RELATED WORK

Many approaches have been applied to the problem of job matching, candidate screening or recommendation[1]. The main trend is to adapt and extend content-based or collaborative-filtering-based methods or their hybrid combinations. Machine-learning-based and knowledge-based methods have also been used to improve performance.

Machine Learning systems are becoming ubiquitous. They often execute certain tasks on their own or guide the processes of human understanding and decision-making. Therefore explainable

EARS'18, July 12, 2018, Ann Arbor, Michigan, USA
© 2018 Copyright held by the owner/author(s).
ACM ISBN
DOI: http://dx.doi.org/xxxxxxx

Machine Learning is now gaining visibility. In the Machine Learning literature, work on explanation often focused on model visualization, intrinsically interpretable models and prediction justification for non-interpretable models[2].

In recent years, many explainable recommendation approaches have been proposed[5]. For content-based recommendation, typical approaches are to inform users about the important content attributes or features of the recommended items they might be interested in. For item-based or user-based collaborative filtering recommendation, similar users to the targeted user or similar items to the recommended items can be reported to the targeted user. In matrix factorization methods, Explicit Factor Model was proposed to align latent factors with explicit product features[6].

3 OVERVIEW OF THE CANDIDATE RECOMMENDATION ENGINE

Figure 1 illustrates the high-level architecture of our Bole engine which is aimed to analyze job requirements, screen talents and recommend candidates to recruiters. The engine consists of feature extraction module, statistical analysis module, candidate recall module, filtering/ranking module and explanation module. These modules run on top of Liepin Big Data Platform.

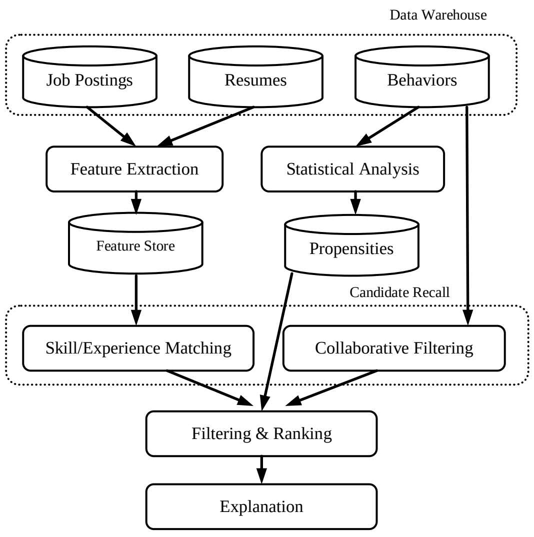


Figure 1: High-Level Architecture of Bole Engine

First, the important features of all the job openings and resumes are extracted from a data warehouse and then stored in a searchable data store. For job openings, the features include employer, industry, location, title, compensation, responsibilities and skill/experience requirements. We use Shallow Parsing technique to extract keywords from job responsibilities and skill/experience requirements and then estimate their importance scores according to requirements. For resumes, the features include work experience, education background, salary and job preference. We also apply Shallow Parsing to work experience data, extract candidates' skills/experience and estimate their strength scores. Then statistical analysis module collects and analyzes user behaviors including candidates' job hunting and recruiters' search, browsing and resume assessment. The function of this module is to quantitatively describe recruiters' preferences and propensities of candidate screening and candidates' preferences towards jobs. After gathering these important features and statistics about job openings, candidates and recruiters, in the candidate recall module, skill/experience matching is performed with a distributed search engine and collaborative filtering algorithms including memory-based and model-based methods are also applied to find potential candidates. The potential candidates are sent to the filtering and ranking module, which filters out impossible ones according to business rules and rank the rest with a Learning-To-Rank model to show recruiters the best candidates. Finally, each of the candidate record is appended with appropriate recommendation reasons generated by the explanation module.

4 USER-ORIENTED EXPLANATION FOR CANDIDATE RECOMENDATION

[3] conducted a study measuring user satisfaction with various explanation/justification types for a movie recommendation system. They found, regardless of type, 86% of users wanted the justifications they were shown. Furthermore, they also found: simple and conclusive explanations like strongly matched features and near-neighbors are most satisfying; technical explanations such as model confidence, model visualization or feature weights are difficult for average users to grasp. Therefore, we take user-oriented approaches to provide explanations for candidate recommendation and focused on three types of explanation: strongly matched features, near neighbors and deep semantic matching of roles and responsibilities.

4.1 Strongly Matched Features

Typically, recruiters screen candidates based on some important criteria including location, skills, experience, education, expected compensation, preference of employer size, etc. For a given job opening, if a recommended candidate has some features which are strongly matched with the job requirements, we first get these features directly from our Feature Store and Propensity Store, and then use data visualization to make them easy to understand.

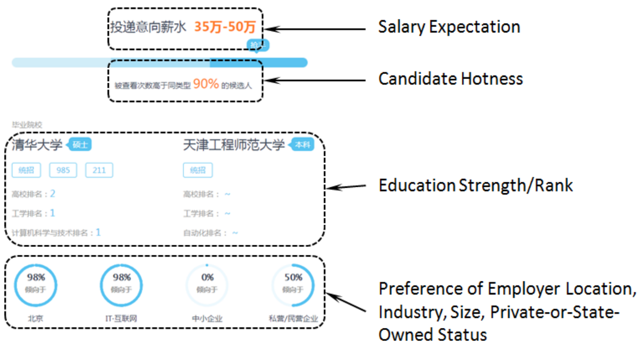


Figure 2: Visualization of Strongly Matched Features

Figure 2 is an example of the visualization of matched features. The typical features we show to recruiters are candidate's salary expectation, hotness, education background and preferences of employer location, industry, size and private-versus-state-owned status. The preferences are estimated with Empirical Bayes method.

Besides the above features, if a candidate has strong skill match with the job description, we also show his top skills with Word Cloud. The top skills are ranked by skill strength score which is estimated by our Feature Extraction module.

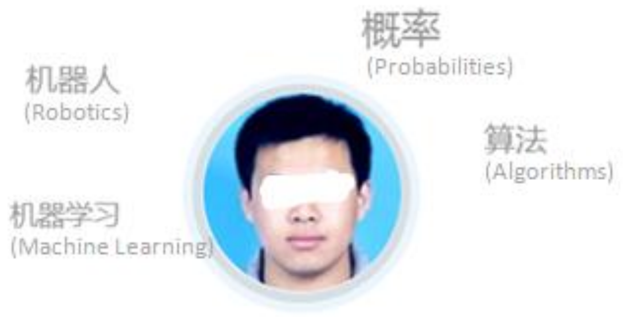
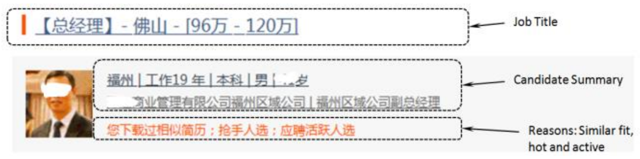


Figure 3: Word Cloud of a Candidate's Top Skills

4.2 Near Neighbor

Near-neighbor-type explanation is based on the previous candidates chosen by a recruiter or based on choices of similar recruiters. Our item-based collaborative algorithms compute Log Likelihood Ratio (LLR) score as similarity measure. To filter out noise, we use 10% percentile value as threshold. A near-neighbor-type explanation for a recommended candidate shows up only if its LLR score is greater than the threshold. Figure 4 is an example of this kind of explanation.


【总经理】 - 佛山 - [96万 - 120万]

福州 | 工作19 年 | 本科 | 男 | [redacted]岁

[redacted]业管理有限公司福州区域公司 | 福州区域公司副总经理

您下载过相似简历；抢手人选；应聘活跃人选

Figure 4: An Example of Near-Neighbor-Type Explanation

4.3 Deep Semantic Matching of Roles and Responsibilities

Roles and Responsibilities (R & R) are the core of job description. They are the key cues for guiding the processes of candidate screening and assessment. According to our recruiter study, highlighting matches of skills and R & R between a job description and a resume is one of the most requested features. Highlighting skills are straight-forward to implement since most skills are simple words or phrases. On the other hand, R & R are complex and typically represented as sentences or short texts, whose meanings do not only depend on the words in them, but also on the way they are combined. In order to highlight the matches of R & R, we apply deep neural network to embed sentences or short texts into a low-dimension vectors and then compute cosine similarity score between two vectors.

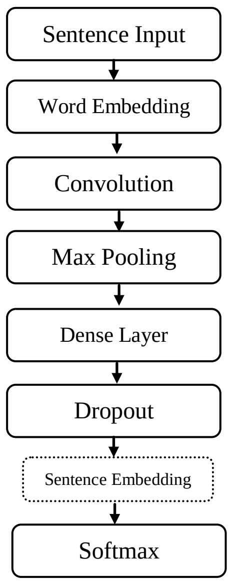


Figure 5: The Neural Network Architecture for Sentence Embedding

Figure 5 is the architecture of our neural network for sentence embedding. The network is similar to TextCNN[4]. It is trained to classify job description into one of 925 pre-defined occupations. The training data are collected from our job posting database. After the model is trained, the input vector of the final Softmax layer is the sentence embedding that we are really interested in. We use the model as an encoder, which takes as input a sentence or short text and outputs a 128-dimension vector as the embedding of the input. To compare a piece of R & R with a sentence of work experience from a resume, these two pieces of text are first encoded into two vectors *u* and *v*, and then their similarity is computed according to the following formula (1). If the score is greater than a certain threshold, a match is detected and the match in resume is highlighted.

$$\mathrm{similarity}(\mathrm{u},\mathrm{v}) = 1 - \frac{\cos^{-1}\frac{\mathrm{u}\cdot\mathrm{v}}{\|\mathrm{u}\|\cdot\|\mathrm{v}\|}}{\pi} \qquad (1)$$

5 EXPERIMENTAL EVALUATION

In this section, we describe an experimental study on Liepin.com to evaluate our explanation approaches. Our evaluation is focused on coverage and user satisfaction.

Explanation coverage is defined as the proportion of the recommended candidates having a certain type of reason in the total recommended candidates. We analyze recommendation logs in recent 7 days and calculate coverage. The result is listed in Table 1.

Type of Reason	*Coverage*
Skill match	75.5%
Might be interested in this job	28.8%
Active candidate	22.5%
Hot candidate	15.5%
Similar to previous fits	5.8%

Table 1 Coverage of Major Explanation Type

The result indicates that skill-related explanation is most commonly seen type. With multiple types of explanation, 100% coverage is achieved in total.

The fundamental purpose of a recommender system is to improve customer satisfaction and business metrics. In our scenario, customer satisfaction can be measured by user click ratios. We focused on two metrics:

- Recommendation List Click Through Rate: It is the percentage of clicks in visits of recommendation list. It is a key business metric.
- Get-More Button Click Through Rate: It is the percentage of the clicks on a get-more button in the visits of recommendation list.
- Recruiter-Wise Conversion Rate: It is the percentage of the recruiters who visit the recommended resume's detail pages in the total recruiters who read the recommendation list. It is a key business metric.

We conduct online A/B split test and investigate the effects of the explanations. Table 2 is the experiment result.

Metric	*Relative Lift*
Recommendation List Click Through Rate	16.0%
Get-More Button Click Through Rate	11.3%
Recruiter-Wise Conversion Rate	28.3%

Table 2 Lifts of Customer Satisfaction Metrics

All the lifts are statistically significant. The results show that our explanation not only improve customer experience, but also boost business metrics.

6 CONCLUSIONS

In summary, we present the architecture of our intelligent candidate recommender engine and our approaches to explain recommendation.

The engine analyzes job requirements and resumes, uses skill/experience matching, collaborative algorithms and learning-to-rank model to screen talents and recommend the best candidates to recruiters.

We take user-oriented approaches to provide recruiters conclusive and meaningful explanation for recommended candidates. The explanations are focused on strongly matched candidate features, near-neighbor indicators and deep semantic matches between job requirements and resumes. In order to effectively find match for Roles and Responsibilities, we develop a deep semantic matching model to compute semantic similarity for sentence or short text.

The experimental evaluation of our explanation approaches shows that our approaches provide excellent coverage and significantly improve customer satisfaction and business metrics.

7 ACKNOWLEDGEMENTS

We thank the following colleagues for their contributions to our candidate recommender system, data warehouse and Big Data infrastructure: Zuo Zhang, Yang Bai, Wei Meng, Jing Sun, Xiangfei Cheng and Chao Li.

REFERENCES

[1] S. T. Al-Otaibi and M. Ykhlef. 2012. A Survey of Job Recommender Systems. *International Journal of the Physical Sciences* Vol. 7(29), pp. 5127-5142. DOI: http://dx.doi.org/10.5897/IJPS12.482
[2] O. Biran and C. Cotton. 2017. Explanation and Justification in Machine Learning: A Survey. *IJCAI-17 Workshop on Explainable AI (XAI)* Proceedings, 8-13.
[3] J. Herlocker, J. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Computer Supported Cooperative Work (CSCW)*, 2000
[4] Y. Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* .
[5] Y. Zhang and X. Chen. 2018. *Explainable Recommendation: A Survey and New Perspectives*. Foundations and Trends in Information Retrieval, 2018.
[6] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. ACM, 2014a.