

# iHR: An Online Recruiting System for Xiamen Talent Service Center

Wenxing Hong  
School of Information Science  
Xiamen University  
Xiamen 361005, P. R. China  
hw@xmu.edu.cn

Lei Li<sup>\*</sup>, Tao Li  
School of Computer Science  
Florida International University  
Miami, FL 33199, USA  
{lli003,taoli}@cs.fiu.edu

Wenfu Pan  
Director of Development  
Xiamen Talent Service Center  
Xiamen 361012, P. R. China  
pan@xmrc.com.cn

## ABSTRACT

Online recruiting systems have gained immense attention in the wake of more and more job seekers searching jobs and enterprises finding candidates on the Internet. A critical problem in a recruiting system is how to maximally satisfy the desires of both job seekers and enterprises with reasonable recommendations or search results. In this paper, we investigate and compare various online recruiting systems from a product perspective. We then point out several key functions that help achieve a win-win situation between job seekers and enterprises for a successful recruiting system. Based on the observations and key functions, we design, implement and deploy a web-based application of recruiting system, named iHR, for Xiamen Talent Service Center. The system utilizes the latest advances in data mining and recommendation technologies to create a user-oriented service for a myriad of audience in job marketing community. Empirical evaluation and online user studies demonstrate the efficacy and effectiveness of our proposed system. Currently, iHR has been deployed at <http://i.xmrc.com.cn/XMRCIntel>.

**Categories and Subject Descriptors:** H.3.3[Information Search and Retrieval]: Information filtering.

**General Terms:** Algorithms, Design, Management

**Keywords:** Job Matching System, Reciprocal Recommendation, Job Recommendation, Bilateral Recommendation

## 1. INTRODUCTION

Online recruiting systems, as effective tools of recruiting services, have emerged and enjoyed explosive growth in the last decades. By using online recruiting systems, enterprises release their recruiting information, and job seekers look for jobs [12]. With the emergence of Web2.0 technologies, a gigantic amount of information relevant to job seekers and enterprises has been collected and kept increasing exponen-

<sup>\*</sup>Lei Li and Wenxing Hong contribute equally to this work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

Copyright 2013 ACM 978-1-4503-2174-7/13/08 ...\$15.00.

tially. As a result, the information becomes overloaded, which renders traditional recruiting systems ineffective.

A practical way to handle the aforementioned issue is to employ recommendation techniques to obtain the information relevant to a target user (a job seeker or a recruiter). By analyzing the characteristics of users and profiling users' personal preferences, online recruiting systems with personalized recommendation are capable of automatically retrieving a list of job positions that satisfy a job seeker's desire, or a list of talent candidates that meet the requirement of a recruiter. With the great convenience and effectiveness for both recruiter and job seekers, job recommender systems have gained broad attention in both academia [11, 19, 22, 27] and industries (e.g., Monster.com and Yahoo! Careers). However, most online recruiting systems in industries aim at providing search services for users, or simple recommendations via analyzing the unilateral preferences of users, which cannot effectively handle the mutual relations between job seekers and recruiters, or called the *reciprocity*.

A successful example in the job marketing community is LinkedIn.com, which breaks through the traditional unilateral-oriented job recommendation paradigm. In LinkedIn, the recommendation for jobs or candidates is achieved by constructing and analyzing the social network that embraces a target user. The user's personal network effectively reflects the social environment and personal preference. However, LinkedIn asks the user to explicitly manage his/her social relations. In most cases, users are not willing to proactively search jobs or candidates once they feed their personal preferences into the system. The way that LinkedIn collects the user's profile requires much efforts from the user, and therefore renders the procedure of profiling less efficient.

### 1.1 Motivation for Developing iHR

Existing online recruiting systems apply different recommendation strategies for users. The techniques for recommendation include content-based (CBR) [22, 31, 32], collaborative filtering (CFR) [7], knowledge-based recommendation (KBR) [5, 11], etc. Among these techniques, CBR is frequently used by most recruiting systems, due to the convenience of collecting users' demographic information. However, it is often difficult to achieve good user satisfaction solely based on recommendation strategies. In practice, a recruiting system should be designed and implemented from a product perspective, including user profiling, recommendation strategies, recommendation representation, user feedback, reciprocity between job seekers and enterprises, etc. For comparison, we investigate four existing recruiting systems based on these aspects, including CASPER, Proactive,

PROSPECT and eRecruiter. CASPER is a classical job application system used for enhancing the performance of JobFinder<sup>1</sup>. Proactive has different recommendation modules applied to its own website<sup>2</sup>. PROSPECT<sup>3</sup> is developed by analyzing and mining the resumes of users. eRecruiter is designed for expanding the functionality and improving the accuracy of the Absolventen.at<sup>4</sup>. Differences among these systems as well as our developed iHR are shown in Table 1.

Table 1: The comparison of recruiting systems.

	User Profiling	Approach	Layout	User Feedback	Reciprocity
CASPER	Individual information and behavior	CFR CBR	Full list	No	No
Proactive	Individual information	CBR KBR	Modular list	No	No
PROSPECT	Individual information	CBR	Full list	No	No
eRecruiter	Individual information and behavior	CBR KBR	Full list	Email	No
iHR	Individual, extracted and behavior	CBR	Full list	Comments	Yes



Figure 1: The screenshot of the system.

As shown in Table 1, the first 4 systems utilize limited information to construct user profiles, and the recommendation strategies applied to each system only focus on unilateral aspects. In practice, job recommender systems deal with people-to-people recommendation, and hence it is necessary to consider the information need of both parties being involved. In this paper, we design, develop, and deploy a web-based online recruiting application for Xiamen Talent Service Center, named as iHR. A screenshot of the system is shown in Figure 1. We highlight the major differences between iHR and other recruiting systems as follows:

- Most recruiting systems generate the users' profiles purely based on their registration inputs. iHR collects users' profiles from multiple information channels, including users' demographic information, the extracted data from external files (e.g., resumes and home pages),

<sup>1</sup>http://www.jobfinder.com.

<sup>2</sup>http://www.proactiverecruitment.co.uk.

<sup>3</sup>http://www.prospects.ac.uk.

<sup>4</sup>http://www.absolventen.at.

and users' daily activities. The fusion of multiple information sources provides a comprehensive picture of the interests of users in a job marketing community.

- Different from the unilateral recommendation of other recruiting systems, iHR incorporates the reciprocity between job seekers and recruiters into the recommendation process for the win-win situation. Such a strategy provides great insights on the underlying principle of people-to-people recommendation, and therefore makes the results more interpretable.

In addition, iHR provides comprehensive explanations on the recommendation results based on the characteristics of the recommended items. Such a representation can help enhance the user experience to a great extent. We also provide an interactive channel for job seekers and recruiters, by which job seekers can give comments on the job positions being viewed, and recruiters can send messages to job seekers. Comments are treated anonymously, and are sent back to recruiters and job seekers to help them improve the description or representation of their profiles. These features are not supported by other recruiting systems in Table 1.

For over five years, a multi-disciplinary team of researchers and officers from Florida International University, Xiamen University, Xiamen Talent Service Center and Xiamen Well Software have been working together to investigate the way of job matching. Now, iHR has become the official online recruiting website for Xiamen talent market. Under the supervision of Xiamen Talent Service Center, iHR provides an interactive platform for both job seekers and recruiters around Xiamen and adjacent areas. Over 1,700,000 individual resumes are possessed in iHR, and over 230,000 verified job positions are posted online. On average there are over 699,600 visits per day. With the considerable user repository, iHR is expected to serve as a scalable recruiting system, and to provide professional search and recommendation for both job seekers and recruiters.

## 1.2 Research Challenges and Solutions

Based on the observations we have made during our preliminary research, we have identified two key issues that need to be addressed in iHR job recommender system:

1. *How to effectively construct profiles of job seekers and recruiters?* The users' profiles are the primary source of knowledge for search and recommendation in iHR. Many different types of information can be included in the profiles. For example, for job seekers, the profiles can include the basic demographic information (e.g., age and gender), the preferred job information (e.g., salary and location), and the behavior and activity information (e.g., the job posts that have been saved and applied). Similarly, different information channels are also provided for enterprises. However, one challenge is how to effectively construct the user profiles.
2. *How to provide effective recommendations?* Due to the flooding information in job market community (thousands of thousands of resumes and job postings), it is imperative to provide effective recommendation for helping users quickly refine the information and therefore locating their expected results. However, a simple recommendation cannot satisfy the users' varied preferences. Also under the circumstance of job matching,

a successful recommendation cannot be determined by unilateral parties. Therefore, the second challenge is how to provide the reasonable recommendations for both job seekers and recruiters.

Our proposed system provides effective ways to construct user profiles and incorporates the reciprocity between job seekers and recruiters into the process for more reasonable results. In particular, to address challenge 1, we analyze users' information from different sources, and then encapsulate such information into users' self-description and users' preference; To address challenge 2, we enrich the recommendation experience by combining results from different recommendation strategies, and particularly, focusing on the reciprocity between job seekers and recruiters.

### 1.3 Roadmap

In summary, iHR is essentially a collaborative platform for job matching and selection that helps both job seekers and recruiters find their expected information. By optimally integrating advantageous resources from Xiamen Talent Service Center, Xiamen job market and Xiamen talent network, iHR has been selected as one of the most prominent and influential job matching platforms in China<sup>5</sup>.

The rest of the paper is organized as follows. §2 presents the system overview of iHR; §3 discusses our profiling strategy; In §4, we describe the recommendation process of iHR, including three different strategies for recommendation; §5 reports the system evaluation; Finally, §6 describes the system deployment and operation and concludes the paper.

## 2. SYSTEM OVERVIEW

iHR is a collaborative solution designed for job markets, on which job seekers and recruiters are able to cooperatively build an effective job matching platform. It provides users functionalities to organize their career development, to improve their corporation culture, and to construct extensive relation network. In iHR, registered users are categorized into two groups: job seekers and recruiters. For job seekers, iHR enables them to input their basic information, to upload and update their resumes, and to receive instant recommendation for job positions; For recruiters, similar functionalities are provided, except for the recommendation of job applicants. Figure 2 shows the system overview of iHR.

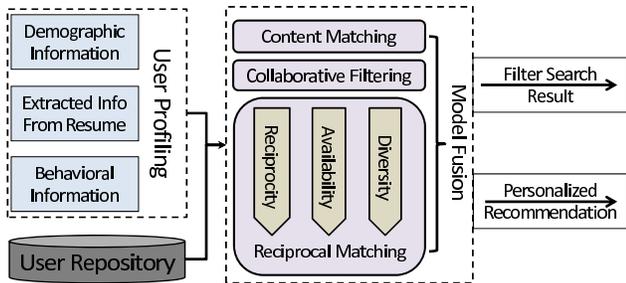


Figure 2: The system overview of iHR.

The information processing and representation functionalities are integrated into the following critical modules: *User*

<sup>5</sup>[http://www.xmrs.gov.cn/zwgk/gzdt/201008/t20100810\\_368778.htm](http://www.xmrs.gov.cn/zwgk/gzdt/201008/t20100810_368778.htm)

*Profiling, Advanced Search and Recommendation.* The seamless integration of these modules makes the system more user-friendly and customer-centric. In the following, we will provide functional details of these modules.

**User Profiling** (cf. §3): To construct user profiles for job seekers and recruiters, iHR considers multiple information resources, including users' basic information, extracted text from uploaded files or links and user's behavioral information. The fusion of different types of information enables us to comprehensively understand a user's exact interest.

**Advanced Search:** In iHR, advanced search functionality for users is provided in order to quickly obtain the information they are interested in. Although search is commonly used in most job matching systems, iHR uses some query expansion techniques [20] to expand the query keywords for more comprehensive results. In this paper, we will not focus on introducing the search service as it has been extensively studied by many existing systems[1, 28].

**Recommendation** (cf. §4): iHR provides flexible interfaces for users to obtain recommendation results related to their preferences. We implement three distinct recommendation strategies, including content filtering, collaborative filtering, and reciprocal recommendation, and also provide integrated recommendation results to users. For reciprocal recommendation, we extensively investigate the bilateral correlation between job seekers and recruiters, i.e., the reciprocity, and therefore help achieve the win-win situation among them.

Besides the major modules mentioned above, in iHR we also provide an interactive channel for job seekers and recruiters, as discussed in §1.1. Figure 3 shows an example of a job seeker's comment management.



Figure 3: The comment management of job seekers.

## 3. USER PROFILING

A natural way to enhance the search and recommendation experience is to maintain and utilize a user's profile. Users' profiles provide us extensive evidence to understand the major preference of both job seekers and recruiters, and great facilities to adjust the search and recommendation output for individuals. In iHR, a user's profile is composed of three components, i.e., the *basic* information, the *extracted* information and the *behavioral* information. In the following, we describe the detailed profiling for each component.

**Table 2: Bilateral features in a job matching system. Features with the prefix “need” are all the preference features, whereas the others are the self-description features.**

Job Seekers				Recruiters			
scalar	categorical		free text	scalar	categorical		free text
age	language	marriage	education	salary	job_type	need_degree	company_desc
work_length	sex	degree	self_desc	need_amount	trade_type	need_sex	job_desc
need_salary	work_status	need_job_type	ability	need_work_length	property_type	need_language	need_ability

### 3.1 Basic Information

A user’s basic information includes demographic data and explicit preference specified by the user. Such information is common in most job matching systems. Table 2 shows sample features of the basic information we collect from users.

In general, the user-specified features can be scalar, categorical or free text, as shown in Table 2. For scalar features, we first transform them into different ranges, e.g., we can transform the length of working experience as 0-3 years, 3-5 years, 5-10 years and 10+years. We then encode these ranges as binary features, e.g., if a user’s working experience falls into the range of 0-3 years, then this feature (0-3 years) would be set to true (1), and all other ranges would be set to false (0). With categorical attributes, we use the same strategy of dealing with working experiences ranges. For free text features, we transform the text into an  $l_2$ -normalized tf.idf-based term vector, and then combine this vector with the vector obtained based on other types of attributes. Note that the demographic information and the preference are separately processed. We denote the demographic vector as  $p^s$  and the preference vector as  $p^p$ .

### 3.2 Extracted Information

Another way to collect a user’s information is to analyze his/her external information, e.g., resumes and home pages. In general, such information might be represented as .doc, .pdf, or .html files, and the text information can be extracted from these files. Take the resumes as an example. The text in resumes contains multiple types of information, e.g., personal information (demographics), educational details (graduation school, degree, major), experiences (activities, research, skills), etc. Automatically extracting structured information from resumes of different styles and formats would be challenging. [32] analyzes the hierarchy of resume information, and then proposes a cascaded two-pass information extraction framework to automatically extract useful features. However, such a paradigm requires substantial effort to estimate parameters in the model, and therefore cannot handle large-scale resume extraction task.

In our system, we simplify the extraction procedure by treating each resume as a text document, and then extracting important features from the text. Specifically, we sample 20% of resumes from the resume repository and transform them to plain text. To ensure the coverage of sampling, we sample resumes from different domains, e.g., *Internet Technology*, *Chemical Engineering* and *Business Management*, etc. We then select a pool of features (words) from each domain, according to the weight information of each feature. Here the weight is represented as the tf.idf value of the corresponding word. For each domain, we empirically choose 1,000 features and then finalize the extracted profile as a feature vector. We concatenate such vector to  $p^s$  introduced in

§3.1. Note that each user’s  $p^s$  has an additional label, i.e., the domain name, for further comparison.

### 3.3 Behavioral Information

Besides the static information provided by users, the system has various ways to interact with users, and consequently collects users’ behavioral activities. For instance, we provide functionalities such as searching and recommendation: users can feed some input, e.g., keywords, into the system, and then click on some preferred profiles from the search result; users can also receive recommendations from the system, and then choose some of them to view. Such behavioral information would be helpful to construct the preference of users and improve the quality of user profiles.

In our system, a user’s activities, including searching and clicking, are automatically recorded and maintained in the format of log files. Table 3 shows an interpreted log file example of recruiters searching job applicants. Note that in *Search Criteria*, the three values correspond to *Keywords*, *Domain* and *Working years*, and the *Duration* means that the time that the user spends on reading candidates profiles. The log file in our system is parsed into two components: the *search interest* and the *click interest*. *Search interest* is obtained from the *Search Criteria*, presented by an interest vector, where each entry denotes a keyword associated with the domain, and the weight of the entry is the normalized duration. This vector is concatenated to the user’s preference vector. *Click interest* is obtained from the *Clicks*. Such information is useful when we perform collaborative filtering based recommendation to individuals.

## 4. RECOMMENDATION

The recommendation module is designed for users who do not have definite preferences on either job positions or job applicants. For example, for job seekers who only have some *general career interest with a broad range of preferences*, the results generated purely based on the job search module might not be able to satisfy such users’ appetite, i.e., the results need to be further refined to help the job seeker figure out his/her preferred jobs.

The recommendation module includes three different submodules, categorized by the recommendation techniques, i.e., *content-filtering* module, *collaborative filtering* module, and *reciprocal recommendation* module. In the following, we will discuss the algorithmic details within each submodule.

### 4.1 Content Filtering

The principle of content filtering methods is to sequentially find items from the search result similar to the target user’s preference in terms of “content”. In iHR, the “content” refers to user profiles that are generated in §3, including job

**Table 3: Sample log file.**

<i>User Identification</i>	<i>Search Time</i>	<i>Search Criteria</i>	<i>Duration</i>	<i>Clicks</i>
ce20180c04c41580	09/Jan/2012 13:53:50	Project Manager, IT, 10	217	John, Michael
38f04d74e6511375	09/Jan/2012 13:54:32	Programmer, IT, 3	501	Chris, Ben, Shanny,...
acef953fe42596a0	09/Jan/2012 13:55:02	Programmer & Java, IT, 2	432	Chris, Shanny, Murphy,...
3110dbe2758556bf	09/Jan/2012 13:55:28	Data Analyst, Business, 8	283	Mary, Denver, Sam
72bf2bde4b64e457	09/Jan/2012 13:56:31	Consultant, Finance, 4	621	Jack, Dory, Devon,...
437e052155f01a80	09/Jan/2012 13:57:09	Data Analyst, Business, 2	421	Mary, Sam, Denver, Nara,...
21947eb595423a15	09/Jan/2012 13:58:50	Sales, Automobile, 3	142	Claydon, Chark, Edda,...
3d5f469d3d7097d3	09/Jan/2012 13:59:29	Sales, Automobile, 5	239	Edda, Jamie

seekers’ and recruiters’ profiles. Specific similarity measurements can be adopted to evaluate the relatedness between the target user and items in the search result, e.g., the relatedness between a job seeker and a series of job posts.

In this module, we focus on evaluating how relevant the users in the search result are to the target user. Formally, given a target user  $u$ ’s profile  $p_u$  and a set of search result  $Q_u$ , our goal is to select a subset  $Q_u \subset Q_u$  such that  $\forall v \in Q_u$ ,  $v$ ’s profile  $p_v$  is relevant to  $p_u$  in terms of a predefined relevance measurement. Under the scenario of job matching, we only consider the features that indicate the preference of the target user when calculating the relevance. For example, assume a job seeker  $u$  has his/her preference profile  $p_u^p$ , e.g., what types of jobs and which salary range that  $u$  prefers. After  $u$  feeds some keywords into the system, the system returns a set of job posts  $Q_u$ . Within  $Q_u$ , we rank all the candidates  $v$  based on the similarity

$$sim(p_u^p, p_v^s) = \frac{p_u^p \cdot p_v^s}{\|p_u^p\| \times \|p_v^s\|}, \quad (1)$$

and then choose the top ranked ones as the final recommendation result. Note that in our system  $p_u^p$  and  $p_v^s$  are processed to have the same cardinality. In summary, this module is built purely based on the content of users’ profiles (i.e., job seekers and recruiters).

## 4.2 Collaborative Filtering

Collaborative filtering methods are designed based on user’s historical accessing behaviors, e.g., what kinds of job posts have been clicked by a job seeker before. It considers “similar” users’ accessing history, and then recommends to the target user a list of items that have been accessed by these “similar” users. Therefore, the key step is to find “similar” users in terms of the accessing history.

In our system, we expand the concept of collaborative filtering to a broader case, i.e., to consider some content information when calculating the similarity between users. Specifically, we take into account a user  $u$ ’s preference profile  $p_u^p$  and the search history profile  $p_u^h$ . The similarity between two users  $u$  and  $v$  (both are job seekers or recruiters),  $sim(u, v)$ , can be calculated as

$$sim(u, v) = \frac{p_u^p \cdot p_v^p}{\|p_u^p\| \cdot \|p_v^p\|} + \frac{p_u^h \cdot p_v^h}{\|p_u^h\| \cdot \|p_v^h\|} \quad (2)$$

After obtaining a list of users similar to the target user  $u$ , we sequentially check the search result to see if users in the list have accessed them or not. We then rank the search result based on the access count and recommend top ranked ones to the target user.

## 4.3 Reciprocal Recommendation

The aforementioned two modules do not take into account the properties of job matching systems. In this module, we analyze the special characteristics within the job matching domain, and then propose a reciprocal strategy.

### 4.3.1 Introduction to Reciprocal Recommender

Recently, a special class of recommender systems, called reciprocal recommender, has emerged. Reciprocal recommender systems refer to systems from which users can obtain recommendations of other individuals by satisfying preferences of both users being involved. Examples of reciprocal recommenders include online dating services, mentor-mentee matching, consumer-to-consumer marketplaces, and etc. The job matching system is also a type of reciprocal recommender systems. Reciprocal recommendation has gained increasing attention in academia in the last decade [2, 3, 4, 6, 10, 16, 23, 24, 25, 29]. Most of the existing works focus on the domain of online dating.

### 4.3.2 Properties in Reciprocal Recommenders

**Reciprocity:** In traditional user×item recommender systems, only unilateral preferences are considered, i.e., the users’ preferences on items. However, in the domain of job matching, both users being involved in the recommendation have their preferences against with each other; in such a situation, only considering unilateral preference might not be reasonable. In other words, the success of a match depends on the bilateral preference, but not solely on the user who receives the recommendation. This is the key feature of a reciprocal recommender that differentiates it from the traditional user×item recommendation paradigm.

In our system, job seekers and recruiters have their self-descriptive information, and also the preferences on either job positions or applicants. Given a target user  $u$  and a search result list  $Q_u$ , we are interested in finding a relevant user  $v \in Q_u$ , such that  $(u, v)$  is a successful match. By *relevance*, we mean that the self-description of user  $v$ ,  $p_v^s$ , matches the preference profile of user  $u$ ,  $p_u^p$ , and at the meantime, the self-description of user  $u$ ,  $p_u^s$ , matches the preference of user  $v$ ,  $p_v^p$ . Therefore, the *relevance* includes two components,  $rel(u \sim v)$  and  $rel(v \sim u)$ . Here the relevance is calculated using the cosine similarity between the vectors. We then formalize the relevance between  $u$  and  $v$  as  $rel(u, v) = rel(u \sim v) \cdot rel(v \sim u)$ .

**Availability:** In traditional recommenders, an item can be preferred by a great amount of users, e.g., a music album by the musician Michael Jackson. However, in a job matching system, people have limited availability towards other peo-

ple, e.g., a job seeker cannot have 100 interviews with different companies simultaneously. Therefore, when designing the recommendation strategy for job matching systems, we need to consider the availability of users so that all the users can obtain reasonable recommendation results.

In iHR, we intentionally record the number of times that a user has been recommended to other users. If a user has been recommended to other users so many times in a time range, e.g. a week, we regard the availability of this user as low; similarly, if a user is a new registered user, then this user has higher availability. Particularly, for job seekers and recruiters, we set different thresholds to indicate the extent of their availability. For example, the threshold for job seekers is set to be 20, i.e., if the number of times that a job seeker is recommended to other users exceeds 20 in a time range, we will not recommend this job seeker any more. Similarly, the threshold for recruiters is set to be 50.

**Diversity:** In general, the search result contains a lot of records, some of which might be similar in terms of specific features. For example, two job candidates may have the same GPA and similar education background. If we recommend both of them to a recruiter, the recruiter may spend time and other resources to distinguish which candidate is better. In our system, we try to go beyond such established paradigm. Instead, we provide users diverse recommendation results to help them efficiently classify the candidates. Here “diverse” means that the candidates in the recommended result might exhibit different personal strengths.

### 4.3.3 Recommendation Methodology

In job matching system, when a user, e.g., a job seeker, searches job positions, he/she might have more preference on the top ranked results. When the user scrolls down the browser and clicks on the job posts, he/she might lose patience to view the details of the result. In other words, the interestingness of job posts with respect to a job seeker could be regressive (the situation also holds when a recruiter searches job candidates), which is known as the “submodularity”. In this recommendation module, we model the recommendation problem as a budgeted maximum coverage problem [9], and incorporate the special properties introduced in §4.3.2 into the solution.

**Introduction to Submodularity:** Let  $E$  be a finite set and  $f$  be a real valued nondecreasing function defined on the subsets of  $E$  that satisfies

$$f(T \cup \{\zeta\}) - f(T) \leq f(S \cup \{\zeta\}) - f(S), \quad (3)$$

where  $S \subseteq T$ ,  $S$  and  $T$  are two subsets of  $E$ , and  $\zeta \in E \setminus T$ . Then  $f$  is called a **submodular** function [21]. By adding one element to a larger set  $T$ , the value increment of  $f$  can never be larger than that by adding one element to a smaller set  $S$ . Submodularity modeling has been employed into multiple research areas, e.g., document summarization [14, 18], news recommendation [15], graph mining [30], etc.

The budgeted maximum coverage problem is then described as: given a set of elements  $E$  where each element is associated with an influence and a cost defined over a domain of these elements and a budget  $B$ , the goal is to find out a subset of  $E$  which has the largest possible influence while the total cost does not exceed  $B$ . This problem is NP-hard [9]. However, [9] proposed a greedy algorithm which picks up the element that increases the largest possible influence within the cost limit each time and guarantees

the influence of the result subset is  $(1 - 1/e)$ -approximation. Submodularity resides in each “pick up” step. A key observation is that submodular functions are closed under non-negative linear combinations [13].

**Submodularity Model:** In our recommendation model, we consider the properties introduced in §4.3.2, i.e., *Reciprocity*, *Availability* and *Diversity*. Given a target user  $u$  and a search result  $\mathcal{Q}$ , we try to sequentially select items from  $\mathcal{Q}$  and then put them into a new set  $\mathcal{S}$ . The selection strategy can be described as follows (note that  $\zeta$  is the item being selected). After selection  $\zeta$ , we expect that (1)  $\mathcal{S}$  should provide more relevance to the target user  $u$ ; and (2) The diversity in  $\mathcal{S}$  should not deviate too much. Per the above strategy, we define a function  $f$  to measure the quality of the current selected set  $\mathcal{S}$  against  $\mathcal{Q}$  as

$$f(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{v_1 \in \mathcal{S}} \frac{1}{\tau} \cdot rel(u, v_1) + \frac{1}{\binom{|\mathcal{S}|}{2}} \sum_{\substack{v_1, v_2 \in \mathcal{S} \\ v_1 \neq v_2}} -sim(v_1, v_2), \quad (4)$$

where  $v_1$  and  $v_2$  denote users in  $\mathcal{Q}$ ,  $\tau$  indicates the availability of the user  $v_1$ ,  $rel(\cdot, \cdot)$  represents the relevance between two users (e.g., a job seeker and recruiter), and  $sim(\cdot, \cdot)$  denotes the similarity between two users (e.g., two job seekers).

In Eq.(4), two components are involved corresponding to the user selection strategy listed above. The former aims to evaluate how relevant that the selected user set  $\mathcal{S}$  is to the target user  $u$ , whereas the latter gives us the evidence that how diverse the selected set  $\mathcal{S}$  is. Note that we use  $\tau$  to indicate the availability of a user, i.e., the number of times that the user has been recommended to other users. We take the reverse of  $\tau$  to reduce the possibility of the user being selected into  $\mathcal{S}$ .  $f(\mathcal{S})$  balances the contribution of different components, and clearly the two components are naturally submodular functions. Based on the non-negative linear invariability of the submodularity function [13],  $f(\mathcal{S})$  is also a submodular function.

Suppose  $\zeta$  is the candidate user, the quality increase is therefore represented as

$$I(\zeta) = f(\mathcal{S} \cup \{\zeta\}) - f(\mathcal{S}). \quad (5)$$

The goal is to select a list of users from  $\mathcal{Q}$  with the largest possible quality increase under the budget. A greedy algorithm is employed to solve this problem. Note that in iHR, the budget is set to 100, i.e., to provide users at most 100 candidates by refining the search result.

### 4.3.4 Recommendation Fusion

The aforementioned recommendation methods capture different aspects of the relevant results. In our system, we provide a recommendation fusion strategy to integrate the recommended results. Specifically, different weights are assigned to the ranking scores of results obtained from the three methods. Formally, let  $r^{cot}$ ,  $r^{cof}$  and  $r^{rep}$  denote the ranking scores from content filtering, collaborative filtering and reciprocal filtering, our recommendation fusion model towards selecting an item  $i$  can be described as

$$r_i = \alpha \times r_i^{cot} + \beta \times r_i^{cof} + \gamma \times r_i^{rep}, \quad (6)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  represent the weights of the corresponding scores, s.t.  $\alpha + \beta + \gamma = 1$ . Different weighting schemes of  $\alpha$ ,  $\beta$  and  $\gamma$  characterize different information needs of users:

- If  $\alpha$  dominates, it indicates that a user has relatively clear preference on the result, and therefore the recommendation will be primarily based on this user's preference profile  $p^p$ ;
- If  $\beta$  dominates, it indicates that a user does not have definite requirement on what information should be provided, and therefore the recommendation can be performed based on similar users' preferences;
- If  $\gamma$  dominates, it indicates that a user is concerned with the status of the recommended result, e.g., a job seeker will consider the availability of a position. The recommendation is achieved by considering the "reciprocal" property within a job matching community.



Figure 4: An illustrative example of the recommendation fusion model.

In our current system, the three scores are treated equally. For our future work, the parameters,  $\alpha$ ,  $\beta$  and  $\gamma$ , will be tuned incrementally as the system collects more information about users. Figure 4 shows some recommended results to a job seeker who is looking for “Manager” positions. The left part of the figure describes the scores of the recommended job positions in a scale of 1-5.

## 5. DATA COLLECTION AND EVALUATION

Up to now, iHR has been deployed online for practical use, with over 699,600 visits per day. In our system, we propose a reciprocal recommendation method that emphasizes the bilateral correlations between job seekers and recruiters.

### 5.1 Empirical Evaluation

We perform quantitative evaluation on our proposed reciprocal recommendation strategy. The data used for experiments is a sampled data set collected from iHR, including the profiles and activities for users from Jan, 2008 to Oct, 2011. We transform the features using the strategy described in §3.1, and then calculate the user relevance based on the new feature space. The data statistics is depicted in Table 4.

Table 4: Statistics of the dataset.

Basic Statistics		training	testing
Job Seekers	199,999	176,423	23,576
Recruiters	46,629	29,850	6,779
# of $p_u^s$	860	—	—
# of $p_u^p$	928	—	—
# of $p_u^r$	928	—	—
# of $p_v^p$	860	—	—
Activities	664,943	493,128	171,815

### 5.1.1 Experiment Setup

For experiments, we split the data set into training and testing sets. Each set includes two sets of users, associated with their interactive activities, as shown in Table 4. For each user in the testing set, we recommend top ranked users (top@10, top@20 and top@30) at each week of the testing range using different strategies. Within the testing set, each user has a series of activities, e.g., adding job positions as favorite. Based on these activities, we use different metrics to evaluate the quality of the recommended list.

*Set Evaluation:* For comparison, we compute the averaged precision and recall based on users' activities. Specifically, the ground truth of a user  $u$ 's activities, including who have been clicked or contacted by  $u$ , is denoted by  $M$ , and the recommended user list by algorithms is denoted by  $N$ . Then the precision ( $P$ ) and recall ( $R$ ) can be computed as

$$P = \frac{M \cap N}{N}, \quad R = \frac{M \cap N}{M}. \quad (7)$$

We then compute the  $F_1$ -score of the recommendation results, i.e.,  $F_1 = \frac{2PR}{P+R}$ .

*Ranking Evaluation:* We employ Normalized Discount Cumulative Gain (NDCG) to evaluate the ranking quality of the recommended list based on a user's actual activity sequence. NDCG at position  $n$  is defined as

$$NDCG@n = N(n) \times \sum_{i=1}^n \frac{2^{r_i} - 1}{\log_2(i + 1)}, \quad (8)$$

where  $N(n)$  is the NDCG at  $n$  of the ideal ranking list, and  $r_i$  is the relevance rating of item at rank  $i$ . In our scenario,  $r_i = 1$  if the user has clicked on or contacted with the recommended users and 0 otherwise.

### 5.1.2 Comparison with Other Methods

Here we only compare the reciprocal strategy in iHR with other existing methods. We choose two recently published collaborative filtering methods [8, 17] as our baselines. [8] (CFIF for short) proposed treating the data as indication of positive and negative preference associated with vastly varying confidence levels, which is a pure collaborative filtering approach. [17] (OCCF for short) exploited the rich user information available in community-based interactive information systems, and incorporated user information into modeling the recommendation. For this method, we use the neighborhood model as the baseline. We also implement GBDTs [6], RECON [23] and CCR [2] for comparison. We use  $F_1$ -score and NDCG to compare these algorithms with iHR. The feature set used in all the methods are identical to the one in our proposed method, and also the parameters are optimally tuned.

Table 5: Comparison with existing methods. (Bold indicates the best performance. \* indicates the statistical significance at  $p < 0.01$ .)

Methods	top@10		top@20		top@30	
	$F_1$	NDCG	$F_1$	NDCG	$F_1$	NDCG
CFIF	0.2301	0.3174	0.3121	0.3813	0.3481	0.4036
OCCF	0.2485	0.3320	0.3219	0.3929	0.3569	0.4127
GBDTs	0.2567	0.3592	0.3304	0.4131	0.3718	0.4432
RECON	0.2604	0.3608	0.3247	0.4025	0.3839	0.4507
CCR	0.2431	0.3745	<b>0.3573</b>	0.3987	0.3912	0.4729
iHR	<b>0.2718*</b>	<b>0.3720</b>	0.3501	<b>0.4316*</b>	<b>0.4098*</b>	<b>0.4875*</b>

The results are shown in Table 5. It is evident that *iHR* significantly outperforms the baselines on both  $F_1$ -score and NDCG. The two collaborative filtering based methods cannot effectively handle the reciprocal task. We investigated the recommendation results of both methods and found that users in most recommended matches are relevant. However, a significant reason that both users in a match have *few or even no interactions* is that *the recommended user has been recommended to multiple users, and therefore he/she has limited availability*. The three reciprocal methods being compared can slightly improve the recommendation performance; however, they only focus on different aspects of the reciprocal community. Instead, *iHR* with reciprocal recommendation provides a comprehensive overview of the reciprocal network, and therefore achieves the best.

## 5.2 A User Study

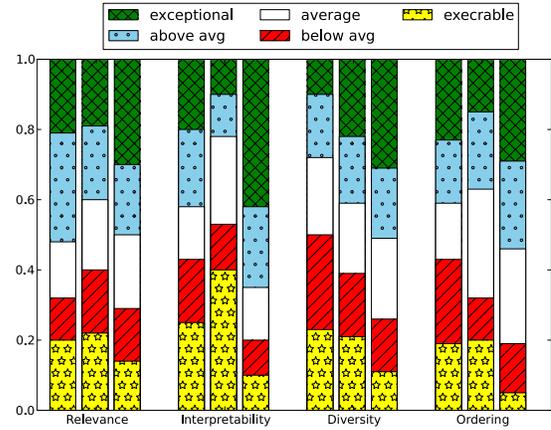
In order to evaluate the efficacy of *iHR*, we present a survey to each valid user to collect user experiences. The purpose of the survey is to evaluate how users feel about the results generated by different algorithms. These three methods are parallel in the system, i.e., users have the choice to check different recommendation results from different algorithms. The survey covers several aspects for evaluation, including *relevance*, *interpretability*, *diversity* and *ordering*. Sample questionnaire statements [26] are listed in Table 6.

**Table 6: Sample questionnaire statements used in our survey. (Remark: The scale is 1-5. 1 – worst, 5 – best. Reverse scale: 1 – best, 5 – worst.)**

Aspects	Statements
Relevance	<ul style="list-style-type: none"> <li>The items in the list matched my interests.</li> <li>The recommender gave me good suggestions.</li> <li>I am not interested in the items recommended to me (reverse scale).</li> </ul>
Interpretability	<ul style="list-style-type: none"> <li>The recommender explains why the candidates (recruiters) are recommended to me.</li> <li>The recommender shows me details to help me digest the recommended results.</li> </ul>
Diversity	<ul style="list-style-type: none"> <li>The items recommended to me are diverse.</li> <li>The items recommended to me are similar to each other (reverse scale).</li> </ul>
Ordering	<ul style="list-style-type: none"> <li>The recommended results maintain a relatively reasonable ordering.</li> <li>The candidates (recruiters) that perfectly match my preference are listed at the top of the result.</li> </ul>

Based on these aspects, we define the corresponding indices to measure the satisfaction of online users (i.e., job seekers and recruiters). Each experience index is rated by users in a range of 1 to 5, where 1 – “Execrable”, 2 – “Below Average”, 3 – “Average”, 4 – “Above Average”, and 5 – “Exceptional”. We collect users’ feedbacks on these experience indices from October 2011 to January 2012. At the end of the evaluation period, we have obtained over 500,000 valid feedbacks from users. To analyze the experience result, we calculate the percentages of users with different ratings on the indices, and then plot them in Figure 5.

From the result, we observe that the reciprocal recommendation method outperforms the other two methods in terms of user experience. Particularly, for the “Relevance” index, over 50% of users vote the reciprocal method as above average, thanks to the paradigm that considers the mutual



**Figure 5: User experience results on different experience indices. For each index, Bar1 represents the results of content filtering, Bar2 shows the results of collaborative filtering, and Bar3 indicates the results of reciprocal recommendation.**

relevance between job seekers and recruiters. For the “Interpretability” index, over 65% of users regard the recommendation result more interpretable, since we explicitly present the bilateral relations between job seekers and recruiters, which renders the result more explainable. For the “Diversity” and “Ordering” indices, over 51% and 54% of users vote the reciprocal method as above average. The underlying reason is that we elaborately design the selection strategy by considering the properties of job matching systems.

## 6. CONCLUSION

Xiamen Talent Service Center has spent over \$300K in the development, deployment and maintenance of *iHR*, and has received over \$150K in sponsored government funding. The system is maintained 24/7 to ensure the smooth running of database, web server and hardware. The system is used by over 1,770,000 users, including job seekers (1,700,000) and enterprises (70,000) from local and adjacent job marketing communities. *iHR* facilitates the booming of the careers fair, and therefore stimulates healthy economic development in the city of Xiamen. *With the development over five years, iHR has been selected as one of the most prominent and influential online recruiting platforms in China.*

In *iHR*, we employ pragmatic techniques in data mining and recommendation to help strengthen the system experience. To effectively construct the users’ profiles, we synthesize information from three different channels to maximally discover and understand users’ preferences. To satisfy different information needs, we integrate search and recommendation into a mutually promoted framework. To enhance the recommendation for more effective information refining, we propose a comprehensive strategy to simultaneously consider the appetite of job seekers and recruiters. The potential impact of this research on the development of job market community is enormous.

Due to the expensive maintenance of the system, in practice it is very difficult to evaluate the effect of each component in our running system, such as the profile construction (using users’ demographic information, extracted informa-

tion and behavioral information for profile fusion) and the individual characteristic of reciprocal recommendation (including reciprocity, availability and diversity). In the future, we plan to perform comprehensive evaluation on these aspects to learn how well each component works in our system.

## ACKNOWLEDGEMENTS

The project was supported by the Natural Science Foundation of Fujian Province of China under Grant No. 2011J05157 and by the National Natural Science Foundation of China under Grant No. 61070151. The work of Lei Li and Tao Li was supported in part by the US National Science Foundation under Grant DBI-0850203, HRD-0833093, DMS-0915110, and CNS-1126619, and by the Army Research Office under Grant W911NF-12-1-0431.

## 7. REFERENCES

- [1] J.T. Addison and P. Portugal. Job search methods and outcomes. *Oxford Economic Papers*, 54(3):505, 2002.
- [2] J. Akehurst, I. Koprinska, K. Yacef, L. Pizzato, J. Kay, and T. Rej. Ccr-a content-collaborative reciprocal recommender for online dating. In *Proc. of IJCAI*, pages 2199–2204. AAAI, 2011.
- [3] L. Brozovsky and V. Petricek. Recommender system for online dating service. *Arxiv preprint*, 2007.
- [4] X. Cai, M. Bain, A. Krzywicki, W. Wobcke, Y. Kim, P. Compton, and A. Mahidadia. Collaborative filtering for people to people recommendation in social networks. *Advances in Artificial Intelligence*, pages 476–485, 2011.
- [5] Chen-Fu Chien and Li-Fei Chen. Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry. *Expert Systems with Applications*, 34(1):280–290, 2008.
- [6] F. Diaz, D. Metzler, and S. Amer-Yahia. Relevance and ranking in online dating systems. In *Proc. of SIGIR*, pages 66–73. ACM, 2010.
- [7] Frank Färber, Tim Weitzel, and Tobias Keim. An automated recommendation approach to selection in personnel recruitment. In *Proceedings of the Ninth Americas Conference on Information Systems*, pages 4–6, 2003.
- [8] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proc. of ICDM*, pages 263–272. IEEE, 2008.
- [9] S. Khuller, A. Moss, and J.S. Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.
- [10] Y. Kim, A. Mahidadia, P. Compton, X. Cai, M. Bain, A. Krzywicki, and W. Wobcke. People recommendation based on aggregated bidirectional intentions in social network site. *Knowledge Management and Acquisition for Smart Systems and Services*, pages 247–260, 2011.
- [11] D.H. Lee and P. Brusilovsky. Fighting information overflow with personalized comprehensive information access: A proactive job recommender. In *Proc. of ICAS*, pages 21–26. IEEE, 2007.
- [12] I. Lee. An architecture for a next-generation holistic e-recruiting system. *Communications of the ACM*, 50(7):81–85, 2007.
- [13] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proc. of SIGKDD*, pages 420–429. ACM, 2007.
- [14] J. Li, L. Li, and T. Li. Mssf: a multi-document summarization framework based on submodularity. In *Proc. of SIGIR*, pages 1247–1248. ACM, 2011.
- [15] L. Li, D. Wang, T. Li, D. Knox, and B. Padmanabhan. Scene: a scalable two-stage personalized news recommendation system. In *Proc. of SIGIR*, pages 125–134. ACM, 2011.
- [16] L. Li and T. Li. Meet: a generalized framework for reciprocal recommender systems. In *Proc. of CIKM*, pages 35–44. ACM, 2012.
- [17] Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proc. of CIKM*, pages 959–968. ACM, 2010.
- [18] H. Lin and J. Bilmes. Multi-document summarization via budgeted maximization of submodular functions. In *Proc. of NAACL-HLT*, pages 912–920. Association for Computational Linguistics, 2010.
- [19] J. Malinowski, T. Keim, O. Wendt, and T. Weitzel. Matching people and jobs: A bilateral recommendation approach. In *Proc. of HICSS*, pages 137–145. IEEE, 2006.
- [20] C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press, 2008.
- [21] GL Nemhauser, LA Wolsey, and ML Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.
- [22] Ioannis Paparrizos, B Barla Cambazoglu, and Aristides Gionis. Machine learned job recommendation. In *Proc. of RecSys*, pages 325–328. ACM, 2011.
- [23] L. Pizzato, T. Rej, T. Chung, I. Koprinska, and J. Kay. Recon: a reciprocal recommender for online dating. In *Proc. of RecSys*, pages 207–214. ACM, 2010.
- [24] L. Pizzato, T. Rej, K. Yacef, I. Koprinska, and J. Kay. Finding someone you will like and who won't reject you. *User Modeling, Adaption and Personalization*, pages 269–280, 2011.
- [25] L.A. Pizzato and C. Silvestrini. Stochastic matching and collaborative filtering to recommend people to people. In *Proc. of RecSys*, pages 341–344. ACM, 2011.
- [26] P. Pu, L. Chen, and R. Hu. A user-centric evaluation framework for recommender systems. In *Proc. of RecSys*, pages 157–164. ACM, 2011.
- [27] R. Rafter, K. Bradley, and B. Smyth. Automated collaborative filtering applications for online recruitment services. In *Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 363–368, 2000.
- [28] B. Smyth, K. Bradley, and R. Rafter. Personalization techniques for online recruitment services. *Communications of the ACM*, 45(5):39–40, 2002.
- [29] L. Terveen and D.W. McDonald. Social matching: A framework and research agenda. *ACM Transactions on Computer-Human Interaction*, 12(3):401–434, 2005.
- [30] Hanghang Tong, Jingrui He, Zhen Wen, Ravi Konuru, and Ching-Yung Lin. Diversified ranking on large graphs: an optimization viewpoint. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1028–1036. ACM, 2011.
- [31] Xing Yi, James Allan, and W Bruce Croft. Matching resumes and jobs based on relevance models. In *Proc. of SIGIR*, pages 809–810. ACM, 2007.
- [32] K. Yu, G. Guan, and M. Zhou. Resume information extraction with cascaded hybrid model. In *Proc. of ACL*, pages 499–506. ACL, 2005.