Efficient Multifaceted Screening of Job Applicants*

Sameep Mehta†
IBM Research India
New Delhi, India

Rakesh Pimplikar
IBM Research India
New Delhi, India

Amit Singh
IBM Research India
Bangalore, India

Lav R. Varshney‡
IBM Watson Research Center
Yorktown Heights, New York, USA

Karthik Visweswariah
IBM Research India
Bangalore, India

ABSTRACT

Built on top of human resources management databases within the enterprise, we present a decision support system for managing and optimizing screening activities during the hiring process in a large organization. The basic idea is to prioritize the efforts of human resource practitioners to focus on candidates that are likely of high quality, that are likely to accept a job offer if made one, and that are likely to remain with the organization for the long term. To do so, the system first individually ranks candidates along several dimensions using a keyword matching algorithm and several bipartite ranking algorithms with univariate loss trained on historical actions. Next, individual rankings are aggregated to derive a single list that is presented to the recruitment team through an interactive portal. The portal supports multiple filters that facilitate effective identification of candidates. We demonstrate the usefulness of our system on data collected from a large organization over several years with business value metrics showing greater hiring yield with less interviews. Similarly, using historical pre-hire data we demonstrate accurate identification of candidates that will have quickly left the organization. The system has been deployed as described in a large globally integrated enterprise.

1. INTRODUCTION

Top talent is a key differentiator for any organization [6], but especially so for complicated knowledge work where performance variances are large [21]. Consequently it is critical for the talent supply chain to be functioning well. In this paper we focus on the problem of recruitment and describe a decision support system to prioritize the efforts of human resource (HR) practitioners by using data mining to automatically screen and rank candidates.

By extending current enterprise database systems with novel analytics and presentation layers, we have developed and deployed an advanced screening system which aids in identifying good candidates for hiring in a large organization.

Large organizations often receive tens of thousands of job applications each month and manual analysis of such large applicant volumes is cumbersome, error-prone, and costly. In using a sequential selection procedure, manual screening may select for interview the first few candidates that match the job requirements, thereby missing potentially better candidates. This fairly ad-hoc process results in a suboptimal pool of potential employees.

Moreover the recruitment process is very costly. The traditional approach first requires HR practitioners to perform resume screening, then a technical team to perform interviews, and finally a further team to perform background checks. An operations team is also needed to manage logistics for interviews. Once a candidate clears screening, interview(s), and a background check, she is offered a job. However, a large fraction of candidates decline job offers (or in certain geographies, accept offers but fail to report to work). When an offer is declined, recruitment cost and effort is for naught.

It can further be noted that if an employee joins the organization but then leaves after only a small period of time, backfilling the position requires the recruitment process anew. There are further costs to attrition, such as training, ramp-up time (often a low productivity period), and understaffing while undergoing backfilling. In certain markets, organizations pay fees to agencies that provide candidate resumes or pay rewards to employees that provide candidate resumes through employee refer-

*Authors are listed alphabetically.

†{sameepmehta, rakesh.pimplikar, amitkumarsingh, v-karthik}@in.ibm.com

‡lrvarshn@us.ibm.com

EDBT/ICDT '13 March 18 - 22 2013, Genoa, Italy

ral programs. These fees are also lost when employees leave organizations.

It is evident that recruiting candidates that are well-matched to the job, that are of high quality, that are likely to accept offers, and that are likely to stay for the long-term is critical not only for cost savings but also for the continuous well being of an organization. With the information overload caused by the large number of applicants in large organizations, however, the use of automatic decision support through data mining and business analytics [3, 17, 8] is necessary. Fully automatic analytics-only talent management methods, however, have not yet been found useful in any organization [16], and so we do not consider this further step of automation.

Goodness of candidates is defined along a number of dimensions including high fitness for the open job (technical match); high probability of being of good quality (quality likelihood); high probability of joining the organization (onboard likelihood); and low risk of leaving the organization (attrition likelihood). Our system analyzes resumes and extracts the relevant features from the resume using text extraction and text mining techniques. Technical match is computed by comparing a candidate's skill set with the skills required for a job. Computation of the several likelihoods are posed as bipartite ranking problems with minimization of univariate loss functions [2, 13, 23]. Since many resume features are categorical, we use random forest methods [7] for the likelihood tasks.

Finally, the several individual rankings are combined to derive one single holistic ranking through a hierarchical rank aggregation procedure. As far as we know, our hierarchical nested structure of ranks for aggregation is novel, cf. [18]. The ranked list along with other candidate information is shown to the recruitment team through an online portal, which also supports a variety of filters.

The system was validated on a database of applicants to a large organization with an excess of $100,000$ candidate records, and has now been deployed to human resource practitioners in the same large organization. Besides data mining accuracy, the benefits of the system have been captured using several business metrics such as decrease in human efforts for screening; reduction in number of interviews required; increase in onboard yield; and reduced (predicted) attrition.

In earlier work, our group had presented a system Prospect that computes technical match given a resume and job description [30]. For completeness, we summarize the basic ideas of Prospect in this paper however, our focus is on the ranking algorithms for other tasks in the recruitment cycle and rank aggregation that extend basic database technologies to provide insights to human resources practitioners in a consumable way.

The remainder of the paper is organized as follows. The next section compares our work with related work in candidate screening. Section 3 provides an overview of the general recruitment process. Section 4 describes the basic data mining techniques we use to build the several value likelihood ranking algorithms. Section 5 describes some experiments that assess system performance. We conclude by recapitulating the experiences of human screeners that have used the system, which may guide future work.

2. RELATED WORK

Computation of technical match between candidates and jobs using text analytics has been developed, e.g. in a system called Prospect [30]. This decision support tool aids in shortlisting of candidates for jobs. The key challenge addressed in this work was that job requirements and resumes are written in natural language and further job requirements often contain complex constraints (e.g. "at least 6 years of J2EE experience"). Prospect addresses this challenge by mining unstructured resumes to extract salient aspects of candidate profiles like skills, experience in each skill, education details and past experience. Extracted information is presented in the form of facets to aid recruiters in screening, overcoming limitations inherent in purely keyword-based matching.

There have been several other attempts to automate various aspects of the recruitment process [26]. For example, [25] suggests using techniques like collaborative filtering to recommend candidates matching a job, whereas [34] describes a method that uses relevance models to bridge the vocabulary divide between job descriptions and resumes. In [19, 22], collaborative filtering measures are combined with content based similarity measures for better candidate ranking. There is also a separate body of work that takes into account individual preferences and team dynamics for staffing [27, 12]. However, most of these studies are performed on synthetic data and not on real unstructured resumes and job descriptions.

There has also been extensive work in the applied psychology literature on factors that attract candidates to jobs and to organizations [11], as well as the decision-making process in whether to accept a job offer [31]. However the goal in these works was not to compute onboard likelihoods.

Besides candidate screening and job choice, the large impact of candidate attrition on the functioning of organizations has inspired researchers and practitioners alike to identify and model the factors that enable organizations to effectively manage employee turnover [32, 15, 9, 28]. These models are used for predicting turnover likelihood once the candidate has joined the organization, rather than before hiring.

In contrast to the previous works, here we describe a system that ranks the candidates not only based on their quality and fitness for the job but also other factors such as likelihood of joining the organization and risk of leaving the organization, all at the pre-hire stage of the human resource lifecycle.

3. SYSTEM OVERVIEW

Following classical work in human resource management [6], Figure 1 depicts the key steps in the human resource lifecycle, focusing on the recruitment process. The process starts with obtaining candidate applica-

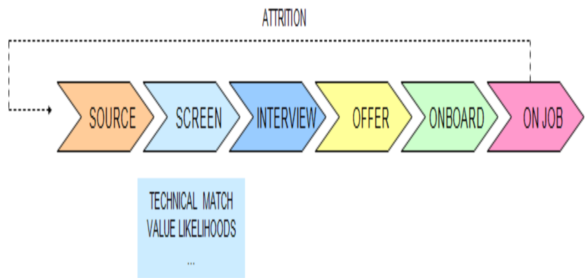

Figure 1: Overview of the recruitment process with the proposed decision support system indicated.

tions from various sources such as employee referral, direct application, or referrals from third parties. These applications are for specific job requisitions and include resumes as well as structured data. In our system, relational databases are used to store the structured and unstructured data that is provided by a given candidate for a given job opening.

These applications are to be screened to determine candidates to advance to the costly interview stage; traditionally this would be done manually but our decision support system is used to make this step more principled and efficient while simultaneously ensuring that better candidates pass through to future stages in the process. Candidates that pass the interview stage are offered a position with a specified compensation. Candidates may accept, decline, or further negotiate an offer. Candidates that accept an offer come onboard and start work. Finally, employees may be tracked through performance evaluation processes. Attrition creates openings that require the whole recruitment process to be followed again.

Each step in the recruitment process provides an opportunity for data analytics and optimization. For example, selecting interviewers for a given candidate based on matches between skills, job roles, location, (and personal and organizational factors [11]) is a challenging optimization problem. In this paper we focus on screening. Traditional screening deals only with technical match [30], but here we also consider several other factors such as likelihoods of offer acceptance and of attrition. The basic algorithms we use are described in the sequel.

Before proceeding, let us discuss how an oracle—one that could rank perfectly along the various dimensions such as quality, onboard likelihood, and attrition likelihood—would be used in business practice. Note however that oracles do not exist even in the infinite big data regime, since even a Bayes-optimal set of algorithms have inherent uncertainty.

The first step is to determine the number of people that are to be hired for a given job requisition, based on labor needs. The second step is to determine the costs of interviewing and of attrition, as well as the value of higher quality employees; these three numbers are just measured in monetary terms.

With access to the monetary figures, the expected utility derived from the number of interviews until reaching the desired threshold of accepted offers can be balanced against the value of high-quality candidates adjusted by how long they stay with the organization. This leads to a weighting on the importance of quality likelihood, onboard likelihood, and attrition likelihood, cf. Table 1 for a typical weighting. This weighting can then be used to determine an appropriate rank aggregation methodology, see Section 4.3.

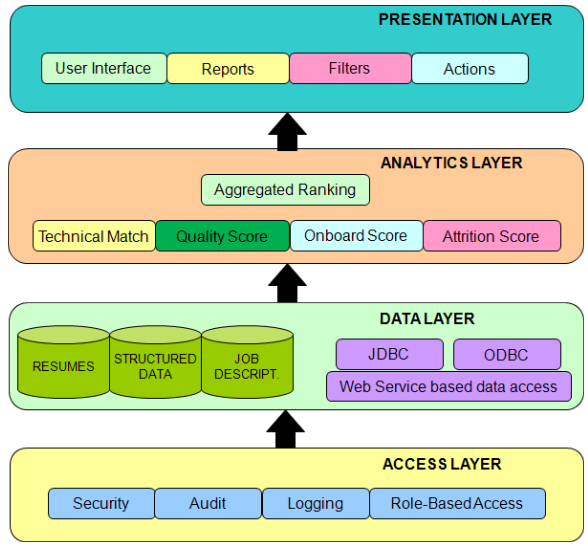

Figure 2: Conceptual architecture of an implemented recruitment analytics decision support system.

Changing labor markets and business imperatives lead to changing relative costs/benefits in rank aggregation. The base individual ranking algorithms, however, should always be the best possible.

This understanding of how to use an oracle has formed the basis for the actual system design that we have developed. Several data management principles are also required to develop the system architecture. Figure 2 presents a simplified version of the conceptual architecture in our implemented system. Input include candidate resumes and structured data as well as associated job descriptions. The analytics layer implements the various data mining algorithms including the text-based technical match (built using Lucene), the bipartite ranking-based likelihood computations, and the rank aggregation. The aggregated ranking is presented to the HR practitioner through an interactive user interface portal. The portal supports role-based access protocols so that HR practitioners and interviewers can only view and act on applications for which they are authorized.

4. DATA MINING TECHNIQUES

The keystone of the recruitment analytics system is the analytics layer with its data mining algorithms. As part of data management, we must ensure that our system is free from discrimination based on race, color, religion, gender, gender identity or expression, sexual orientation, national origin, genetics, disability, age, or other factors unrelated to legitimate business interests [1]. This precludes the use of many 'biodata' features that others have explored [20], cf. [4]. Notwithstanding, we make use of many typical features from resumes and structured candidate data, such as number of years of experience, highest degree level, and list of technology skills.[1] Technical match is also governed by the skill requirements listed in job descriptions.

4.1 Notation and Basic Subsystems

Before discussing the data mining algorithms, let us first introduce some mathematical notations. Let J represents the list of all open positions, where $J^i = \{J_1^i, J_2^i, \ldots, J_k^i\}$ represents the i^{th} job requisition and its vector elements capture its requirements such as skills and years of experience. Similarly, let the database of candidate applications/resumes be represented by A such that $A^j = \{A_1^j, A_2^j, \ldots, A_n^j\}$, where A^j represent information about the jth candidate and the vector elements capture features from the structured candidate data and features extracted from unstructured resumes (e.g. from a text analysis system [30]).

Further, assume that $|J^i| = K$ for all i and $|A| = N$ for all j. Let $\mathcal{Z}$ be a subset of the Cartesian product of J and A, where every record $z_i \in \mathcal{Z}$ represents an application of a candidate x_u for the job she has applied for p_v. Besides the job and candidate features, a record z_i may also include application status such as how far along in the hiring lifecycle (Figure 1) a given application is in or whether it failed in a given step. Notation that operates on this status variable is as follows. By example, we use $\mathcal{Z}_{\{\text{accept-offer},\text{attrit}\}}$ to indicate all records in $\mathcal{Z}$ such that the state is either 'accept-offer' or 'attrit.'

Next we briefly present what the several algorithms depicted in the analytics layer of Figure 2 are meant to do. We return to algorithm details afterward.

Technical Match.

We denote the technical match ranking of the N applications as T determined using the individual normalized scores t_i for each application. This score t_i is determined using text fields (such as skills) extracted from the unstructured resume of the candidate by matching to the job description and computing the quality of match. We use the technical match score produced by PROSPECT [30].

Quality.

Technical match only performs skill matching and essentially only ensures that the candidate pool considered for a given job requisition is on target. However, features in the application may actually be indicative of whether the candidate is generally competent. For example, the university attended by the candidate or his/her past employer may cue a human screener or an automatic ranking system to the quality of a candidate. Using a historical dataset of what human screeners have decided to do, we use a bipartite ranking algorithm to learn a quality ranking function. That is, we use a historical set $\mathcal{Z}_{\{\text{pass-screen}\}}$ as positive examples and a historical set $\mathcal{Z}_{\{\text{fail-screen}\}}$ as negative examples in learning. In highly-selective organizations, training data is imbalanced with many more negative samples than positive samples: this needs to be considered in model training.

When applied on a new set of applications, we denote the resultant ranking as Q, determined using the individual quality likelihood scores q_i for each application. Note that training is performed on a historical dataset containing different people than the unseen new applicants; hence there is no issue of cold-start as faced in personalized recommendation systems [29].

Onboard.

Computing the likelihood of a candidate accepting an offer and coming onboard to the organization rather than declining an offer is also formulated as a bipartite ranking problem: again there is historical binary training data available on candidates that accepted offers and candidates that declined offers. Many of the same features that are predictive of quality are also predictive of onboard likelihood, but additional features available at the screening stage such as tentative title and salary for the job requisition are also used. Note that salaries and titles may be negotiated at later stages in the hiring process, however tentative values or ranges are often fixed early on.

Here the historical training set of positive examples is $\mathcal{Z}_{\{\text{onboard}\}}$ and the historical set of negative examples is $\mathcal{Z}_{\{\text{decline-offer}\}}$. In highly-desirable organizations, training data may be imbalanced with more onboards than offer declines. As before, this should be considered in model training.

When applied on a new set of applications, we denote the resultant ranking as O, determined using the individual onboard likelihood scores o_i for each application. Note that although this ranking can be applied broadly, it is only valid on candidates that are of suitable match and quality to warrant an offer.

Attrition.

Like the two previous ranking algorithms, attrition likelihood can also be scored using a bipartite ranking algorithm trained on historical data. The features that are used for attrition likelihood ranking are pretty much the only ones that are available at the screening stage of the recruitment lifecycle and so they are used. The historical training set of positive examples is $\mathcal{Z}_{\{\text{long-tenure}\}}$ and the historical set of negative examples is $\mathcal{Z}_{\{\text{attrit}\}}$.

When applied on a new set of applications, we denote the resultant ranking as V, determined using the individual attrition likelihood scores v_i for each application. Note that although this ranking can be applied broadly, it is only valid on candidates that are of suitable match and quality to warrant an offer and that are actually very likely to have come onboard.

[1] Due to reasons of business confidentiality and to mitigate the need for repeated adaptations and counter-adaptations in the signaling game of personnel selection that arises when selection criteria are fully specified [5], we do not list the precise features that we use.

Optimal Rank Aggregation.

A single rank must be developed for consumption by HR practitioners. Although a single optimization goal could be established *a priori* rendering the need for the four individual ranks described above unnecessary, business needs and labor markets can quickly shift. Hence it is much more robust to develop good technical match, quality, onboard, and attrition ranks individually and then have a flexible rank aggregation procedure that reflects current priorities.

4.2 Bipartite Ranking with Univariate Loss

In the previous section, we described several scoring functions that need to be developed. Indeed similar scoring functions can be developed for other stages in the human resource lifecycle beyond quality, onboard, and attrition. As we have described them, all of these ranking algorithms are to be learned using historical training sets that provide a binary label for whether a given candidate passes or fails a particular stage in the lifecycle. Consequently, we use of bipartite ranking algorithms [2].

The goal of a ranking algorithm is to establish a total order on candidates such that positive instances precede negative ones in the ranked list. Consequently, traditional algorithms for bipartite ranking call for minimizing the number of disagreements (or misorderings) among *pairs* of ranked samples [2]. Such algorithms reduce ranking to a binary classification problem by treating each pair of instances as a single object that should be classified as positive if the pairwise ordering is correct and negative if the pairwise ordering is incorrect. Given both the large numbers of features that are typically used in hiring decisions and the large training sets, this is often computationally infeasible.

Instead, we apply a classification algorithm directly on the training data (where each candidate is labeled as passing or failing) rather than on pairs of candidates (where each pair is labeled as correctly ordered or incorrectly ordered). Using classification algorithms directly on the labeled training data to perform ranking has been demonstrated in many domains to perform well in practice [14] and is also computationally simpler. In particular, we perform bipartite ranking through minimization of a standard univariate loss function [23]. This approach of using univariate loss to do ranking has provably good performance for margin-based classifiers [23]. Many margin-based classifiers, however, like support vector machines and AdaBoost do not work well with the kind of categorical features that are typically present in hiring processes.

Due to the presence of many categorical features, we train random forest classifiers for binary classification [7], which are ensembles of decision trees trained on random subsets of training data. In operation, new instances are classified using each tree in the ensemble. If we were to use the random forest for a binary classification tasks, we would take the majority vote of the individual trees, but since we are interested in ranking, we use the actual distribution of votes as a score upon whose basis to rank. For example, if there are 100 trees in the ensemble and 36 trees say an unseen candidate i is predicted to onboard and 64 trees say the candidate is predicted to decline the offer, then the onboard likelihood score will be $o_i = 0.36$.

Algorithm 1 describes the learning phase of the bipartite ranking algorithm using a historical dataset and Algorithm 2 outlines the score computation phase of the bipartite ranking for previously unseen examples.

In contrast to margin-based classifiers or discriminant techniques such as linear discriminant analysis or partial least squares discriminant analysis, random forests do not explicitly maximize the margin, thus making the score/margin an unbiased measure that is directly related to generalization error. In contrast to these other methods [23], it remains an open question in machine learning theory to determine provable performance guarantees on random forests for bipartite ranking.

Algorithm 1 Train Bipartite Ranking Algorithm with Univariate Loss Random Forests

Require: historical features $\mathcal{Z}$, labels L, ensemble size N, sampling factor S

random forest, $F \leftarrow \phi$
$C \leftarrow S \times |\mathcal{Z}|$
for $i = 1$ to N **do**
  $\mathcal{Z}' \leftarrow$ randomly sample C instances from $\mathcal{Z}$
  $L' \leftarrow$ labels for the instances $\mathcal{Z}'$
  $T \leftarrow$ train decision tree using $\mathcal{Z}'$ and L'
  $F \leftarrow F \cup T$
end for
RETURN F

Algorithm 2 Compute Bipartite Ranking Score with Univariate Loss Random Forests

Require: random forest $F = \cup T_i$, new instance z

Initialize $Classification$ array of length $|F|$ with zeros
for $i = 1$ to $|F|$ **do**
  $Classification[i] \leftarrow$ decision for z using T_i
end for
$Score \leftarrow$ normalize $Classification$
RETURN $Score$

4.3 Joint Ranking

The various individual ranking methods along the dimensions of technical match, quality, onboard, and attrition likelihood will generally produce different rankings. Presenting many rankings to HR practitioners directly, however, would cause confusion and necessitate some potentially quirky practitioner-specific unified ranking. For consistency and reduced information overload, it is imperative that the system merge rankings to produce a single ranked list that takes into account all the different facets according to current business concerns.

An important point to note about several of the individual rankings is that they are valid only for the subset of the candidate population similar to the population used for training. Indeed, due to the sequential nature of the hiring stages (Figure 1), there is a hierarchical

Business Conditions	Weight Vector $(\lambda_t, \lambda_q, \lambda_v, \lambda_o)$
normal	$.33, .34, .16, .17$
urgent delivery need	$.2, .2, .5, .1$
niche skills	$.6, .2, .1, .1$
easy to learn skills	$.1, .4, .1, .4$
recession	$.2, .2, .05, .55$

Table 1: Business Objectives and Weight Vector

nature to the individual rankings. The exact nature of the hierarchical relationship will depend on business setting. However, some practical rules of thumb are:

- Technical Match rank is applicable to all candidates for a given job requisition.
- Quality rank is applicable only to candidates for whom the skill set matches the job requirement (eligible for interview).
- Onboard rank is applicable only to candidates which pass interview and are offered the job (eligible for offer).
- Attrition rank is applicable only to candidates that have come on board (now employees).

Although several algorithms exist for rank aggregation [18], for clarity of understanding by the human resource executives that implement current business concerns, we use a simple weight-based method that respects the hierarchical structure. Formally, the overall score S_i of the i^{th} application is computed as:

$$S_i = \lambda_t t_i + \lambda_q q_i + \lambda_o o_i + \lambda_v v_i, \tag{1}$$

where the weight vector λ has positive elements summing to one. Due to the hierarchical structure of the several screens however, the scoring cannot be applied directly as described. Instead, we must create a ranked list at the lowest level of the hierarchy first (by setting hierarchically higher weights λ_q, λ_o, and λ_v to zero) and then only include further factors as we move up the list past given thresholds. So in fact we end up with a nested structure of ranks that determine the complete rank. Because of the nesting, one can think of the final rank as a partially ordered set of groupings that are internally ranked into a total order. As far as we know, such a hierarchical nested structure of ranks is novel to this work, cf. [18].

With this hierarchical weight-based approach, weights λ can be chosen and matched to current market conditions, business goals, computed benefits of high quality candidates, and costs of the hiring process and of attrition. Some typical business conditions and the corresponding weight vectors at the highest level of the hierarchy are shown in Table 1; these are only meant as typical examples.

If a position needs to be filled urgently, then the enterprise may give most attention to onboard likelihood and overlook attrition risk. If there is a need for niche skills, then the technical match may be most important. On the other hand, for easy to learn skills, the organization may decide to compromise on technical match and to train people that have high absolute quality; attrition risk would be a large factor so as to maximize the return on training investment. In a recession, it likely that candidates will come onboard if offered a position, but it may be important to ensure low attrition risk once macroeconomic conditions improve.

In general it is possible to determine the appropriate weights herein from other HR data analytics systems that look at market conditions and business needs, but this is beyond the scope of the present paper.

4.4 System Usage

Let us briefly describe how the data mining driven rankings are used for decision support by human resource practitioners in our deployment. Recall the goal is to enable faster and better decisions. Before going into different strategies for generating an actionable plan from the ranking, we group the candidates in three buckets and explain the desired action for each group.

- **High Ranked Candidates (H)**: These are the top candidates, high on all four facets. These candidates have a high technical match with the job description, high quality, high chance of onboarding, and low attrition risk.
- **Average Ranked Candidates (A)**: These candidates provide low or average scores in one or two rankings. Depending on the dimension on which they are ranked low, the HR personnel can take a business decision.
- **Low Ranked Candidates (R)**: These candidates with poor ranking will most likely will be rejected. Hence, they should be processed only if necessary when resources are available.

As in every discretization scenario, determining the cut-offs for binning is a hard problem. There are several ways to determine thresholds that delineate the groups, but we find it is best to use a method driven by the relationship between hiring demand target D and the number of applicants S. A key advantage of supply/demand-aware categorization is that the effect of supply & demand dynamics on hiring strategy is already taken into account. For example if $D \approx S$, then nearly all candidates should be in group H. On the other hand if $D \ll S$, then many candidates should be placed in the R group.

Depending on the rank of a candidate, there may be different action strategies. We present three strategies which range from highly conservative to highly ambitious along with the business metric which will be most impacted.

- **Strategy 1: Conservative** In this approach, the candidates are processed according to their ranking. However, the business process of conducting multiple round of interviews does not change. The advantage is that a smaller number of candidates will be processed (majority of high ranked candidates will be offered and will onboard) to fill the required vacancies. Therefore, the overall cost

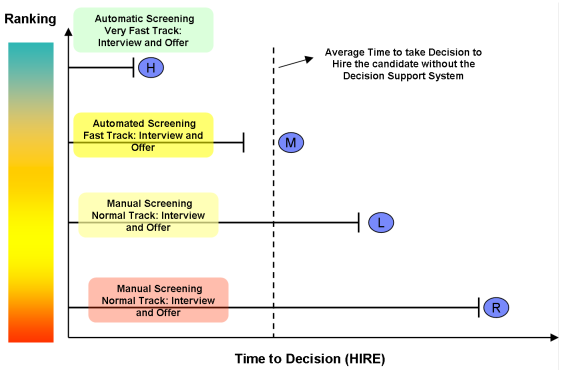

Figure 3: Using the data mining techniques through decision support as candidate prioritization.

will decrease, however, the lead time to hire will remain similar to the business-as-usual process.

- **Strategy 2: Moderate** In this approach, the high-ranked candidates will be fast-tracked through the interview and offer process while the average-ranked candidates will be processed in a business-as-usual methodology. The low ranked candidates will only be considered in the unlikely event of open positions which are not filled by higher ranked candidates. The overall cost of hiring as well as lead time to onboard will decrease by adopting this methodology.

- **Strategy 3: Ambitious** In this approach, the interview process for high-ranked candidates is skipped and is replaced by an online technical test. The candidates are directly called to discuss the offer. At the same time, the low-ranked candidates are rejected while the average candidates are processed in business-as-usual mode. This strategy will lead to significant improvements in lead time as well as cost reductions.

Evidently, strategy choice depends on an organization's ambition and confidence in embracing analytics-based solutions.

We suggest that organizations start with a conservative approach, analyze business performance and gradually shift to a moderate approach and after further analysis of business performance shift to an ambitious strategy. Moreover, different methods can be chosen for different job descriptions. Finally, the supply-demand characteristics will also play a role. If $D \ll S$, then ambitious approach is more suitable than when $D \approx S$ where the conservative approach should be followed.

5. EXPERIMENTAL VALIDATION

We conducted experiments on real candidate hiring data collected by a large enterprise over the period of three years which included hundreds of thousands of applications as well as the eventual status of these applications.[2]

Rather than showing all of the traditional machine learning metrics for the specific data mining components of our system [2, 23], we demonstrate value through two metrics that have business importance. These are the hiring yield and the attrition accuracy. By focusing on business metrics, we are able to align objectives with the larger systematic concerns of the sociotechnical system.

Hiring yield is the ratio of number of candidates actually onboarded to the number of candidates we predicted as onboards. If the yield value is closer to 1, the enterprise incurs less cost in interviewing additional candidates. Similarly, attrition accuracy is the ratio of number of candidates who actually turned out to be long term employees to the number of candidates we predicted to be long term employees. Again, costs are lower with greater attrition accuracy.

We used the large database of available job applications to train the bipartite ranking algorithm for onboard likelihood and for attrition likelihood, see Section 4. We selected the ensemble size to be 100 trees by looking at the out-of-bag classification accuracy as a function of ensemble size to prevent overfitting.

For evaluation, we considered the largest eight job requisitions in terms of the number of applicants for a distinct pool of applications separate from the data used for training. Although in practice we perform ranking, it is often more insightful to report performance numbers using the classification with reject option formulation [33]. Using the trained onboard likelihood model, which scores likelihoods between 0 and 1, we predicted that all candidates scoring above 0.55 would onboard whereas candidates scoring below 0.45 would decline an offer. Those in the middle were marked as 'manual' and should be considered by human screeners.

The hiring yield with manual hiring was 59.8% whereas using the decision support system applied as described above, the hiring yield would have been 74.4%. Thus using the tool would have led to a dramatic 14.6% increase in hiring yield over business as usual; significantly fewer interviews could have been conducted to have the same number of candidates onboard.

Further, we binned candidates by onboard likelihood and measured yield, as shown in Figure 4. As designed, candidates with higher onboard scores have higher yield.

For attrition likelihood training, we treat a candidate as a positive example if he/she has completed a tenure of $t + \delta$ months whereas we treat the candidate as a negative example if candidate left the company before completing t months. A gap of δ months is used, so as to mitigate certain aspects of data noise. We restrict $t \leq 12$ since features in pre-hire data become rather irrelevant after one year and factors such as job satisfaction, pay, and organizational factors become key determinants of employee turnover [24]. This is evident in 10-fold cross-validation results presented in Figure 5 wherein lines corresponding to smaller values of t are above the larger ones (green $\succ$ blue $\succ$ orange) indicat-

[2]Due to the confidential nature of the data used, we do not list precise values of headcounts or show them in the figures.

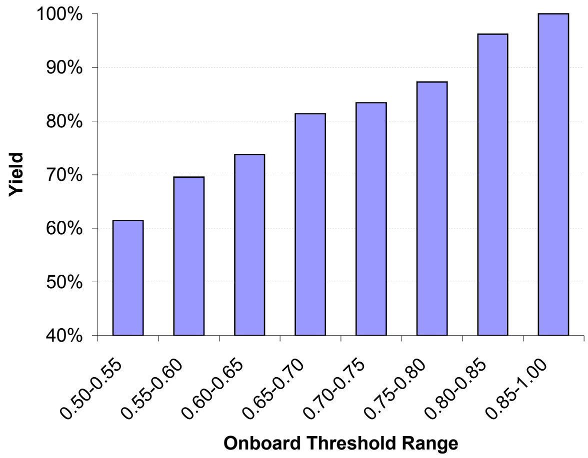

Figure 4: Onboard likelihood is effective at ranking for hiring yield.

ing greater relevance of features for smaller values of t. Figure 5 also captures the effect of δ on classifier performance: increasing values of δ for the same value of t results in improved data mining performance as measured through the F_1 score that incorporates precision and recall.

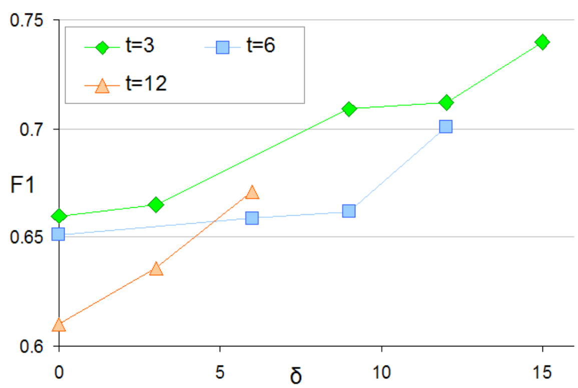

Figure 5: Effect of t and δ on the precision-recall performance.

Looking in detail at some features reveals that contrary to expectation, intuitive attributes thought to be associated with job hopping such as number of previous employers and average time in earlier organizations are not strong indicators of early attrition.

6. DEPLOYMENT

Having demonstrated the efficacy of the predictive analytics tools through experiments on real data, we built a user interface (UI) that aids in efficient multifaceted screening of candidates in order to deploy the system to HR practitioners. As was suggested in Figure 2, the UI is meant to support filtering and screening resumes, generating business intelligence reports, and taking actions for given candidate applications. Figure 6 shows a screenshot of the deployed system. Several options are highlighted in the screenshot, as briefly described below.

Job Id Search Box.

This is to search all candidates that have applied for the given job id. It lists candidates in the default order from highest to lowest rank.

Resume Search Box.

This is to filter listed candidates using certain keywords that appear in their profile information.

Facets on Scores.

As shown in the screenshot, there are two facets on scores. The 'Order by Scores' facet is to define the weighted ranking criteria λ of candidates based on the several individual scores. Joint score is displayed for every candidate and is used to rank candidates in descending order. Another facet 'Filter by Scores' is used to assign a threshold for each score so as to facilitate the hierarchical nature of joint ranking. Only those applications that satisfying all thresholds will be shown in the ranked list.

Skills Facet.

There are several other facets along with skills facet like work experience facet, applicant source facet, etc. (not shown in screenshot). This was previously presented as part of PROSPECT [30].

Upon deployment, HR practitioners in a large organization have reported the efficacy and ease of use of the system. It makes big data a resource to meaningfully prioritize their efforts, rather than a cause for information overload.

7. EXTENSIONS TO SOCIAL RESUMES

Candidate hiring through social media has gained a lot of interest from HR professionals recently. Key reasons include access to a much larger talent pool and lower cost of hiring as compared to sourcing candidates from external agencies or through employee referral programs. Various algorithms as well as commercial tools[3] are available which combine the social structure of these sites with simple keyword (skills) search to discover and connect to potential candidates.

A typical *social hiring* process has the following steps:

Search.

The HR professional uses the search facility to find candidates which have skills sets needed for the current job description.

Analysis.

The HR professional manually looks at each profile to find candidates which are best suited. Various filters

[3]Examples include the following websites:
`http://www.selectminds.com/`
`http://www.jobs2web.com/`
`http://www.fadv.com/candidate-sourcing/`

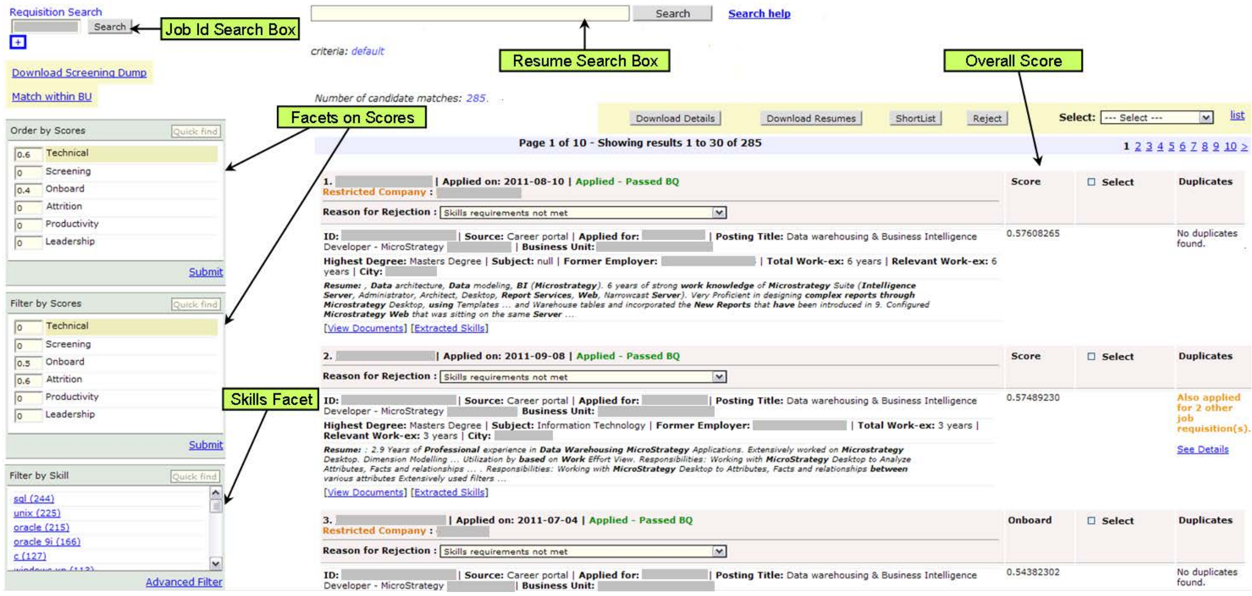

Figure 6: Screenshot of the user interface of the deployed system.

like education and previous employers can also be used to discover candidates.

Reach.

Candidates can then, again, be manually analyzed to discover a reachability path based on the HR professional's existing network. The candidates can then be contacted through the social network or via 'cold' email to discuss job opportunities.

Communication.

Once the potential candidate is reached, the actual hiring dialog starts which may result in one of the following outcomes: candidate not interested; candidate rejected through an interview process; job offered but candidate declined; and candidate hired and onboarded.

Existing methods have two inter-related problems. First, the manual Analysis and Reach steps are time-consuming, cumbersome, and error-prone. There is no guarantee that even after spending much time, the HR professional will be able to meet the final goal of hiring candidates. Second, one can note that determining that a candidate is not interested involves very little cost. Effectively, it requires one email or short phone conversation for judging the interest level of the candidate. However, other results are rather costly since the organization would have spend efforts in conducting interviews as well as in structuring offers.

The basic abilities of our multifaceted screening system can help to greatly reduce this cost, by performing an intelligent and automated analysis that ranks each candidate.

Thus far, we have developed and deployed our system for the use of an organization where resumes are collected in traditional file formats like `.doc`, `.rtf`, or `.pdf`. However, we have taken preliminary step to make use of resumes available on professional social networking sites as well. The only change, as compared to PROSPECT, is in the conditional random field models to extract the structured information from the semi-structured social resumes. In fact, the task is much easier for social resumes than traditional resumes due to the standardized structure with sections like education, skills, and job experience. Moreover, the information within each section is also structured. This contrary to traditional resumes where neither structure nor content are standardized. As a simple example, traditional resumes may refer to an education section by multifarious names like *Educational Background*, *Education*, *Academics*, or *Academic Qualifications*.

Since the information from traditional and social resumes is rather similar, we can use the data mining models trained on traditional resumes for evaluating and ranking social resumes as well. This is in progress.

Please note that the extended system will only fetch and process publicly available data. Moreover, some social network sites impose restrictions on how much data can be crawled or scraped.

8. CONCLUSION

We have presented a system to aid human resource practitioners in achieving more informed and better decision making in the recruitment process. Our system extracts key features from candidate resumes and structured data; prioritizes applications on a joint criterion that considers several factors such as technical match to job requirement, quality, likelihood of joining (if offered), and likelihood of early attrition (if joined). Given the nature of historical training data available in the various stages of the hiring lifecycle, most of the data mining and predictive analytics components are bipartite rankers that minimize univariate loss functions using random forest classifiers. The efficacy of the system has been demonstrated on large dataset of real candidate applications. HR practitioners in a large organization find great value in the system, especially since it is difficult for them to manually predict onboard and attrition likelihoods using only pre-hire data and these factors are of importance to controlling business costs.

One might wonder whether macroeconomic factors and labor market conditions that change with time might negatively impact the performance of machine learning algorithms that are trained on historical job candidates but applied on current candidates. There can certainly be non-stationarity and it is an ongoing challenge to balance the training sample size against the coherence between the training and active sets of candidates. Notwithstanding, our experimental results show the given retraining period length is effective in trading off coherence and sample size. In practice, retraining can be performed periodically, e.g. on a quarterly basis, or performed when practitioners recognize significant changes to the hiring environment.

In this paper we limited discussion of business metrics to efficiency gains. Beyond efficiency gains, however, we believe the system also helps in hiring better quality candidates that may otherwise get lost in the shuffle due to the sequential selection strategy often adopted by human resource practitioners facing information overload. The best workers may be 40% more productive than others [21]. We are currently tracking the performance of candidates selected using our system to quantify this gain.

As it currently stands, our system has been trained, validated, and deployed for hiring individual contributors in technical jobs such as computer programmers and testers. Although this constitutes the bulk of hiring in the services industry, there are a wide range of other applications. Once features that capture candidate attributes like creativity, leadership, and risk-taking are validated, the system may be used in the hiring processes for sales, management, and perhaps even research.

Finally, we note that the problem of selecting a few candidates from a large pool arises not only in job hiring but also in college admissions settings [10]. We are exploring the possibility of deploying our system in several other domains, including as a feature within social networking sites.

9. REFERENCES

[1] IBM business conduct guidelines, 2012.

[2] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth. Generalization bounds for the area under the ROC curve. *J. Mach. Learn. Res.*, 6:393–425, Apr. 2005.

[3] C. V. Apte, S. J. Hong, R. Natarajan, E. P. D. Pednault, F. A. Tipu, and S. M. Weiss. Data-intensive analytics for predictive modeling. *IBM J. Res. Develop.*, 47(1):17–23, Jan. 2003.

[4] H. Aryani. The legal implications of biodata use as an employment selection practice. *Univ. Penn. J. Bus. Law*, 11, Summer 2009.

[5] A. Bangerter, N. Roulin, and C. J. König. Personnel selection as a signaling game. *J. Appl. Psychol.*, 97(4):719–738, July 2012.

[6] J. A. Breaugh and M. Starke. Research on employee recruitment: So many studies, so many remaining questions. *J. Manage.*, 26(3):405–434, June 2000.

[7] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001.

[8] H. Cao, J. Hu, C. Jiang, T. Kumar, T.-H. Li, Y. Liu, Y. Lu, S. Mahatma, A. Mojsilovic, M. Sharma, M. S. Squillante, and Y. Yu. OnTheMark: Integrated stochastic resource planning of human capital supply chains. *Interfaces*, 41(5):414–435, Sept.-Oct. 2011.

[9] H.-Y. Chang. Employee turnover: a novel prediction solution with effective feature selection. In *Proc. 3rd WSEAS Int. Conf. Comput. Eng. Appl.*, pages 252–256, 2009.

[10] L. Chang. Applying data mining to predict college admissions yield: A case study. *New Dir. Institutional Res.*, 2006(131):53–68, Autumn 2006.

[11] D. S. Chapman, K. L. Uggerslev, S. A. Carroll, K. A. Piasentin, and D. A. Jones. Applicant attraction to organizations and job choice: A meta-analytic review of the correlates of recruiting outcomes. *J. Appl. Psychol.*, 90(5):928–944, Sept. 2005.

[12] V. Chenthamarakshan, K. Dey, J. Hu, A. Mojsilovic, W. Riddle, and V. Sindhwani. Leveraging social networks for corporate staffing and expert recommendation. *IBM J. Res. Develop.*, 56(3):11:1–11:10, Nov. 2009.

[13] S. Clémençon and N. Vayatis. Tree-based ranking methods. *IEEE Trans. Inf. Theory*, 55(9):4316–4336, Sept. 2009.

[14] C. Cortes and M. Mohri. AUC optimization vs. error rate minimization. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[15] J. L. Cotton and J. M. Tuttle. Employee turnover: A meta-analysis and review with implications for research. *Acad. Manage. Rev.*, 11(1):55–70, Jan. 1986.

[16] T. H. Davenport, J. Harris, and J. Shapiro. Competing on talent analytics. *Harvard Bus. Rev.*, 88(10):52–58, Oct. 2010.

[17] T. H. Davenport and J. G. Harris. *Competing on Analytics: The New Science of Winning*. Harvard Business School Press, Boston, 2007.

[18] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proc. 10th Int. Conf. World Wide Web (WWW'01)*, pages 613–622, May 2001.

[19] F. Farber, T. Keim, and T. Weitzel. An automated recommendation approach to selection in personnel recruitment. In *Proc. 2003 Am. Conf. Inf. Syst. (AMCIS)*, Aug. 2003.

[20] S. Hansell. Google answer to filling jobs is an

algorithm. *New York Times*, Jan. 2007.

[21] J. E. Hunter, F. L. Schmidt, and M. K. Judiesch. Individual differences in output variability as a function of job complexity. *J. Appl. Psychol.*, 75(1):28–42, Feb. 1990.

[22] T. Keim. Extending the applicability of recommender systems: A multilayer framework for matching human resources. In *Proc. 40th Hawaii Int. Conf. System Sciences*, page 169, Jan. 2007.

[23] W. Kotłowski, K. Dembczyński, and E. Hüllermeier. Bipartite ranking through minimization of univariate loss. In *Proc. 28th Int. Conf. Mach. Learn. (ICML 2011)*, pages 1113–1120. June 2011.

[24] A. I. Kraut. Predicting turnover of employees from measured job attitudes. *Org. Behav. Human Perform.*, 13(2):233–243, Apr. 1975.

[25] S. Laumer and A. Eckhardt. Help to find the needle in a haystack - integrating recommender systems in an IT supported staff recruitment system. In *Proc. Special Interest Group Manage. Inf. Syst. 47th Annu. Conf. Comput. Personnel Res. (SIGMIS CPR'09)*, pages 7–12, May 2009.

[26] I. Lee. An architecture for a next-generation holistic e-recruiting system. *Commun. ACM*, 50(7):81–85, July 2007.

[27] J. Malinowski, T. Weitzel, and T. Keim. Decision support for team staffing: An automated relational recommendation approach. *Decis. Support Syst.*, 45(3):429–447, June 2008.

[28] S. Parasuraman. Predicting turnover intentions and turnover behavior: A multivariate analysis. *J. Vocat. Behav.*, 21(1):111–121, Aug. 1982.

[29] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proc. 25th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval (SIGIR '02)*, pages 253–260, Aug. 2002.

[30] A. Singh, C. Rose, K. Visweswariah, V. Chenthamarakshan, and N. Kambhatla. PROSPECT: a system for screening candidates for recruitment. In *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage. (CIKM'10)*, pages 659–668, Oct. 2010.

[31] P. Soelberg. Unprogrammed decision making. Technical Report 239-67, Alfred P. Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA, 1967.

[32] R. P. Steel and N. K. Ovalle. A review and meta-analysis of research on the relationship between behavioral intentions and employee turnover. *J. Appl. Psychol.*, 69(4):673–686, Nov. 1984.

[33] K. R. Varshney. A risk bound for ensemble classification with a reject option. In *Proc. IEEE Workshop Stat. Signal Process. (SSP 2011)*, pages 769–772, June 2011.

[34] X. Yi, J. Allan, and W. B. Croft. Matching resumes and jobs based on relevance models. In *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval (SIGIR '07)*, pages 809–810, July 2007.