Analytics-Led Talent Acquisition for Improving Efficiency and Effectiveness

Girish Keshav Palshikar, Rajiv Srivastava, Sachin Pawar, Swapnil Hingmire, Ankita Jain, Saheb Chourasia and Mahek Shah

Abstract Large IT organizations every year hire tens of thousands of employees through multiple sourcing channels for their growth and talent replenishment. Assuming that for each hire at least ten potential profiles are scrutinized and evaluated, the Talent Acquisition (TA) personnel ends up processing half a million-candidate profiles having multiple technical and domain skills. The scale and tight timelines of operations lead to possibility of suboptimal talent selection due to misinterpretation or inadequate technical evaluation of candidate profiles. Such recruitment process implementation due to manual, biased, and subjective evaluation may result in a lower job and organizational fit leading to poor talent quality. With the increased adoption of data and text mining technologies, the recruitment processes are also being reimagined to be effective and efficient. The major information sources, viz., candidate profiles, the Job Descriptions (JDs), and TA process task outcomes, are captured in the eHRM systems. The authors present a set of critical functional components built for improving efficiency and effectiveness in recruitment process. Through multiple real-life case studies conducted in a large multinational IT company, these components have been verified for effectiveness. Some of the important components

G. K. Palshikar · R. Srivastava (✉) · S. Pawar · S. Hingmire · A. Jain · S. Chourasia · M. Shah
Tata Research Development and Design Centre, Pune 411013, India
e-mail: rajiv.srivastava@tcs.com

G. K. Palshikar
e-mail: gk.palshikar@tcs.com

S. Pawar
e-mail: sachin7.p@tcs.com

S. Hingmire
e-mail: swapnil.hingmire@tcs.com

A. Jain
e-mail: ankita7.j@tcs.com

S. Chourasia
e-mail: saheb.c@tcs.com

M. Shah
e-mail: shah.mahek@tcs.com

A. K. Laha (ed.), *Advances in Analytics and Applications*, Springer Proceedings
in Business and Economics, https://doi.org/10.1007/978-981-13-1208-3_13

elaborated in this paper are a resume information extraction tool, a job matching engine, a method for skill similarity computation, and a JD completion module for verifying and completing a JD for quality job specification. The tests performed using large datasets of the text extraction modules for resume and JD as well as job search engine show high performance.

Keywords e-Recruitment · Talent acquisition · Resume information extraction
Job matching · Skill similarity · HR analytics

1 Introduction

The HR analytics is an active area of research and talent acquisition, in particular, has garnered significant attention of late. The HR journals as well as IT applications and systems journals have encouraged the analytics and technology enablement of the HR processes. Schiemann (2014) gives importance to the selection process and interview process to focus on measuring effectively the candidate alignment and engagement to the organization's goals values and culture. The failure results in a misfit and can be improved using the technology-enabled recruitment processes as well as by evaluating candidates on the right set of parameters related to the organizational culture, values, and environment. Parthasarathy and Pingale (2014) inform that the use of e-recruiting and web-enabled functions has brought the collaborative approach in talent acquisition and management. He also emphasizes that the online experience of web browser access with interactive user-interfaces, social networking enabling collaboration, and participation of community is essential. He emphasizes that the efficiency metrics such as days-to-hire are popular but now recruitment effectiveness in terms of the quality of talent hired is also being targeted for measurement.

Srivastava et al. (2015) provide several predictive analytics based point solutions to address TA needs such as predicting joining delay, selection likelihood, offer acceptance likelihood, and other similar solutions to improve effectiveness of the TA processes. Dutta et al. (2015) highlight the importance of the quick decision-making, innovative methods of talent acquisition, and focused metrics for the function of the HCL TA Group (TAG) as it is gearing up for strategic recruitment. HCL has realigned TAG by implementing change initiatives for its members using aggressive SLAs with the business stakeholders. They are yet to start use of data mining for insights and text mining for efficiency improvement. Faliagka et al. (2012a, b) describe an approach for ranking job applicants for recruitment in web-enabled systems. Their proposed system implements candidate ranking, using objective criteria which are made available from the applicant's LinkedIn profile. The candidate's personality features are also extracted from her social activity using linguistic analysis. The Faliagka et al. (2012a) use Analytic Hierarchy Process (AHP) for ranking profiles, whereas we have devised functions for similarity and matching score computation which are based on the acquired domain knowledge. The Faliagka et al. (2012b) use text mining of LinkedIn for creating profile and use linguistic analysis for inferring personality characteristics. In our work, we are extracting attributes from candidate's resume using Resume Information Extractor (RINX) as well as planning to combine information from multiple online and social platforms for the technical and domain skills using extraction tools. Bui and Zyl (2016) give insight into how gamification, a new technique, can be used for acquiring talent. The findings suggest that gamification platforms can be used to align the prospective employee's interests and identity with the organization. It can also act as a personalization tool which may make employee onboarding smoother.

Edmundson (1969) has proposed simple rule based techniques for automated extraction of structured information from the unstructured textual data. Mooney and Bunescu (2005) have applied knowledge extraction from the unstructured text using text mining. With increase of machine learning and natural language processing techniques, Téllez-Valero et al. (2005) and other researchers tried to solve this problem of automatic extraction. On resume documents different extraction techniques are used to make candidate selection process (Tomassetti et al. 2011) easier and automatic.

2 Analytics for Talent Acquisition

Talent Acquisition (TA) is a fundamentally important function within a company's HR function, responsible for recruiting high-quality workforce, and important for the successful operations and growth. The recruitment fulfills demand pipeline of futuristic domains and technical skill requirements in shortest possible time frames, at lowest costs, using diverse channels and from multiple locations. Leading IT service providers which have strength of more than 200,000 Full-Time Employees (FTEs) every year hire 40,000–50,000 employees for their growth and talent replenishment. These organizations work in multiple industrial domains requiring combinations of technical and domain skills for executing projects worldwide. The experienced candidates are sourced from recruitment agencies, online job portals, and social collaboration platforms such as LinkedIn™[1]. The fresh graduates are engaged and groomed through social, community portals run by these organizations who are subsequently screened for hiring as FTEs. Assuming that for each hire at least ten potential profiles are shortlisted by the TA personnel, it requires half a million-candidate profiles collected for scrutiny and proportionate efforts for the complete recruitment process. These shortlisted candidate profiles are further screened by the requestor group who selects, for example, five out of ten profiles for the interviews and onward selection process as depicted in Fig. 1.

The talent selection which results from inadequate evaluation or misinterpretation of candidate profiles would result in lower job and organizational fit resulting in poor talent quality. These losses can be attributed to the large volume of candidate profiles being screened and profiles mostly being textual leading to subjective and biased interpretations by the personnel involved in shortlisting and selection. An important

[1] LinkedIn™ is a trademark owned by LinkedIn™ recruiting services.

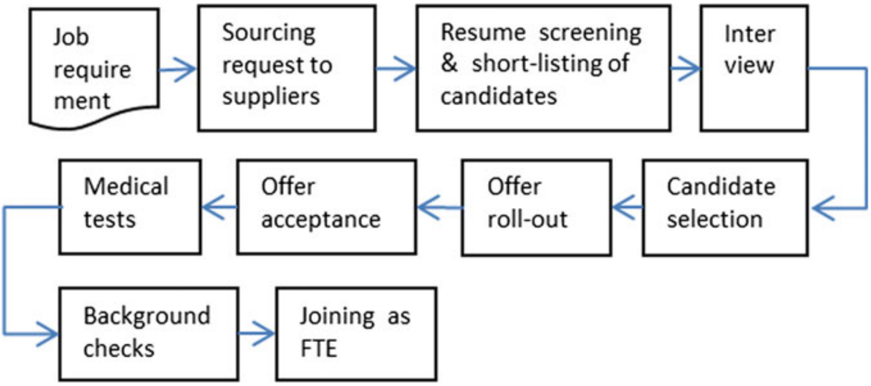


Fig. 1 A typical recruitment process flow

issue is inadequate technical or domain knowledge held by the TA personnel who have HR background. Other process constraints such as fulfilling a talent requirement within a given time limit or requirement being in bulk for a quick ramp-up would put additional pressure on the recruitment personnel leading to inadvertent oversights during the selection. Also, a JD being a crucial talent requirement document any oversight such as noninclusion of an important supplementary skill may lead to suboptimal fit, utilization of available talent as well as possible effort and time overrun in a project. To avoid these situations, an intelligent JD completion recommender can be developed which does an automatic review and suggests best skills or roles or task names likely missing from a JD.

With the increased adoption of technologies enabling analytics, the recruitment processes are also being reimagined. With the increase in number of skills being used across knowledge-based industries as well as the increase in available skilled candidates, it is an imperative for TA function to adopt analytics based on the data mining, text mining, machine learning, and statistical methods (Hastie et al. 2008; Tomassetti et al. 2011) to incorporate intelligence and automation for improving effectiveness and efficiency.

The major information sources, viz., candidate profiles, the JDs and TA process outcomes such as shortlists, interviewers, evaluations, and selected profiles, are available in electronic form. These data are captured in the eHRM (Bondarouk et al. 2009; Strohmeier 2007) systems deployed for automating workflows and tasks of recruitment process such as shortlisting, interviewing, offer generation as well as early engagement of the selected candidates to increase probability of acceptance of job offer and early joining. In the commonly used workflow automation systems, the inputs of candidate profile, JD details, and various decision support activities are completely manual. The primitive matching available in the job portals uses only a few attributes such as experience range, role, and skills. Most of the data being textual, the word similarity-based matching is elementary and is prone to suboptimal solutions. There are several manual processing steps which are effort intensive, repetitive, and strenuous, and these lead to the end-user dissatisfaction due to the low-quality hires. Some of the important automation opportunities are listed as follows:

- Candidate has to *manually re-enter* details of her profile to facilitate structured storage in the employer database through a web interface, though her resume already has all details in it.
- Sourcing personnel has to *manually derive* additional latent attributes from the candidate profile, such as quality of education, skill level, as well as soft skills such as communication abilities, leadership potential, etc. to help in shortlisting.
- Sourcing personnel has to *shortlist* a list of matching candidates for a JD using a rich set of latent quality attributes, in addition to experience and held skills.
- Continuing the previous point, *automated matching* would require user to understand the salient parts of the *given textual JD* to form a query to retrieve best matching candidate profile.

In addition to the possible automation of these abovementioned manual tasks, there are several improvements possible by use of the data and text mining techniques (Ronen et al. 2006). For example, using information retrieval we can improve the accuracy of the automated matching of a JD to candidate profiles for increasing effectiveness of the talent acquisition function. A few of such improvement opportunities are listed as follows:

- Deriving and using *skill similarity* for shortlisting candidates for a JD
- Suggestions to *improve a submitted JD* specification for better job fitment.

In this paper, we present a set of components built during multiple real-life case studies for improving the efficiency and effectiveness of TA function in a large multinational IT company. The list of important functionalities implemented as components is as follows:

- A tool for resume information extraction, RINX;
- A search and matching engine for job description and similar queries, RINX-SE;
- A module to derive the richer set of quality attributes from resume and social media;
- A JD completion module for suggesting possible improvements in a JD;
- A module for computing skill similarity.

Earlier, extraction task was performed using gazette-based matching in which entities are marked in the text if that particular word or sentence is present in the gazette. This "strict" pattern-based matching does not handle variations, i.e., minor spelling mistakes, new evolving patterns, etc. leading to weak matching results. These proposed components have been developed and tested with large sample size of ~10,000 resumes and 200 JDs for matching. The extraction results for resume and JD are evaluated using the *F-measure*.[2] The *F-measure* achieved for extraction of various entities from JD is above 0.8, and from resume it averages around 0.75.

[2] *F-measure—harmonic mean of precision and recall. F-measure* $= \frac{2 * precision * recall}{precision + recall}$.

The rank correlation measures such as *Kendall's* τ *(tau)*[3] in job matching and skill similarity achieve value of 0.71 and 0.94, respectively, compared to earlier pattern-based matching.

3 Data Mining and Text Mining-Based Solution Components

The following subsections provide description and salient features of the developed components.

3.1 Resume Information EXtractor (RINX)

The *resume* or *curriculum vitae* (CV) or *bio-data* document contains vital information about the education, skills, experience, and expertise of a person. A resume contains a summary (or profile) of the entire work experience of a person. Typical information elements in a resume are personal details (e.g., name, current address, phone number, e-mail, etc.), work experience profile (e.g., period, organization, designation, etc.), educational qualifications (e.g., degree, branch, year of passing, college, score, etc.), and project worked on (e.g., duration, role, services delivered, technologies used, etc.). One significant way to improve the operational efficiency of people who need to use resumes is to automatically extract all the relevant information elements from each resume in the given set. For automated resume content extraction, Resume Information eXtractor (RINX) tool has been developed. RINX uses NLP techniques (Aggarwal and Zhai 2012) such as parsing (using *Google TensorFlow syntaxnet*) and Part-of-Speech (POS) tagging (kindly refer Brown Corpus manual at http://clu.uni.no/icame/manuals/BROWN/INDEX.HTM for a complete list of POS tags), named entity recognition for enriched text generation to facilitate robust pattern specification for the targeted information extraction as well as RINX includes gazette-based look-up.

As an example the sentence "`My role as a Business Intelligence Expert is responsible for Managing Business Intelligence Architecture`" from a resume is enriched for extraction using NLP tools as follows:

[3] *Kendall's* τ *considers rank ordering among entities. It is the ratio of difference between number of concordant pairs and discordant pairs to the total number of ranked pairs possible.*

```
<S><NP><PRP$>My </PRP$><NN>role </NN></NP><IN>as </IN><NP><DT>a
</DT><nmod_role><NNP>Business </NNP><NNP>Intelligence
</NNP><NNP>Expert </NNP></nmod_role></NP><VP><VBZ>is
</VBZ><JJ>responsible </JJ><IN>for </IN><NP><NNP>Managing
</NNP><NN>Business </NNP><NNP>Intelligence </NNP><NNP>Architecture
</NNP></NP><PUNCT>. </PUNCT></VP></S>
```

In this enriched text of the sentence from resume, the Noun Phrases (NPs) `My Role` and `Managing Business Intelligence Architecture` are eliminated since none of the hypernym trees for nouns `Role` and `Architecture` contain any word from the given list {*creator*, *expert*, *specialist*, *specializer*, *planner*, *person*, *individual*}. The NP `a Business Intelligence Expert` is selected as a “Role” value because it contains the role indicating noun *expert* from the list. The same technology of entity extraction from resume is used to automatically convert the textual JD into structured form.

3.1.1 Extraction of “Service Line”

A “Service Line” is a domain-specific (often technical) task or activity carried out by humans, with or without tools. A “Service Line” is typically mentioned as part of the project or job description which represents the actual work done or the service provided by the resume author. A “Service Line” is typically indicated by a Noun Phrase (NP), such as user interface design. Figure 2 shows a few examples of the ways in which service lines are mentioned in resumes.

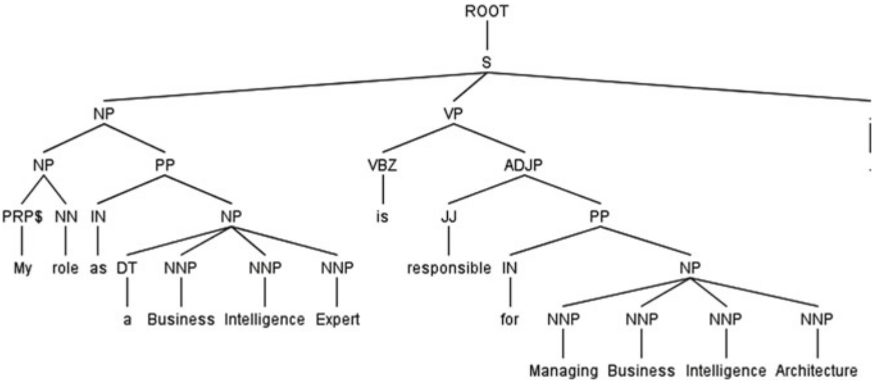
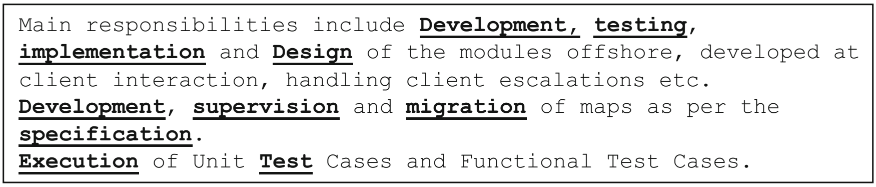


Fig. 2 Parse tree for an example sentence



Main responsibilities include **Development, testing**, **implementation** and **Design** of the modules offshore, developed at client interaction, handling client escalations etc.
Development, **supervision** and **migration** of maps as per the **specification**.
Execution of Unit **Test** Cases and Functional Test Cases.

Fig. 3 Examples of "service line"; values (underlined)

"Service Line" values are usually domain-dependent; e.g., "Service Line" in IT and finance domains are different. For higher flexibility, we want to minimize the use of gazettes (i.e., lists) containing known values for "Service Line". Moreover, we want to standardize different "Service Line" values, so as to facilitate retrieval/comparison in later stages (e.g., when matching candidates to specific job requirements). Lastly, we aim for at least 90% overall accuracy for extracting "Service Line" values. All this makes extraction of "Service Line" challenging (Fig. 3).

Algorithm 1 Service Line extraction

```
algorithm Service_Line
input set of $N$ sentences $\{S_1, S_2, ..., S_N\}$ from given resume
input $k$ // max no. of word senses to consider (default $k = 2$)
input $m$ // max no. of 'Service Line' values per project
            // description (default $m = 7$)
input  $L_1$; // list of prohibited nouns: work, scope etc.
input  $L_2$; // list of known 'Service Line' values: walk-through,
           // documentation, review etc.
output  $L$ // set of service lines selected for given resume
for each sentence $S_i$ and its parse tree $T_i$ do
  for each lowest-level NP (say $X$) in the sentence $S_i$ do
    if $X$ occurs under a VP node containing a copula verb then
      continue;
    if $X$ occurs under SINV node then continue;
    if $X$ contains comma or 'and' then // a list of nouns
      treat each noun in $X$ as a separate NP // e.g., if $X$ = "design
        // and development" then treat "design" and
      // "development" as two separate NPs
    if $X$ does not contain at least one noun (NN, NNS, NNP etc.)
      then continue;
    if $X$ fails any one of the tests specified in function check_np
      then continue;
    $L=\emptyset$; // $L$ = list of 'Service Line' values found in sentence $S_i$
    for each noun $Y$ in $X$ do // parent node of $Y$ in parse tree $T_i$
                                // has label NP, NNS, NNP etc.
      if $Y \in L_1$ then continue; // prohibited noun
```

```
         if $Y'$ is similar to a value in $L_2$ then // known value
            Add $Y'$ to $L$; continue; end if
         if $Y \in$ WordNet (e.g., budgeting) or $Y$ has a related noun
         then
            obtain root $Y'$ of $Y$ // testing→test, budgeting→budget
            Let $\mathbf{H} = \{H_1, H_2, \ldots, H_k\}$ denote the set of top $k$
            hypernym trees for $Y'$ obtained using WordNet;
            // use first k senses
            if no $H_i$ in $\mathbf{H}$ contains one of {"speech act", group
               action", "action", "change", "activity"} followed by
               "human activity" then continue; // not a human action
            Remove articles and prepositions from $X$;
            Add $X$ to $L$; // $X$ is a possible value for 'Service Line'
            break; // finished checking $X$; go to next NP
         end if
      end for
   end for
end for
Remove duplicate (or very similar) entries in $L$, if any;
// Keep at most $m$ NPs in $L$;
Select NPs which contain nouns from the list of known 'Service Line' values; if the number of such
NPs is less than m then keep other NPs in order of appearance in the project description;
return($L$)
```

Function check_np is used to filter out NPs which cannot be a possible value for "Service Line" entity.

Algorithm 2 For checking Noun Phrase (NP)

```
Boolean check_np($X$) // check if given NP $X$ is acceptable
// e.g., 'Oracle Retail 12 Applications' or 'Tally 4DOT5'
   if $X$ contains a number then return(0);
   // e.g., 'Raymark Merchandising Systems' is ruled out because
   // Raymark is not in WordNet.
   if $X$ contains a word not present in WordNet then return(0);
   if $X$ contains an acronym then return(0); // GIS, I.B.M.,
   // 'GE Consumer Finance' etc.
   if $X$ contains a cue word for an organization then return(0);
   // e.g., systems, company, incorporated, inc, bank.
   if $X$ contains a word having underscore then return(0);
   // e.g., 'SQL_Server'
   // Following strong conditions can be turned off by user
   if $X$ contains more than 2 nouns then return(0);
   // e.g., 'Sales Order Management System'.
   if $X$ contains ALL words with first letter capitalized
      then return(0); // e.g., 'Warehouse Management System'.
return(1); // OK
```

3.1.2 Extraction of Other Entities

RINX extracts a large number of entities from resumes: name, address, phone number, e-mail address, gender, date of birth, career profile (e.g., period, organization, and designation), educational qualifications (e.g., degree, branch, year of passing, college, score, etc.), and project worked on (e.g., duration, role, services performed for a project, technologies used, etc.). We describe extraction of a few of these entities now.

Period of a Project

The date range, if mentioned in a project description, usually indicates the period of that project. It always consists of two dates. The individual date is mentioned in various formats. We designed several complex patterns using regular expressions to convert a date in a commonly occurring format to a standard date format. A few examples of the common formats used for mentioning period of projects are "`Jan 06—Till date`", "`August 04-Dec 05`", and "`Oct 2001-Dec 2002`".

Technology and Tools

Within a given business domain, a technology refers to an organized body of formal knowledge and techniques that help the organization in providing better service or in producing better products. A technology is often implemented (and made useful in practice) as a machine, device, or a computer program. In IT domain, a technology is often indicated in the form of software tools (e.g., the database technology is indicated by tools such as Oracle, Sybase, and SQL Server). The names of tools mentioned in project descriptions in a resume are considered for deriving competencies, as these are the technologies used by the associate in actual work. Some examples of the technologies from IT domain are databases, graphics, programming languages, compilers, image processing, data mining, and model-driven development. The following list gives sample statements from resumes, which contain mentions of technology and tools.

(a) `Solution Environment: WINDOWS-2000/XP`
(b) `Tools: Eclipse 3.0, MS Visual Source Safe 6.0`
(c) `Support for Queue implementation in Message Driven Beans in MasterCraft 6.5.2 and 7.0`
(d) `Support for Weblogic 9.0 appserver in MasterCraft 6.5.2.`

RINX has complex patterns (as well as extensive gazettes of known tool names) to recognize technology and tools.

3.1.3 Experimental Results

Accuracy of the algorithm is measured on 160 free-form resumes of candidates in which entities are manually annotated. The resumes with manually annotated entities are part of the "Gold copy" and are used for the verification purposes. Only

Table 1 Extraction accuracy of RINX

Entity	Recall	Precision	F-measure
Date of birth	0.99	0.95	0.97
E-mail	0.98	0.93	0.95
Phone	0.96	0.95	0.95
Project date range	0.92	0.9	0.91
Project role	0.95	0.7	0.8
Project service line	0.97	0.95	0.96
Job duration	0.74	0.94	0.83
Employer	0.94	0.91	0.93
Degree name	0.86	0.9	0.88
Degree marks	0.71	0.93	0.81
Degree specialization	0.86	0.9	0.86
Degree university	0.77	0.89	0.82
Year of degree	0.74	0.94	0.83

information technology domain resumes have been selected for testing. The entities like date of birth, e-mail, phone numbers, project dates, service line, role, education qualifications, and career profile were extracted using RINX, wherein the extraction results were post-processed, e.g., cleaning with a negative list and then manually compared with the content of resume. The results consisting of precision, recall, and F-measure are provided in Table 1.

Please note that the extraction accuracy of entity ‘role’ is further improved by deriving it from the extracted service lines, as majority of service lines grouped together imply a role.

3.2 *RINX Search Engine (RINX SE)*

A resume search engine, for matching JDs or similar queries to the resume profiles (Patil et al. 2012), is developed to incorporate domain-specific knowledge to enhance the quality of search results. Given a search query for “`Microsoft`” as previous employer, a domain agnostic search engine would not be able to distinguish between resumes mentioning `Microsoft based skills` or `Microsoft` as an employer. As resume repository is first processed using RINX to extract structured information qualifying “`Microsoft`” as an employer eliminates error possible with a domain agnostic search engine. The extracted structured as well as unstructured resume contents are further indexed using Lucene, a text search engine library, for domain knowledge enabled search. Generally, the matching is performed using the technical skills and experience-based attributes specified in a typical JD, which only evaluates the “Job Fit” of a candidate. Our search engine provides domain-driven

Table 2 Skill competency results for selected technologies

Competency	Similarity score
ASP.NET 2.0	1
Microsoft.Net compact framework	0.8009
ASP.NET 4.5	0.7032
Microsoft collaboration tool	0.4762

approximate or elastic matching for attributes such as technical skill, role, specialization, etc. For the enrichment of the profile, we also derive further knowledge from resume, including the attributes such as "Quality of education", "Quality of experience", and "Proficiency Level in a skill", to aid evaluation of "Organizational Fitness". A set of domain-specific matching functions are also developed for ranking query results for shortlisting and expert identification purposes.

3.3 Skill Similarity Computation

The most crucial component for matching profiles to a JD is the skill or competency equivalence computation module. Going by the adage that *a word can be understood by the company it keeps,* we model this problem as a comparison of two feature vectors which comprise important topics and associated features of the two skills. For comparing and ranking skills, we have created a corpus of the skill definition documents. These skills are represented as a *tf-idf* [4] feature vector of length equal to the size of the vocabulary. We computed similarity between two competencies using cosine distance between their respective feature vectors.

We use the standard way of identifying similarity between two documents d_1 and d_2 by computing cosine similarity between their vector representations $\vec{V}(d_1)$ and $\vec{V}(d_2)$, where "." denotes the scalar product of the vectors $\vec{V}(d_1)$ and $\vec{V}(d_2)$.

$$sim(d1, d2) = \frac{\vec{V}(d1) \cdot \vec{V}(d2)}{\left|\overrightarrow{V}(d1)\right|\left|\overrightarrow{V}(d2)\right|}$$

Some example results are presented in Table 2.

[4] *tf-idf: term frequency-inverse document frequency is a measure of importance of a word in a document belonging to a corpus or collection.*

3.4 JD Extraction

Job Description (JD) document is an important document that describes talent requirement in detail including important aspects, namely, technical skills such as “Java” and “Big Data”, behavioral skills such as “Communication Skill” and “Team work”, service lines such as “Banking” and “Manufacturing”, educational qualifications such as “MCA” and “B.E.”, location of job, and so on. JDs also include desired expertise levels in skills and experience range. To define expertise level, a JD uses various adjectives such as “Extensively Experienced”, “Strong”, “Proficient”, “Good”, “Experienced”, and “Understanding”. A JD typically also has (a) the profile of the recruiting company in the form of statements “About Company”, “Company Strength”, “Working Areas”, etc. and (b) other elements which are divided into the following categories:

- Technical skills (Technological area) which include attributes such as name, experience (in years), expertise level, mandatory or not, etc.
- Educational qualifications which contain degree name, specialization, subject, area, duration of study, part-time or full-time, etc.
- Service lines represent task involved, experience and expertise level in a task category, etc. Behavioral skills along with a description of expertise level, location, and language(s).

The purpose of extracting content from JD is manifold, and the relevant ones are as follows: (i) the structured attributes from a JD can be used for automated matching and ranking candidate profiles (as described in section on RINX-SE) and (ii) the JD information is useful for verification and validation of a submitted JD for the purpose of completing it in all respects. Given a JD, assuming it can be improved, we describe the recommendation methods targeting completion of JD in this section. The completion of the JD would improve its specification resulting in the identification and shortlisting of better quality candidates for hiring.

3.4.1 Extraction of Technical Skill

A technical skill consists of the knowledge and abilities required to accomplish engineering, scientific, or computer-related work, as well as related specific tasks. A technical skill in IT domain refers to an ability to use a specific IT technology platform, software system, software product, framework, programming language, operating systems such as “Java”, “ASP.NET”, “SAP”, “COBOL”, “J2EE”, etc. In a JD, the technical skill requirements are specified in multiple forms possible in language. Few examples are as follows:

(a) 3–5-year experience in “Core Java”, “Spring”.

(b) 3-year experience in “Java”, 2 years of experience in “Spring”.

(c) `At least 2 years of experience in “Hibernate” or related frameworks.`
(d) `Experience in at least (one/two) technology/frameworks “Hibernate”/“JPA”/“EJB”.`
(e) `Knowledge of “Angular JavaScript” is preferred (in this case, the technology is not mandatory).`
(f) `Strong knowledge of “Java”.`
(g) `Proficiency in “Java”.`
(h) `Must have “Core Java” experience.`
(i) `“Spring” and “Hibernate” are must.`

In examples mentioned above, the technical skills are “Core Java”, “Spring”, “Hibernate”, “JPA”, and “EJB”, and keywords like “Strong”, “Experience in”, “Knowledge of”, and “Proficiency in” specify the level of expertise required. We analyzed few hundred JDs to identify a list of cue words which are commonly used to specify the expertise levels. The cue words can help in getting intuition which technology is important over other in the current job description profile. Some cue words such as “Extensive Experience”, “Strong”, and “Must have” indicate that the technologies are of higher importance in the JD and need be assigned higher importance during use in either profile matching or in recommending technical skill to an incomplete JD. Mandatory skills are those which are specifically mentioned in the JD as a “must have” or as a “mandatory skill”. The desired experience in using a skill is mentioned as a numeric value for the number of years as “2–4”, “2”, “2 plus”, “2-4”, etc. The year as a time unit can be written as “year”, “years”, “yrs”, “yrs.”, etc.

To extract fields such as skill, expertise level, and experience from JDs, we use the approach as described earlier in section on RINX. A set of examples of the extraction patterns written and applied to the enriched text are given as example. Input sentence is a text input from which entities and its expertise level to be extracted. In the second step, sentence has been enriched using part-of-speech tagging. On this enriched text, handcrafted patterns in the form of regular expressions described in step 3 have been applied to extract the entities with the expertise level. The extraction results for these sentences are shown in Table 3.

Table 3 Sample entity extraction result

S. no.	Expertise level	Skill name	Experience (in years)
1	Experience	SAP Hana Live	5
2	Good	MySQL, Oracle	–

Input Sentence 1: Minimum 5 year experience with SAP HANA Live content and browser.

Input Sentence 2: Good understanding and hands on experience in MySQL or Oracle.

Enriched Text for Sentence 1: <S><ROOT><S><NP><nsubj_**experience**><NNP>Minimum </NNP></nsubj_experience></NP><NP-TMP><tmod_experience><CD>5 </CD><NN>year </NN></tmod_experience></NP-TMP><VP><VBP>experience </VBP><PP><prep_experience><IN>with </IN><NP><NP><pobj_with><ORGANIZATION><NNP>SAP </NNP></ORGANIZATION><NNP>HANA </NNP><NNP>Live </NNP></pobj_with></NP><ADJP><amod_Live><JJ>content </JJ><CC>and </CC><JJ>browser </JJ></amod_Live></prep_experience></ADJP></NP></PP></VP></S></ROOT><PUNCT>. </PUNCT></S>

Enriched Text for Sentence 2: <S><ROOT><NP><NP><JJ>Good </JJ><NN>understanding </NN><CC>and </CC><NNS>hands </NNS></NP><PP><prep_understanding><IN>on </IN><NP><pobj_on><NN>experience </NN></pobj_on></prep_understanding></NP></PP><PP><prep_understanding><IN>in </IN><NP><pobj_in><ORGANIZATION><NNP>MySQL </NNP></ORGANIZATION><CC>or </CC><ORGANIZATION><NNP>Oracle </NNP></ORGANIZATION></pobj_in></prep_understanding></NP></PP></NP></ROOT><PUNCT>. </PUNCT></S>

Pattern used for Sentence 1:

<CD>([\d\-\s]+)<\/CD>.*?years.*?<\/IN>.*?(<NNP>|<NN>)([A-Za-z\s]+)(<\/NNP>|<\/NN>).*?(Excellent|Understanding|expert|proficient|strong|experience|extensive experience|Knowledge|develop|proficiency|exposure).*?level.*<\/IN>.*?<pobj_in>(.*?)<\/pobj_in>

Pattern used for Sentence 2: (Expert|Strong|Good|Experience|extensive experience|Knowledge|develop).*?(level|skills?)?<\/IN>(.*)(<\/cc_Batch>|<\/[A-Za-z_]+(in|with)>.*?<\/IN>.*?<pobj_[A-Za-z]+>(.*)?<\/pobj_[A-Za-z]+>)

3.4.2 Experimentation and Results

The entities like technical skills, service lines, behavioral skills, location, and language were extracted from the gold copy, wherein the extraction results were post-processed, e.g., standardized using a master list and then manually compared with the content of JDs. The results consisting of precision, recall, and F-measure are provided in Table 4.

Table 4 JD extraction results

Entity	Precision	Recall	F-measure
Technical skill	0.91	0.81	0.85
Behavioral skill	0.96	0.91	0.93
Service line	0.95	0.89	0.92
Location	0.95	0.95	0.95
Language	0.98	0.99	0.98

3.5 JD Completion

The JD completion module generated recommendations for completing the JDs. This can be framed as a problem of mining and learning the most frequent association rules from the training data which are the past JDs. The details of the JD completion module are given in Fig. 4.

As the name suggests, JD completion module tries to *complete the missing information* in case of an *incomplete job description* and makes it more unambiguous in terms of requirements of both technical and behavioral skills. As an example if "`Spring`" technology is present in a JD, then "`Java`" technology is *a must* for the same JD as the Spring framework is built in Java and is used along with Java. There can be many similar examples of such relations as "`JAXB`" (`Java` framework for `XML` binding) and "`Java`" signify use of "`XML`" technology. This problem can be a viewed as identifying the pre-requisite skill names for an advanced technology or frequently used technologies in implementing IT systems. We provide few rules and their examples which guide the completion of a JD:

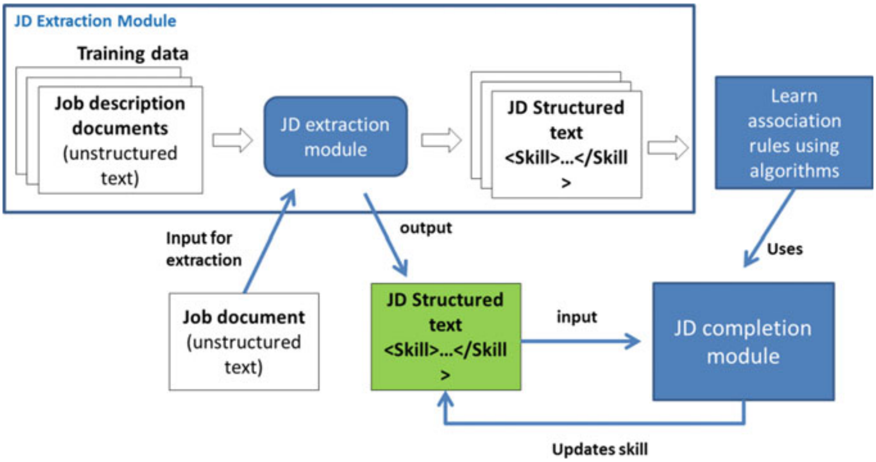

JD Extraction Module
Training data
Job description documents (unstructured text)
JD extraction module
JD Structured text <Skill>...</Skill >
Learn association rules using algorithms
Input for extraction
output
Uses
Job document (unstructured text)
JD Structured text <Skill>...</Skill >
input
JD completion module
Updates skill

Fig. 4 JD extraction pipeline

1. If “Spring” is present in a JD, then “Java” should also be included, as “Spring” is a framework developed in “Java”. Or if “JAX-RS” (Java framework for creating web services) is present, then “Java” and Web services should also be present. Hence, we define “Rule 1” that *if a framework is present then their respective base technology should also be included* in JD.
2. If “Java” is present along with a framework such as “Spring” or “Hibernate”, then their IDE’s like “Eclipse”, “Netbeans” can also be included, as they provide best platform for maintenance of code. Therefore, “Rule 2” can be that *for a framework and root technology, a suitable IDE’s should also be included.*
3. If “Oracle” database and “Java” is present, then “JDBC” should also be present, means if two root technologies are present then their connecting framework should also be present. Hence, “Rule 3” will be *if any two root technologies are present, then their connecting framework should also be included*.
4. Few technologies depend upon operating systems such as “Big Data” and “Hadoop” depends on “Linux” operating system. Hence, “Rule 4” will be *if operating system-dependent technologies are present in a JD then include their respective operating systems*.
5. With technologies such as “Java” or IDE like “Eclipse”, it is good to have knowledge about version control framework. A candidate has to work in a team and a version control tool is generally used to manage the code versions and collaboration. Hence, “Rule 5” will be a *version control framework recommendation based on technologies present in JD*.

The proposed solution approach uses domain knowledge to lay out a structure or template for a JD, and then uses association rule mining to identify the most likely technology or frameworks to fill the placeholders in the template. The current algorithm is based on co-occurrence in the past data and associated domain knowledge about the composition of a JD.

For the frequent dataset mining, the algorithms used are (a) Apriori and (b) Frequent Pattern (FP)—Growth. For the association rule mining, the algorithms applied are (a) TopK Rules nonredundant (TNR) (Fournier-Viger and Tseng 2012) and (b) closed rule discovery using FP close (Grahne and Zhu 2005) for frequent dataset mining. *TopK Rules* algorithm finds Top“K” rules from the training dataset based on the value of confidence; here, we have used another variant of TopK rules, i.e., *TNR (TopK nonredundant rules)* which removes the redundant rules from the association rule list produced in TopK rule algorithm. In this example of a redundant rule, the rule {a} ->{c, f} is redundant given another rule, {a} ->{c, e, f}.

3.5.1 Experimental Results

For completing a JD, dataset of nearly 300 JDs of "Java" and "SAP" has been used for training, and 56 JDs for testing. We have used the following experimental approach to test the mining techniques, for recommending five best matching technology names to a JD using Apriori and FP-Growth algorithm:

1. Generate set of frequent patterns(T_{FP}) using training dataset of JDs using an algorithm.
2. Given a test JD, which contains technology set T = {$tech_1$, $tech_2$, $tech_3$, $tech_4$, $tech_5$}, remove two technologies randomly, denoted by $T_{\mathrm{rem}} = \{tech_3, tech_5\}$, and the set of technology names remaining is denoted by $T_{\mathrm{left}} = \{tech_1, tech_2, tech_4\}$.
3. In T_{left}, create all different combinations of technology set and put in T_{set}, e.g., {$tech_1$, $tech_2$}, {$tech_1$, $tech_2$, $tech_4$, ...}, and so on, starting with the longest subsequence.

   a. Find all the sequences from FP algorithm which have all technologies associated with the test JD mentioned. For example, in sequence {$tech_1$, $tech_2$, $tech_4$}, two sequences such as $T_{list} = \{\{tech_1\ tech_2, tech_4, tech_8, tech_5\}$ and $\{tech_1, tech_2, tech_4, tech_5\}\}$ contain all three technologies.
   b. Make a list of all technologies which are not part of T_{left} but present in T_{list} and count their occurrence. The resultant list of such technologies would be {($tech_8$, *1*), ($tech_5$, 2)}.
   c. Add these technology names to recommendation list (T_{Recomm}).
   d. Repeat from step one with relatively smaller sequences until T_{Recomm} contains at least *five recommendations*.

4. If the technology added to T_{Recomm} matches with any one of the T_{rem}, we mark it as a hit (1) or else a miss (0). The hits are counted for all JDs in the test set as N_{Hits}.
5. Let N_{nRec} represent the count of the JDs for which there are no recommendations and N_{JD} is total number of JDs, then

$$Precision@5 = \frac{N_{Hits}}{N_{JD} - N_{nRec}} * 100$$

The accuracy of the results is measured using Precision@5, which implies the correct matches among the top 5 recommendations. The correct match during testing can be for one or two randomly removed technology names. The results of the correct recommendations against two of the removed technologies are termed as "Horizon = 2" in Table 5.

Table 5 JD completion results

S. no.	Algorithm and setup variables	Hit	Total count	No of Reco.	Precision@5, Horizon = 2
1	Apriori, FP Growth	18	56	–	32.14
2	TNR (top k rules nonredundant) Confidence = 0.8, TopK rules = 2000	12	56	–	21.42
3	TNR (top k rules nonredundant) Confidence ≥ 0.8, TopK rules = 5000	17	56	21	48.71
4	TNR (top k rules nonredundant) Confidence ≥ {0.9–0.4}, TopK rules = 5000	21	49	12	56.75
5	TNR (top k rules nonredundant) Confidence = {0.9–0.4}, TopK rules = 5000	24	55	16	61.53
6	TNR (top k rules nonredundant) Confidence = {0.9–0.4}, TopK rules = 5000	31	49	8	**75.6**
7	Closed rule using FP close	17	54	13	41.46

4 Conclusion and Future Work

In this paper, we focus on the domain-driven text and data mining components to build point solutions to deploy analytics at appropriate juncture in recruitment process. Each such component is designed to address a specific business issue related to efficiency, quality, or cost in TA-related processes. In this paper, we outline text and data mining-based components to build important and mandatory capabilities for enabling efficient and intelligent recruitment.

The future work would aim to assess behavioral attributes to gain deeper insights into the candidate profiles. Also, we would apply information fusion to create comprehensive profile by harvesting data and inputs from social networking and collaboration platforms. The continuous improvement using learning algorithms for better measurement of technology similarity as well as for updating gazettes and ontologies would be pursued. This would be necessary as the evolving technological and business landscape would require continuous mining of the information sources to maintain the readiness of the deployed solutions and capabilities.

References

Aggarwal, C. C., & Zhai, C. X. (2012). Mining Text Data (1st ed.). Springer.

Bondarouk, T., Ruël, H., Guiderdoni-Jourdain, K., & Oiry, E. (2009). Handbook of Research on E-Transformation and Human Resources Management Technologies—Organizational Outcomes and Challenges. IGI Global.

Bui, H. Q., & Zyl, L. T. V. (2016). Talent acquisition gamified: Insights from playing the game at PwC Hungary, Master thesis, Lund University, School of Economics and Management.

Dutta, D., Mishra, S., Manimala, M. J. (2015). Talent acquisition group (TAG) at HCL technologies: improving the quality of hire through focused metrics (p. 22). IIMB-HBP. http://research.iimb.ernet.in/handle/123456789/6698.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM, 16,* 264–285.

Faliagka, E., Tsakalidis, A., & Tzimas, G. (2012a). An integrated e-recruitment system for automated personality mining and applicant ranking, *Internet Research*, **22**(5), 551–568.

Faliagka, E., Ramantas, K., Tsakalidis, A., & Tzimas, G. (2012b). Application of Machine Learning Algorithms to an online Recruitment System. In *ICIW 2012: The Seventh International Conference on Internet and Web Applications and Services* (pp. 216–220). IARIA.

Fournier-Viger, P., & Tseng, V.S. (2012). Mining top-K non-redundant association rules. In *Proceedings of 20th International Symposium on Methodologies for Intelligent Systems (ISMIS 2012)* (pp. 31–40). Springer, LNCS 7661.

Grahne, G., & Zhu, J. (2005). Fast algorithms for frequent itemset mining using fp-trees. *IEEE Transactions on Knowledge and Data Engineering, 17*(10), 1347–1362.

Mooney, R. J., & Bunescu, R. (2005). Mining knowledge from text using information extraction. *ACM SIGKDD Explorations Newsletter, 7*(1), 3–10.

Parthasarathy, M., & Pingale, S. (2014). Study of talent acquisition practices—A review on global perspective. *International Journal of Emerging Research in Management & Technology., 3*(11), 80–85.

Patil, S., Palshikar, G. K., Srivastava, R., & Das, I. (2012). Learning to rank resumes, FIRE 2012: In *Proceedings of the 4th Annual Meeting of the Forum on Information Retrieval Evaluation*. ISI Kolkata, India.

Ronen, F., & James, S. (2006). The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data (Vol. 1). Cambridge University Press.

Schiemann, W. A. (2014). From talent management to talent optimization. *Journal of World Business., 49*(2), 281–288.

Srivastava, R., Palshikar, G. K., & Pawar, S. (2015). Analytics for Improving talent acquisition processes. In *International Conference on Advanced Data Analysis, Business Analytics and Intelligence, ICADABAI 2015*. IIM, Ahmedabad, India.

Strohmeier, S. (2007). Research in e-HRM: review and implications. *Human Resource Management Review, 17*(1), 19–37.

Téllez-Valero, A., Montes-y-Gómez, M., Villaseñor-Pineda, L. A. (2005). Machine learning approach to information extraction. *Computational Linguistics and Intelligent Text Processing*, 539–547.

Hastie T., Tibshirani, R., & Friedman, J. (2008). Elements of Statistical Learning (2nd ed.). Springer.

Tomassetti, F., Rizzo, G., Vetro, A., Ardito, L., Torchiano, M., & Morisio, M. (2011). Linked data approach for selection process automation in systematic reviews. In*15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)* (pp. 31–50).