Adaptive Employee Profile Classification for Resource Planning Tool

Tere Gonzalez, Pano Santos, Fernando Orozco
Information Analytics Lab
HP Labs
Palo Alto, Ca, US
{maria-teresa.gonzalez.diaz, cipriano.santos, fernando.orozco2}@hp.com

Mildreth Alcaraz, Victor Zaldivar, Alberto De Obeso, Alan Garcia
Depto de Electronica, Sistemas e Informatica
ITESO
Tlaquepaque, Jalisco, Mex
{Mildreth,victorhugo, Alan,ado}@iteso.com.mx

***Abstract*— Matching the right people to the right job considering constraints such as qualifications, availability and cost is the cornerstone of IT projects delivery services. We present a study to improve data accuracy and completeness for resource matching by integrating unstructured data sources and introducing text mining techniques to dynamically adapt resource profile for resource planning decisions. Our approach discovers resource categories by extracting and learning new patterns from employee resumes; and incorporating resource experience for the job-matching optimization during the resource planning exercise.**

Resource planning; text mining; supervised classification; information extraction;nlp; nltk; ontology; relation discovery; bag of words, naïve Bayes classifier;optimization.

I. INTRODUCTION

Matching the right people to the right job considering constraints such as qualifications, availability and cost is the cornerstone of IT projects delivery services. Talent management represents an important technological differentiator for IT companies in which one of the main sources of competitiveness comes from the talent of their workforce. Large IT companies have a variety of service lines and project portfolios along with continually emerging technologies making difficult to identify qualified resources that can satisfy all project needs and preferences. In addition, employees constantly acquire new skills and experiences that makes difficult to track and evaluate employee's qualifications and proficiency.

We have built a Resource Planning (RP) system that optimizes the matching of resources and potential project positions while considering resource capabilities, availability and training opportunities[1]. Currently, RP uses structured data from different sources which can adversely impact the accuracy and completeness of the professional's background for resource-job matching, especially for training and experience tracking. The pre-defined structured data of the current RP system faces the limitation of containing only partial information. Additional challenges of the system are the amount of effort required for the data integration and standardization.

In this paper we address these problems by an approach that integrates various text mining algorithms to discover useful information for resource planning decisions. First, in section 2 we introduce the RP system and overall architecture design for unstructured data sources integration. Next, in section 3, 4 and 5 we present the text mining techniques explored for information extraction, classification and adaptive learning. Finally, in section 6 and 7 we presented our current results of the resume composition, classification and the impact of the resource planning optimization.

II. TEXT MINING APPROACH FOR RESOURCE PLANNING DECISIONS

The Resource Planning system is a decision support tool for project based services –IT services in particular, that considers employee profiles and job requirements to optimize the tradeoff between demand fulfillment and employee utilization. The RP system assumes a well defined set of attributes that characterizes the employee profile and job requirements. These attributes describe the resource qualifications and availability to satisfy the main goals of resource planning: right resource, right time and right location. First, the tool captures technical employee profile that enables the resources to perform a job, such as capabilities, technologies, job level, industry domain, role, etc. Second, the tool uses organizational attributes that link resource availability over time, business segments and locations, such as geography, site, business unit, etc. Then, the optimization component evaluates the resources qualifications and availability related to the pre-defined job attributes along with business preferences and labor demand priorities.

In practice, however, there is often important unstructured information such as professional experience, personal background, and other qualitative information that are currently not used as part of the planning activities in the RP tool, but this data can enrich the quality and the quantity of the supply and demand matching.

A. Related Work

Generally, unstructured data are used for resource recruitment or search in tools such as Linkedin.com, Monster.com, or Taleo.com[2][3][4], but none of them used for planning purposes. In the other hand, there are software products for resource planning such as HP PPM, Oracle Primavera, or SAP Success Factor that does not have included unstructured data for planning decisions. Nevertheless, important studies from IBM research has

978-0-7695-4770-1/12 $26.00 © 2012 IEEE
DOI 10.1109/SRII.2012.67

developed and integrated algorithms for improving accuracy with unstructured data in solutions such as SPSS Modeler and Optimatch [5]. For instance, the Optimatch framework extracts required job skills from resumes and selects the useful descriptions, measuring how well the resume fits a position that is integrated into the matching criteria on the workforce optimization model. Another relevant IBM study evaluates machine learning methods to match job-categories based on employee directories and organizational data focused on large scale problems [6].

Therefore, given the potential opportunity to enrich data completeness, flexibility and expressiveness by integrating unstructured information into the RP Tool, we propose a text-mining approach to directly impact in the quality of the employee-job matching for resource planning.

B. *Text Mining Integration Approach*

Fig.1 depicts our study that comprises three main components: Information Extraction, Adaptive Learning and Resource Classification.

1) Information Extraction: Knowledge is extracted from resumes that represent one of the most detailed sources of professional experience. The information extraction (IE) is focused on discovering technical composition from the resume content used to identify employee classification. One of the most important challenges is that resumes have different structures and styles that makes difficult to extract relevant information of the employee profile.

2) Adaptive Learning: The Adaptive Learning (AL) module supports the IE module to dynamically update the knowledge base by discovering new information and patterns from the resumes. The AL module builds an IT domain-oriented ontology to analyze semantic relationships of the technical attributes and enhance the resume extraction for the resource classification algorithms. One of the most important challenges is that the rapid growth of emerging technologies, tools and skills that makes difficult to have an up-to-date system to accurately detect technical profile trends.

3) Resource Classification: Resource classification is a key module that represents the connection between the resume analysis and resource-job matching optimization. This module uses the extracted resource composition to identify in which classes the employee is categorized for each technical attribute. One of the most important challenges is to predict the best fit into the RP categories based on the job history.

The following sections describe in detail each component.

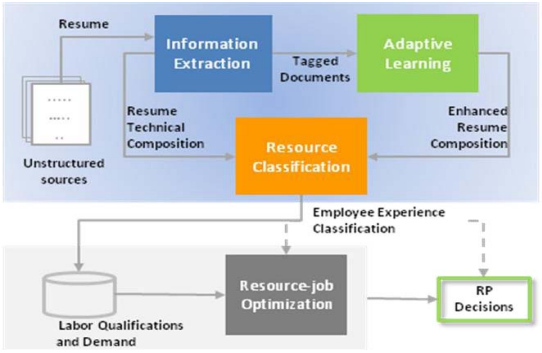

Figure 1 Adaptive Resource Classification Approach

III. INFORMATION EXTRACTION

The Information Extraction (IE) Module is part of the generalized Text Mining process, which is usually followed by a mining process [7][8]. The goal of our IE stage is to obtain structured and relevant information related to programming languages, roles, industry domains, technologies, tools and their relationship with time. We focus our study on resumes written in English and Information Technology domain.

A. *IE Framework*

The framework used in the IE Module is shown in Fig. 2. It comprises five stages in order to extract the target information from the resume documents: Preprocessing, Tokenizing, Tagging, Chunking (entity detection) and Relation Extraction Stages.

1) Preprocessing Stage: The main input is the resume collected in plain text format. In this stage, every document is read and converted into a string. Then, the spelling corrections take place, e.g. Javascrpt is corrected as javascript. After that, the string is split into single sentences. In this step, we decided to keep together the sentences, since it is important to maintain the context of the phrase. The output of the stage is a set of corrected sentences for each input document. The details of the stage are shown in Fig. 12 (a).

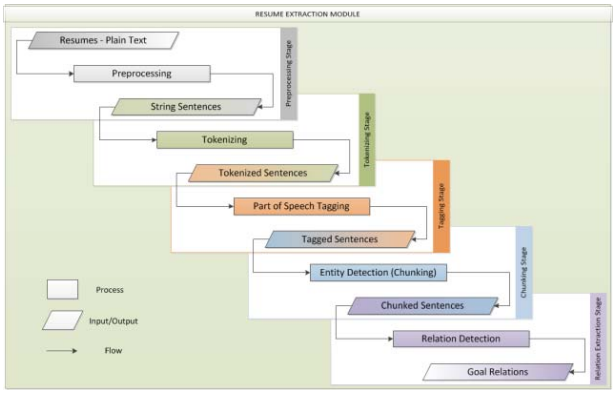

Figure 2 IE Framework

2) Tokenizing Stage: In order to setup the most adequate tokenizer algorithm for our objective, we must create first our token pattern. The token pattern express the set of words that belongs to the language we want to create for our purpose, i.e., how we want to separate the sentences into single words. There are some special considerations we should do due to the IT context of the documents and typical writing in this field, e.g., skill and tool versions usually are separated by a slash, comma, or hyphen. Then, we create the tokenizer using the token pattern, which finally is used for tokenizing the input of this stage that is the set of corrected sentences for each input document, as shown in Fig. 12 (b).

3) Tagging Stage: This is a main and a very sensitive stage, since at this stage we aggregate the intrinsic dictionaries including the keywords of interest, e.g., roles, technologies, programming languages, and so on. Initially, we create a tagged pattern which considers the word meaning (token) in the general context. Using this pattern, then we create the tagger to apply it to the set of tokenized input sentences, as the first tagger of a sequence of three. At this step, we use the intrinsic dictionaries to create a Tag Model. This model and the set of tokenized sentences are the input of the unigram tagger, to finally apply a default tagger to tag any word out of this knowledge. The output of this stage is a set of tagged sentences for each target document. Details are shown in Fig.12 (c).

4) Entity Detection Stage: This stage is also known as chunking. Chunking is referred to joint tokens into a short phrases form. This process looks at finding patters that belong to an entity class. This process helps to understand the entity relation that the tokens have in the sentences. See details in Fig. 12 (d).

5) Relation Extraction Stage: This stage is related to the extraction of relationships between the named-entities identified in the previous stage. For example, in "20 years of business success as Programmer/Analyst", we want to relate "20 years" with "Programmer" and "analyst". The first step in this stage is creating the target relations, and then extracting these relations from the previously chunked sentences.

An example of the output of Resume Extraction Module is shown in Fig. 3.

B. Experimentation

We have randomly selected 200 resumes as sample dataset with plain text format.

First, we analyzed the resume composition extracting information using two different technologies: NLTK with Python [9] and Rapid Miner [10]. RapidMiner (RM) is a data mining tool that can be leveraged for text mining endeavors with the Information Extraction and Text Processing plug-in extensions [17]. RM uses a powerful visual programming paradigm in which operators are dragged-and-dropped into a canvas, connections among them are made and custom behavior is achieved through model configuration.

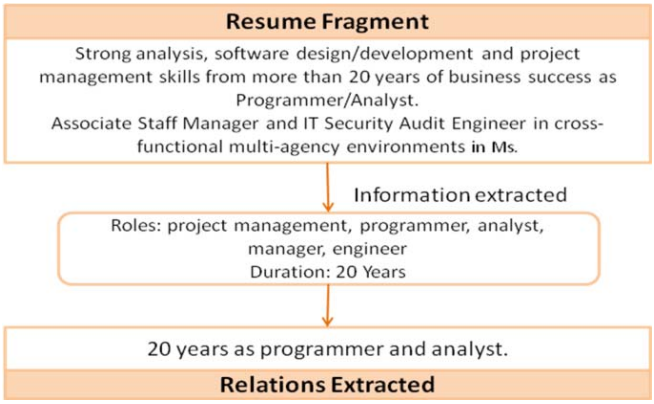

Figure 3 Example of the output Resume Extraction Module

Nevertheless, when a higher degree of control is required the task becomes harder. RM presents clear advantages considering the visualization of data. On the other hand, the NTLK with Python represents a flexible technology to generate complex patterns that can be customized by the developers for robust future implementations.

Second, the extraction provides visibility to establish the possible methodologies that can improve the classification and adaptive learning modules. Similarly, we found that it is difficult to identify entities comprised by more than one word: roles, technologies and domains (an evidence of this phenomenon is the low frequency in which these entities are found and the lack of precision). We are planning to address this problem enhancing Entity Relation methods in a future research.

IV. Adaptive Learning

As we have seen, the information extraction process uses dictionaries in order to tag entities and perform some basic chunking. Thus, if we have better and up-to-date dictionaries, we will obtain better results from this process. However, this is a difficult task in such a dynamic domain as IT where every year there a lots of new tools, methodologies and technologies that become available, as well as new terms that acquire new meanings related to IT ("cloud", "social network", "agile", to name but a few). On the other hand, when dealing with natural language, there is often more than what is written. For instance, when one is reading a resume, one can easily infer than someone has a senior level qualification for a job if there are several years of experience in similar jobs. Similarly, one can infer that someone may not be a good candidate if he or she keeps changing jobs several times in a year. As these examples show, one can use inference in order to obtain some structured information needed for resource planning, but also, there is some qualitative information that can be inferred and that is also important for this process.

The Adaptive Learning module has two main objectives: first it must use semantic data to enhance the information extraction and the classification process. Second, it must use the information extraction results to discover new terms and

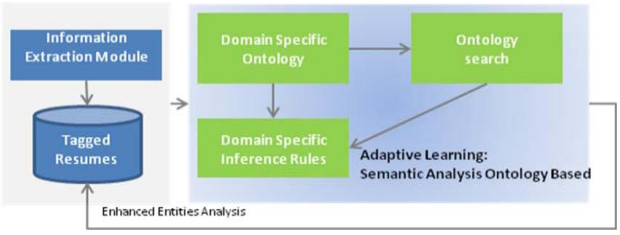

Figure 4 Adaptive Learning Framework

new semantic relationships between them. Fig. 4 shows the overall framework for the semantic analysis and the new pattern recognition.

A. *Domain Specific Ontology*

As part of the AL framework, we are building a domain-oriented ontology. Fig. 5 shows a simplified image of our representation. This ontology will allow us to represent knowledge in a more structured and complex way than the dictionaries. We will be able to represent the tags (used in the information extraction process) but also some tags properties and, more important, some tags relationships with other tags. This representation will allow us to add structure to the tags dictionaries, for instance by defining tags classes and categories and organizing them in a domain specific taxonomy.

We are using Protégé-Frames [11], because we will represent each of the tokens in the dictionaries used by the IE module as a Frame so that we can add slots that represent token relationships and token attributes. Fig. 5 shows a simple view of the ontology generated by Protégé-Frames. In this Fig. 5 we have exemplified the description and relationships of "Web Development". We can see that "Web Development" is a "Skill" that uses technologies such as "JSP", "ASP" and ".NET" (which are represented as instances of "Technology"). We can also see that "Web Development" uses "Programming Languages" such as "Java", "C#" and "VB". We can store the ontology as RDF [11].

B. *Inference Rules Design*

Once the knowledge is organized in the ontology, we will be able to define inference rules that allow obtaining more information based on tag relationships. The extracted entities will imply rules such as association, equivalence, dependence, etc. For example, table 1describes types of derived association rules.

TABLE 1 INFERENCE RULES EXAMPLES

Domain-Specific Inference Rues	
Rule	***Description***
$\text{ProgramLanguage}(C\#) \Rightarrow \text{Technology}(Web)$	The technology is infered based on the programming language.
$\forall x Experience(x) \wedge x = 5 \Rightarrow SkillLevel(Senior)$	The level of expertise is infered based on the years of experience.
$\forall x Experience(x) \wedge x < 5 \Rightarrow SkillLevel(Senior)$	Infering expertise level based on the years of experience

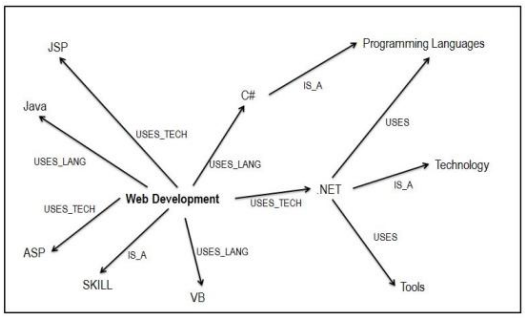

Figure 5 A simple view of ontology for resource profile

C. *Ontology Search*

Having the ontology will allow us to develop a strategy to in order to automatically maintain it up-to-date. This mechanism will be triggered by the presence of a "entity" that cannot be recognized as meaningful for classification purposes. Whenever this happens there will be an external search in a broader domain ontology such as DBpedia[13], WordNet[14] of OpenCYC[15], in order to obtain some semantic characteristics of the token. We will use the information obtained in order to add the new tokens, its attributes and relationships in our ontology.

D. *Experimentation*

We are currently developing the ontology and inference engine using Stanford's Protégé Frames and developing some scripts that implement the inference rules described above. The next step in the development of the Adaptive Learning Module will be to build the scripts that allow us to query external broader domain ontology and modify the ontology according to the information obtained by the query.

V. RESOURCE CLASSIFICATION

As part of the mining process, the solution explored a supervised text classification process that helps on categorizing a document in known classes. The classification module takes the extracted data from the resumes prepared in previous stages to predict classes for technologies, industry domains, roles and job levels attributes.

A. *Resource Classification Framework.*

The process uses the information that reflects directly the employee's professional experience to discover the classes that are relevant for the resource planning allocations. For instance, web, windows, mobile and business intelligence are examples of technology classes; whereas manufacturing, banking and healthcare are examples of industry domain classes. Each of these RP classes comprises specific technical experience that can be learned from the resumes composition. For example, an employee who has experience on web technology may include some or all of the following items: a) technical skills in web services and client-server applications, b) programming languages like JavaScript, Asp.net, Php or Html; c) tools like IIS, AJAX, Tomcat or Apache. Therefore, the goal is to identify the class in which

the resume may fit based on the technical content as is described in Fig. 6.

B. *Classification Model*

The approach introduces the naïve Bayes classifier which is a simple and popular method for text mining applications based on the document content [6][16]. The naïve Bayes classification assumes a feature set input to predict the probability of a document d_i belongs to a class c_k. The feature set extraction process is performed by module 1 and 2 to construct a bag of words as feature set which characterized the words that were found in the document. Then, assuming x_i as a vector of the found words, the classifier predicts the probability of the vector x_i belongs to the c_k by applying Bayes Rule and conditional independence property of the vector elements as follows:

$$P(c_k|\vec{x}) = P(c_k) \times \frac{\prod_{j=1}^{n} P(x_j|c_k)}{P(\vec{x})} \qquad (1)$$

Where $P(c_k|x)$ is the estimated probability given that $P(c_k)$ is the class probability determined by the m documents in the training set; $P(x|c_k)$ is the prior probability of the feature set being classified as c_k and $P(x)$ is the likelihood of a random feature set being occurring in the training set.

C. *Experimentation*

Naïve Bayes classifier has been selected among others due to the advantage of the small amount of training set required for the parameter estimation[16]. In our current analysis, the naïve classifier is trainned using a bag of words considering three different scenarios of feature selection to evaluate precision, recall and F-mesuare metrics. As part of the experimentation, NLTK library has been used to evaluate a small data set. NLKTclasses provides a naïve classifier object to train, classify, obtain probability classification and metrics.

VI. CURRENT RESULTS AND FUTURE WORK

A set of experiments have been designed to evaluate text mining models and verify the potential benefit of using resumes by augmenting technical expertise information for the resource-job matching.

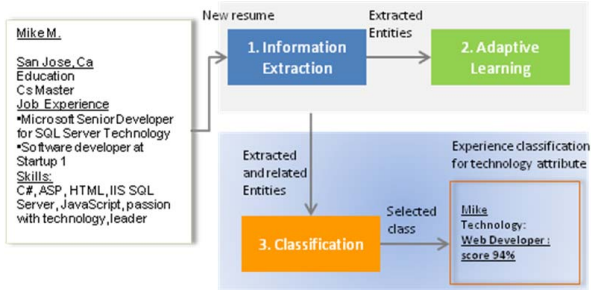

Figure 6 Resource Classification Framework.

Our current results are described as follows. Sections A, describe our results in terms of evaluating the text mining discovery to ensure that the information extracted can be useful for the resource planning decisions. In section B, we present the current results of the classification model that confirms the benefit of our approach comparing precision and recall metrics. In section D, we present a simulation study of the benefit on resource utilization and demand fulfillment when we will incorporate the text mining results, since the whole approach has not been integrated yet.

A. *Resume Compostion Analsys*

1) Repetition Pattern: Frequently, resume content includes a historical description of different jobs that the employee has performed; this history of jobs provides an indication of the work experience of an employee. Since employees inherit the knowledge and skills job by job, a common pattern of entities repetition was found in the technical background. Fig. 7 describes two main types of results: frequencies of entities with repetition, and frequencies of unique entities extracted from our sample data set of 200 resumes as described on section 3A. As the chart shows, the repetition pattern is common for all the entity categories.

For example, the most common entity found is role -- developer, manager, engineer, etc. with a total of 2146 words found and 38% of repetition rate. On the other hand, the highest repetition rate is found in programming languages with a 47% with a total of 1459 words. The repetition rate helps to analyze the relevance of the skills. The repetition discovery represents useful information that will facilitate the classification problem for the skills – technologies, programming languages, and tools to use frequency oriented classifiers. However, the role repetition rate represents a challenge; since the total of roles is considerable small (our current dictionary has about 40 different roles). The role repetition may be used for relevance, but the classification will find problems determining the best fit. Specially for trade-offs over recent role versus more frequent. These challenges will be addressed on a future research. A different pattern is found in job levels which the number of extracted entities was considerably small.

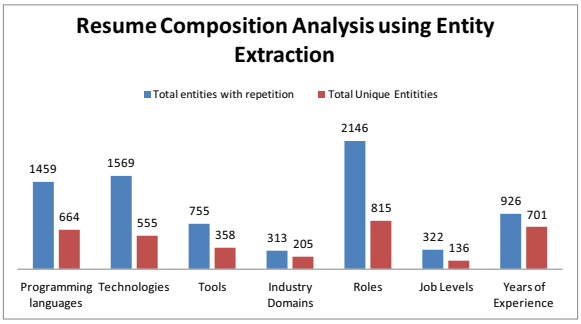

Figure 7 Resume Composition Analysis

This pattern opens a research topic for the adaptive learning module that will require enhancing the extraction by inferring seniority using years of experience data points. Industry domains entities are also a weak element that required further investigation.

Industry domains are important for the RP Tool which has visibility of open positions from different industries and portfolios that are difficult to fulfill. We will explore identifying relationships between companies and products to discover related job industries.

2) Unique Entity Pattern: A second analysis derived by the extraction process is the understanding of unique entities occurrence distribution. We refer as unique entities to the distinct words found on the categories for all the resumes. Fig. 8 shows that most of the resumes include unique entities between 2 and 6. While Fig. 9 shows that the entities related to proficiency are following a different pattern. Basically, the pattern shows that the number of entities found is between 1 and 2. This result helped us to detect a skew in the dataset which indeed is demonstrated in the pattern. It was found that most of the resumes describe the job experience into 2 buckets: "current" and "past experience", and this is the reason why extraction detected this pattern.

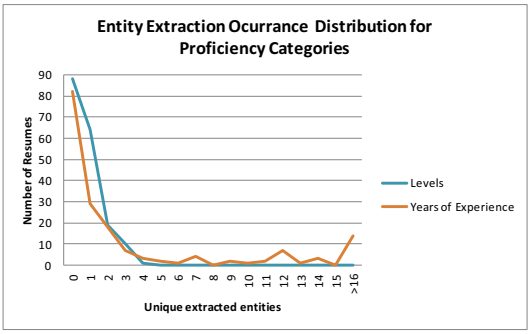

Figure 8 Unique Entities Distribution

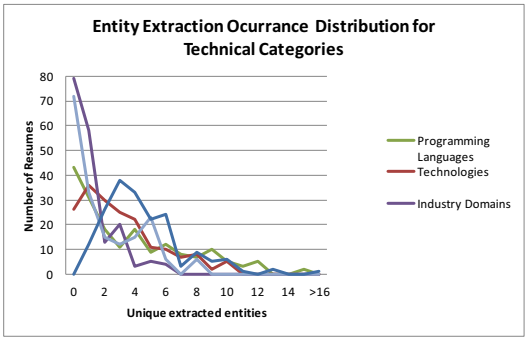

Figure 9 Unique Entities Distribution

B. Classification Results

As example for the classification problem, we have designed experiments to evaluate the quality of the classification for a Web Technology category. Technology classes are an interesting case due to they have strong relationship to several entities in the document content, such as tools, frameworks, and programming languages skills.

The experiments include three main scenarios:

1) Baseline scenario: The classification is executed over the full text of the resume. The bag of words is created using all the words that the resume contains.

2) Feature selection using IE Module: The bag of words is created by adding only the technical words extracted about programming languages, tools and technologies key words by the IE Module.

3) Feature selection enhanced by augmenting key words for IE Module: The bag of words is enhanced adding 1% more key words in the dictionaries for the extraction. This process will be done automatically by the AL module once is completed.

For the scenario 2 and 3, table 2 describes a sample of the training set used for the classification. The feature set column represents the words extracted and used to build the bag of word representation for the naïve Bayes classifier. The class column is the manual classification done by a subject matter expert (SME).

The results of the classification demonstrate the advantage of integrating the information extraction and adaptive learning modules before executing the classification model. The table 3 shows that the precision of the resume classification is about 51% using the full content. On the other hand, adding a specialized information extraction to enhance the feature selection can improve the precision to 76%. Furthermore, incrementing 1% of relevant key words to the feature selection results on 2% increment of precision.

TABLE 2 TRAINNING SET FOR RESUME CLASSIFICATION PROBLEM

Resume	Feature set			Class
	Programming Languages	***Tools***	***Technology***	***Is Web tech-nology?***
Cv1	vb.net,vbscript, asp, c#, php, javascript, javascript, asp.net 3.5	IIS, VS 2010	IIS	yes
Cv2	javascript, perl, php, asp, asp.net	Tomcat, Ajax	Web application	Yes
Cv3	java, object, java, postscript, java	NetBeans, Ant	Embedded systems	No
Cv4	natural, clean, visualbasic, c standard	Windows		No
Cv5	Java, perl, c++	Windows		No

TABLE 3 CLASSIFICATION METRICS RESULTS

Scenario	Classification Metrics					
	Web Technology Class			*No Web Technology Class*		
	Precision	*Recall*	*F measure*	*Precision*	*Recall*	*F measure*
Baseline	51%	100%	68%	100%	13%	23%
Feature Selection with IE	76%	100%	86%	100%	27%	42%
Feature Selection with IE+AL	78%	100%	89%	100%	33%	50%

We plan to update our key word dictionary automatically by the Adaptive Learning Module to impact on the precision classification as is demonstrated in our experiments. We expect that AL module will enrich the key words based on the ontology and inference rules designed.

Another observation is the precision behavior in which the resume is not classified into the web category. The resume that does not match with the web technology has very high precision due to the nature of the training set. We found that most of the resumes that are not in a web category contain rarely technical content related to web. For instance, those resumes are either networking or embedded technologies making easy the classifier prediction.

Finally, we concluded that this classifier is particularly useful for discovering the skill content, but it may be not an accurate method to other technical attributes such as role or job level. For example, this method does not produce accurate results for this attributes, because a role or level has specific properties in which the relevant category is related to chronological ranking rather than only frequency. This topic will be addressed in a future research.

C. *Resource Planning Impact Results*

Assuming that process of text-mining has accurately identified relevant information for resource profile as was described on our previous sections. Now the goal is to use the discovered knowledge and the classified employee experience for suitable allocation decisions in the optimization component.

Our current optimization model uses only current job description that is maintained as part of the resource management process; whereas the employee experience is complicated to maintain and track. This research has aim to understand the best approach to leverage resume information in terms "Experience Score" from the job history. Experience score is a measure to represent the relevance of the past experience computed by a combination of classification and timeline ranking model. The score is used in the optimization model to determine the best resource allocation based on qualification, experience and availability constraints. For instance, let's assume that resource A has 75% of Web Development experience comparing to resource B who has 90%. The optimization can select resource B if both are equally available and qualified.

We have implemented a simulation algorithm to execute experiments that demonstrate the benefit of using the resource experience that was ignored due to lack of structured data sources. The experiments measure the impact over two main RP metrics: Resource Utilization and Demand Fulfillment when the optimization model uses resource experience. The simulation uses realistic data generated from old datasets of the RP Tool. The algorithm introduced random variables to perform changes on demand, qualifications and availability. The simulation consisted of two main approaches:

1) Demand Growth Simulation

This simulation consists of five cases that increment the demand from 1000 positions to 1500 i.e. 10%, 20% 40% and 50%. We have established a baseline case that isolated a small problem where 1413 employees are available who may be allocated from 1000 to 1500 positions for a planning horizon of 8 weeks. Fig. 10 depicts the impact when we introduced a 20% randomized increment on resource qualifications based on the classified experience. The baseline case finds low utilization due to excess of resources for the expected demand, and then the experience benefit is minimal --around 0.26% of utilization and 0.3% of demand fulfillment. On the other hand, the experience introduction can help increasing up to 3% of demand fulfillment and 2.5% in resource utilization for case 3 and 4, where demand and availability are very close (1413 resources for 1400 to 1500 positions). The chart shows the trend that confirms our theory of the benefit. Resource experience has a high effect when the dynamic of the company tries to keep very close demand fulfillment and utilization metrics, because the model requires more information to select the right resource and minimize the unfilled demand.

2) Availability Reduction Simulation

This simulation consists of four cases that reduce the availability from 1413 resources to 1059, i.e. 15%, 20%, and 25%.

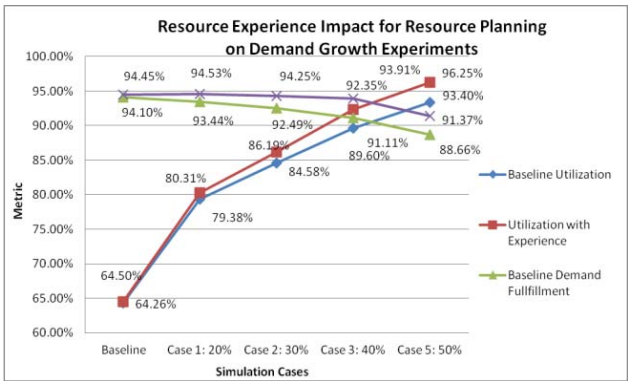

Figure 10 Demand Growth Impact

We have established a baseline case that isolated a small problem where 1413 employees are available who may be allocated to a given 1000 positions for a planning horizon of 8 weeks. Fig. 11 depicts that the baseline case finds low utilization due to excess of resources for the expected demand, and then the experience benefit is minimal – around 1% in utilization and 0.8% in demand fulfillment. In cases 3 and 4, the chart reflects the impact of the 20% randomized increment of experience helping up to 7.5% of demand fulfillment and 9% in resource utilization given that demand and availability are very close (1059 resources to 1000 positions). The chart shows how experience information consistently helps when there is a stretch in resource availability as was found on demand growth case.

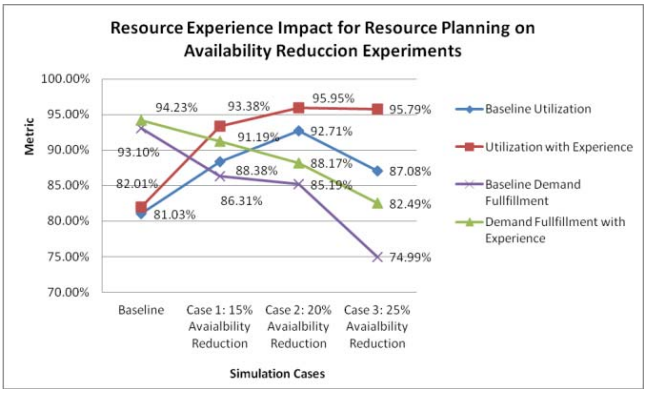

Figure 11 Availability Reduction Impact

VII. CONCLUSIONS

This paper presented a study of introducing text mining approach to classify employees using resumes in order to improve the resource utilization and demand fulfill metrics in resource-job matching in the RP tool. Initial results have derived potential benefit and future research challenges towards integrating a robust text mining solution:

First, our framework minimizes the system integration and manual efforts to keep tracking of relevant employee background in the enterprise while automatically discovering and integrating technical profile for the resource planning optimization. Our approach addressed the quality of the text mining process by proposing an experience score as a mechanism to measure the relevance of the job history for allocation decisions. We have identified that the experience score has high impact when the demand and resource availability are very close. Our simulations have reported up to 9% improvement of resource utilization and 7.5% in demand fulfillment. In these cases, the optimization model has successfully explored resource experience alternatives in order to find and allocate resources to a job instead of reporting unfilled positions or recommending hiring.

Second, the classification results have highlighted the importance of building specialized extraction module which increases up to 20% of the precision. Our framework detects relevant information instead of using the whole document text to feed the classifier model. The information Extraction module implements specific requirements derived from the RP Tool in order to aim on the selection of the right resources in terms of technical capabilities for the jobs. The Adaptive Learning module complements the extraction module adding related concepts to the technical attributes: skills, role, level, industry domain and technologies. The AL enriches the collection of words using a domain specific ontology to improve further the classification accuracy as was presented in our results.

Third, our extraction and classification experiments have discovered important requirements of mining employee profiles to enhance the classification and the resource-job matching quality. The key requirements to be addressed in a future research are:

- Document Structure. The IE mechanism will examine algorithms to take into account documents formats instead of using plain text.
- Chronological Analysis. The classification model must consider time variables to determine the relevance of employee experience.
- Classification Technical Attribute based. Each technical attribute i.e. skills, job level, role, industry domain, or technology requires different classification model to improve precision and recall metrics.

Finally, our study has identified opportunities to use this approach in other areas of resource planning beyond IT services, such as Airlines Industry or Healthcare. We will explore them once this solution is implemented in the forthcoming months.

ACKNOWLEDGMENT

Thanks to Shailendra Jain for his management support in this study; Alex Zhang for his help on the results review and discussions; and Aldo Garcia for his rich discussions and insights on the classification problem.

This project was partially supported by CONACYT under the program Estímulo a la Investigación, Desarrollo Tecnológico e Innovación N°. 155874.

REFERENCES

[1] Santos, C., T. Gonzalez, H. Li, K.-Y. Chen, D. Beyer, S. Biligi, Q. Feng, R. Kumar, S. Jain, R. Ramanujan, A. Zhang (2011), *HP Enterprise Services Uses Optimization for Resource Planning*, accepted for publication in INFORMS/Interfaces.

[2] Singh A., C. Rose, K. Visweswariah, V. Chenthamarakshan. "PROSPECT: a system for screening candidates for recruitment". Proceedings of the 19th ACM international conference on Information and knowledge management. 659-668. 2010

[3] LinkedIn Web Site, "Skills and Expertise", 2012 . http://www.linkedin.com/skills/?trk=skill-rank

[4] Monsters Web Site, "Advance Search", 2012, http://jobsearch.monster.com/AdvancedSearch.aspx

[5] Richter, Yossi; Naveh, Yehuda, Gresht, Donna L;.Connorst, Daniel P "Optimatch: Applying Constraint Programming to Workforce Management of Highly-skilled Employees", Service Operations and Logistics, and Informatics, 2007. SOLI 2007. IEEE.

[6] Yan Liu, Zhenzhen Kou, Claudia Perlich and Richard Lawrence "Intelligent System for Workforce Classification", in KDD 2008 Workshop on Data Mining for Business Applications.

[7] Raymond J. Money and Un Yong Nahm, *Text mining with information extraction*, Multilingualism and Electronic Language Management: Proceedings of the 4th International MIDP Colloquium, September 2003, pp. 141-160.

[8] M. Grobelnik and D. Mladenic, "Text-Mining Tutorial". In the Proceeding of Learning Methods for Text Understanding and mining, Grenoble, France, January 26-29, 2004.

[9] Perkins, Jacob "Python Text Processing with NLTK 2.0 Cookbook", PACKT Publishing, 2010.

[10] Mierswa, I .,Wurst, M., Klinkenberg, R., Scholz, M., Euler, T (2006) "YALE: Rapid Prototyping for Complex Data Mining Tasks", in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06).

[11] Stanford University. "What is Protégé-Frames". 2012 http://protege.stanford.edu/overview/protege-frames.html

[12] W3C, "RDF Primer". 2004. http://www.we.org/TR/2004/REC-rdf-primer-20040210/

[13] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, Sebastian Hellman "DBPedia – A Crystalization Point for the Web of Data", Journal of Web Semantics: Science, Services and Agents on the World Wide Web, issue 7, pp. 154-165, 2009

[14] George A. Miller, WordNet: A Lexical Database for English. Communications of the ACM Vol. 38. No. 11, pp. 39-41 (1995)

[15] Cycorp Inc. "About OpenCYC".2012. http://www.opencyc.org

[16] Tsuruoka, Yoshimasa, Tsujii, Jun'Ichi "Training a naive bayes classifier via the EM algorithm with a class distribution constraint", Proceeding CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4.

[17] Jungermann, F. (2009). Information Extraction with RapidMiner Artificial Intelligence Group, TU Dortmund, http://www-ai.cs.tu-dortmund.de

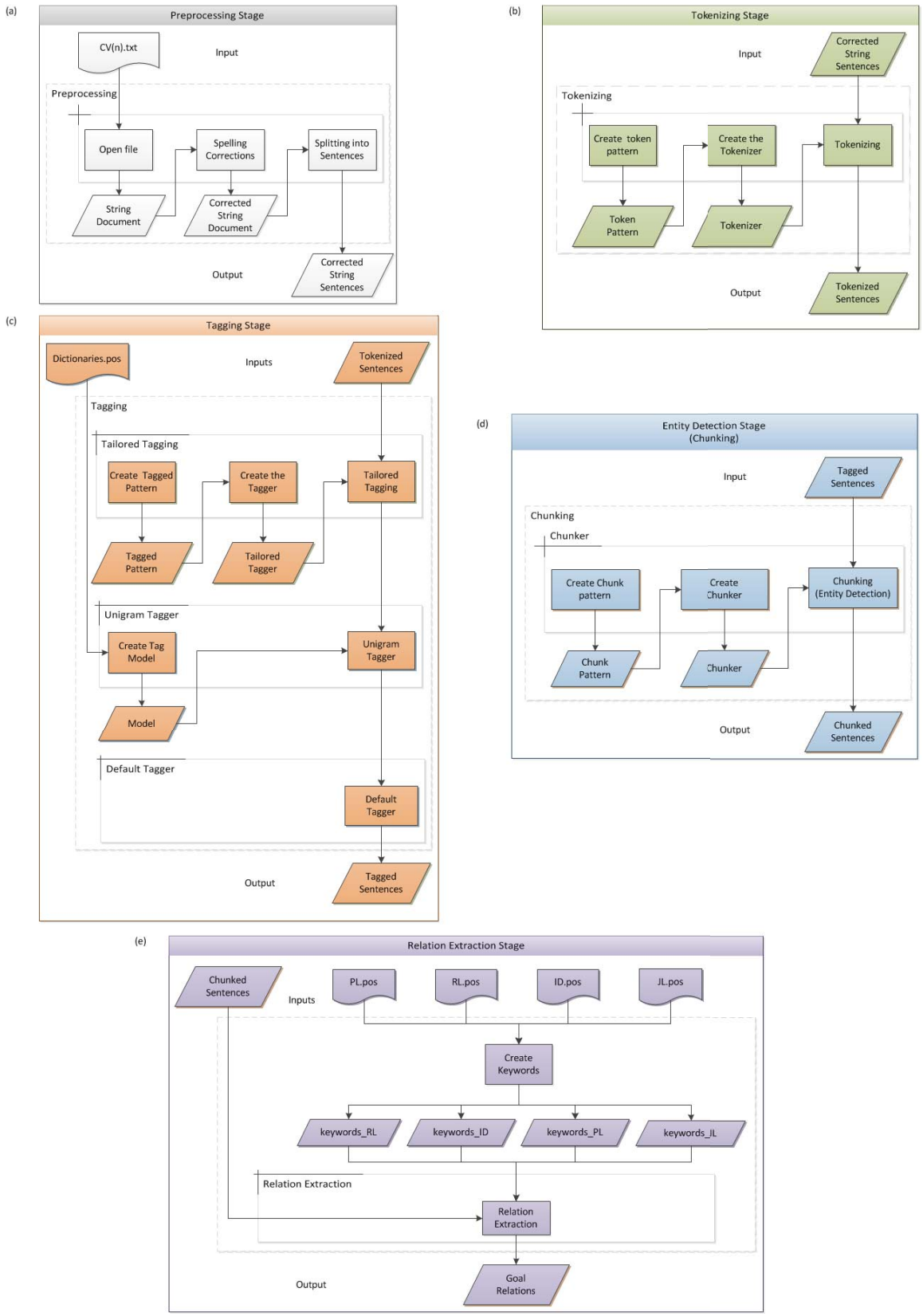

Figure 12 Information Extraction Process