An Approach to Extract Special Skills to Improve the Performance of Resume Selection

by

Sumit Maheshwari, Abhishek Sainani, P Krishna Reddy

in

6th International Workshop on Databases in Networked Information Systems
(*DNIS2010*)

Report No: IIIT/TR/2010/44

Centre for Data Engineering
International Institute of Information Technology
Hyderabad - 500 032, INDIA
March 2010

An Approach to Extract Special Skills to Improve the Performance of Resume Selection

Sumit Maheshwari, Abhishek Sainani, and P. Krishna Reddy

Center for Data Engineering,
International Institute of Information
Technology, Hyderabad, (IIIT-H),
Hyderabad, Andhra Pradesh, India - 500032.
pkreddy@iiit.ac.in

Abstract. In the Internet era, the enterprises and companies receive thousands of resumes from the job seekers. Currently available filtering techniques and search services help the recruiters to filter thousands of resumes to few hundred potential ones. Since these filtered resumes are similar to each other, it is difficult to identify the potential resumes by examining each resume. We are investigating the issues related to the development of approaches to improve the performance of resume selection process. We have extended the notion of special features and proposed an approach to identify resumes with special skill information. In the literature, the notion of special features have been applied to improve the process of product selection in E-commerce environment. However, extending the notion of special features for the development of approach to process resumes is a complex task as resumes contain unformatted text or semi-formatted text. In this paper, we have proposed an approach by considering only skills related information of the resumes. The experimental results on the real world data-set of resumes show that the proposed approach has the potential to improve the process of resume selection.

Keywords: Special features, Resume selection.

1 Introduction

In the Internet era, large number of resumes are received on-line, through e-mails or through services provided by companies like Info Edge Limited [3]. For companies, it is a difficult and time consuming process to select the appropriate resume from such a large number of resumes. Research efforts [6][11] are going on to develop the methods for improving the performance of resume selection process.

Normally, resumes share document-level hierarchical contextual structure where the related information units usually occur in the same textual block and text blocks of different information categories usually occur in relatively fixed order [11]. One can observe the hierarchical structure in the resumes. The first layer consists of different sections such as education, experience, skills etc. and the second layer consists of text about corresponding sections.

S. Kikuchi, S. Sachdeva, and S. Bhalla (Eds.): DNIS 2010, LNCS 5999, pp. 256–273, 2010.

Table 1. Sample resume with corresponding sections and their respective features

Education
1. b.tech. (computer science & engineering) iiit, hyderabad (expected may, 2009) 6.66/10 cgpa. 2. senior secondary instrumental school, kota (cbse board 2004) 72%. 3. secondary st. sr. sec. school, ajmer (cbse board 2002) 83%.
Skills
1. programming languages: c, c++ 2. operating systems: windows 98/2000/xp, gnu/linux 4. scripting languages: shell, python 5. web technologies: html, cgi, php 6. other tools: microsoft office, latex, gnu/gcc, visual studio 2005/08 7. database technologies: mysql
Experience
1. audio-video conferencing over ip networks: 2. duration: nov. 2007 nov. 2008 team size: 2. technical environment: c++ abstract: the objective of this project was to develop an audio/video conferencing system which enables multiple users to communicate with each other via a global server with improved efficiency in terms of voice clarity and low latency. the system is equipped with resources to facilitate text chat, voice chat and voice/video chat between multiple clients. this client server application was developed using c++ and .net framework in windows environment. 3. windows firewall 4. duration: july-nov 2007 team size: 1 technical environment: c abstract: packets from or to a network are analyzed and according to the users settings actions are taken on how the packets would be handled. various options are provided to the user in accordance to which action is taken ranging from what the packet contains to the source of the packet. 5. document request form automation 6. duration: sep-nov 2006 team size: 2 technical environment: php, mysql abstract: project developed for iiit hyderabad administration. this web-based tool automates the processing of the various documents needed by students. 7. implementation of outer loop join 8. duration: jan-march 2007 team size: 1 abstract: implementation of the above operation as a part of the database management systems course. 9. myshell 10. duration: aug-oct 2006 team size: 1 abstract: developed a program which acted as a shell, starting and running command line arguments as part of our operating systems course. 11. other studies and presentations 12. analysis of animation video viewing 13. using an eye tracker to track which point on the screen were the viewers focusing on, i analyzed various trends in animation video viewing. this study was done as a part of cognitive course. 14. case study in software design 15. i was a part of a six member group involved in the thorough analysis of a software design problem and the task of coming up with a solution pertaining to the problem. this was done as a part of software engineering course.
Achievements
1. secured 1573 air in all india engineering entrance examination, 2005. 2. secured 2216 air in iit-jee screening examination, 2005 3. cleared national talent search examination level 1 in 2002. 4. was among the finalists of the rajasthan state science talent search

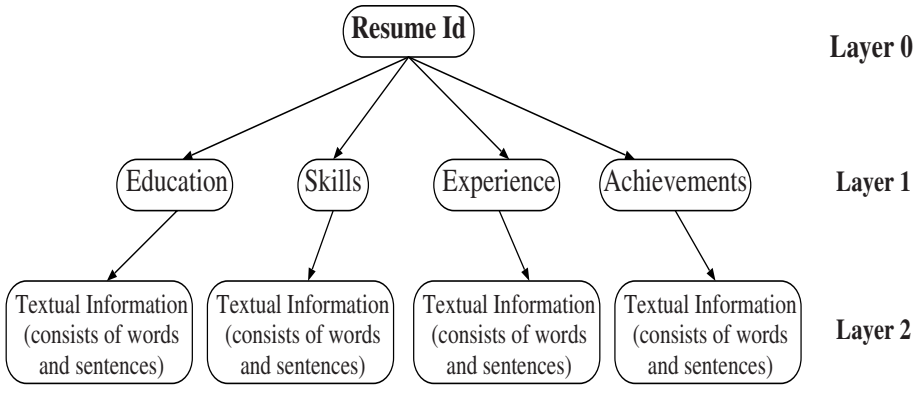

Fig. 1. Hierarchical structure of resume

Table 1 shows the sample resume. Each resume contains different sections and each section contains words and sentences as features. The numbering in each section denotes a feature separated by a delimiter ('newline' in our case). Figure 1 shows the corresponding hierarchical structure for Table 1. The top layer which is termed as "Layer 0" contains resume id. It can be observed that sections like education, experience, skills and achievements form the first layer of the resume. Each section is described by the text containing words and sentences which forms the second layer of the resume. Based on the structure of the content, the text of each section in the second layer can be organized into several layers.

The large number of resumes are reduced to few hundred potential ones based on some filtering techniques or search services [1] [2] [3]. The set of resumes hence obtained are similar to each other as they satisfy the search criteria or requirements for a company. In the current scenario, it is necessary to manually analyze each resume to select appropriate resumes. We define this problem as 'Problem of resume selection from set of similar resumes'. The research issue here is developing an improved approach which could help in selecting appropriate resume by processing similar resumes.

In this paper, we have made an effort to propose an improved approach based on the notion of special information. We consider that there may exist special information in some resumes as compared to others. For example, a resume may contain specialty in education, specialty in experience, specialty in skills or specialty in achievements and so on. Special information may exist in one or more sections of a resume. We assume that identifying such special information and organizing them efficiently enhance the performance of the resume selection process.

In the literature, an approach has been proposed to identify special features to improve the process of product selection in E-commerce environment [9]. They have exploited the fact that every product possesses some specialness, which is exhibited through few special features. Their approach identifies the special features and organizes the features of the product in an effective manner. In this paper we have extended the notion of special features to improve the process of resume selection. However,

resume selection process is a more complex problem than product selection process because:

(i) It was observed that the features are standardized in the product selection scenario whereas in case of resumes, features are free-form text which is difficult to process.
(ii) Resume has a hierarchical structure as compared to the product feature descriptions having single layered structure. Each resume contain several sections, and each section contains different types of text. For example, experience section contains long sentences, skills section contains skill type (programming languages) and skill values (c++, java).

So, development of an approach to process the resume dataset is a complex task, as separate approach has to be developed for identifying special information in each type of text and integrating the same appropriately. In this paper, we have proposed an approach to improve the performance of resume selection by considering only the skill related information of the resumes. The development of approach to extract information from other sections of the resume is a part of the future work.

We have conducted experiments on the real world data-set of resumes and the results show that the proposed approach has the potential to improve the process of resume selection.

The rest of paper is organized as follows. In Section 2, we discuss the related work. In Section 3, we explain the notion of "special features" used in the product selection framework [9]. In Section 4, we explain the basic idea of the proposed approach and discuss the algorithm. In Section 5, Experimental results are presented. Discussion is presented in Section 6. Summary and Conclusion are provided in Section 7.

2 Related Work

In [6], a toolkit named as "Learning Pinocchio $(LP)^2$", was applied on resumes to learn Information Extraction rules from resumes written in English. The information identified in their task includes a flat structure of Name, Street, City, Province, Email, Telephone, Fax and Zip code. Learning Pinocchio is an adaptive system for IE, based on a kind of transformation based like rule learning. Rules are learned by generalizing over a set of examples marked via XML tags in a training corpus.

An approach has been proposed in [11] for resume information extraction to support automatic resume management and routing. A cascaded information extraction (IE) framework is designed. In the first pass, a resume is segmented into consecutive blocks attached with labels indicating the information types. Then, in the second pass, the detailed information, such as Name and Address, are identified in certain blocks (e.g. blocks labelled with Personal Information), instead of searching in the entire resume. Based on the requirements of an ongoing recruitment management system which incorporates database construction with IE technologies and resume recommendation (routing), general information fields like Personal Information, Education etc. are defined.

In [9], the problem of 'selecting a product from a group of similar products in E-commerce environment' faced by the customer is investigated. It proposes an improved

approach to help a customer select the appropriate product by exploiting the fact that every product possesses some specialness, which is exhibited through few special features. The approach identifies the special features for each product and organizes the features of the products in an effective manner.

In Web mining research area, efforts are being made to identify outliers in a given set of web documents. Web content outliers are documents with varying content compared to similar Web documents taken from the same domain [4]. These approaches concentrate more on classifying objects as outliers rather than mining specialness of each object.

In [8], an effort has been made to discover unexpected information from the competitor's site. It compares the user's Web pages with those of the competitors and finds several kinds of unexpected information.

In data mining research, there have been efforts to mine outliers in numerical dataset. In [5], an approach has been proposed in which each object is assigned a degree of being an outlier, which is called local outlier factor. It is local as the degree depends on how isolated the object is with respect to the surrounding neighborhood. In [7], an approach has been proposed to mine the distance based outliers. The notion of K-nearest neighbor has been used to identify outliers in [10].

In this paper we have made an effort to extend the notion of special features to improve the process of resume selection. It can be observed that the work done on resumes mostly focuses on information extraction of resume or building a classifier to extract the information from resume and storing it in structured manner. Discovering unexpected information approach does not make an effort to exploit the special properties of each object within the whole set of pages including competitors and users set of pages. The outlier algorithms discussed in data mining area deals with numerical data-sets and have not applied in case of Web documents. The outlier algorithms proposed in Web mining area concentrates more on classifying objects as an outlier, rather than mining specialness of each object. None of the above approach tries to extract special information from the given set of resumes.

3 Identification and Organization of Special Feature Knowledge

In this section, we explain the basic framework related to the notion of special features as discussed in [9]. The framework consists of two main approaches: extraction of special features and organization of special features.

3.1 Extraction of Special Features

Before explaining the notion of special features we define the term 'degree of specialness'.

Degree of specialness: Let O be the set of 'n' similar objects, where object 'o_i' $\in$ O and each object 'o_i' possesses set of features 'f(o_i)'. Let 'F' be set all features such that F = $\cup_{i=1}^{n} f(o_i)$. Each feature in F is denoted by f_j where $0 \leq \mathrm{j} \leq |F|$ and n(f_j) denote the number of objects to which feature f_j belongs to. Note that, the set F is a

multi-set where each feature is represented as a tuple of $< resumeid, feature >$ and two different tuple may contain the same feature more than once since one feature can belong to multiple objects.

Let f_j be a feature, such that $f_j \in$ f(o_i). The degree of specialness (DS) of a feature f_j is its capability of making the object o_i separate/distinct/unique/special from other objects. The DS value for a feature varies between zero to one (both inclusive). The DS value of the feature f_j is denoted by DS(f_j). Then,

$$DS(f_j) = \begin{cases} 1 & \text{if n}(f_j) = 1 \\ 1 - (n(f_j)/|O|) & \text{otherwise} \end{cases} \quad (1)$$

About special features: Based on the DS values of features, features can be classified as common features, common cluster features and special features. The features that are occurring in all the objects have their DS value as '0' and are called common features. The features that are occurring in very few objects have their DS value close to 1 and are called **special features**. The other features are called common cluster features.

3.2 Organization of Features

After assigning the DS values to all the features in the data-set, the next issue is to organize the features in an effective manner by taking into account the corresponding DS values. On the basis of DS values, features can be categorized into one of the three categories: common features, common cluster features and special features.

Three-Level Feature Organization Approach: The features are distributed into three levels: I-level, II-level and the III-level. The I-level contains the common features, II-level contains common cluster features and III-level contains special features. It can be noted that, for any object o_i, its complete set of features f(o_i) is a combination of (i) the common features at I-level (ii) common cluster features for the cluster in which o_i is a member and (iii) special features of object o_i at III-level.

Figure 2 depicts the organization of four similar objects o_1, o_2, o_3 and o_4, where each object has some set of features. The I-level at the top shows the common features present in all the objects. The II-level shows the common features in each cluster. o_1 and o_2 form one cluster and similarly o_3 and o_4 form another cluster. The III-level shows the special features for each object. The complete set of features for object o_1 is combination of 'common features of all the objects' present in I-level, 'common cluster feature of the cluster containing o_1 and o_2' in II-level and 'special features of o_1' present in III-level.

The procedure to organize the features using three-level approach is given in [9]. Here, we provide a summary for the three-level approach. It is a clustering algorithm that takes the set of objects O, similarity threshold (ST) and feature set F as input and returns common features, common cluster features and special features for each object with formation of clusters as an intermediate step. The similarity measure used in clustering algorithm is described below.

Common Features of all the objects **(I-level)**		
(II-level) Common Cluster Features	Object Identifier	**(III-level)** Special Features
Common features of O1 and O2	O1	special features of O1
	O2	special features of O2
Common features of O3 and O4	O3	special features of O3
	O4	special features of O4

Fig. 2. Three level representation of resume skill features

The similarity between the objects o_i and CL(i) (where CL(i) denotes the i'th cluster) is denoted by sim(o_i, CL(i)) and is calculated as follows:

$$sim(o_i, CL(i)) = |f(o_i) \bigcap CF(i)|$$

where CF(i) represent the features that are common among all the objects present in the cluster CL(i) and $f(o_i)$ denotes the features present in object o_i.

The approach contains two parts. The summary of the first part is as follows. The first cluster is initialized with some object. Next, the following step is repeated for each object: For each other object o_j, if the similarity of o_j with the existing cluster or clusters is greater than similarity threshold, the object o_j is put into into the cluster with maximum similarity; Otherwise, new cluster is initialized with o_j.

In the second part, the features of each cluster are organized into three-levels. The I-level contains the features of all clusters with DS value as '0'. The II-level contains the common features of each cluster. The III-level contains the remaining special features of each object.

About setting similarity threshold (ST) value: Organizing the features using three-level method is an iterative process. The value of ST should be chosen such that the objects are clustered into a reasonable number of clusters and the number of features shown to user can be reduced. The objective of clustering the objects is to reduce the effort of users by providing them with more convenient view and also less number of features. In case of large number of clusters, it leads to more confusion. Finally, we can set the ST to particular value which gives minimum number of clusters, and minimum number of features to be shown to user. For example, ST could be chosen as fifty percent of the average number of features in an object eliminating the common features. The threshold can be gradually increased, and the number of clusters formed and total number of features shown to user can be observed. If the number of features to be shown decreases significantly, we can increase the threshold and check the same. It can be observed that if the threshold is decreased, the number of common features for each cluster would decrease and consequently number of clusters shown to user would be increased.

4 Proposed Approach

In this section, after explaining the problem definition, we will discuss the basic idea and the proposed approach. We also discuss the overall framework and options for using the proposed framework.

4.1 Problem Definition

It is a difficult and time consuming process to select appropriate resume from a large set of resumes. Currently available filtering techniques or search services [1] [2] [3] filter thousands of resumes to few hundred potential ones. Since these filtered resumes are similar to each other, examining of each resume becomes essential to know about the potential candidates. The problem is defined as follows: Given a set of similar resumes, develop a methodology to help the enterprises/recruiters to improve the performance of resume processing.

We define 'similar resumes' as a set of resumes that are produced as result after filtering through the enterprise's resume management systems. Similar resumes consist of resumes with same experience or applying for the same job. The input to the proposed approach is a group of similar resumes and the output is an organization of resumes based on their specialness.

4.2 Basic Idea

The problem here is to develop an approach to select the appropriate resumes efficiently. It can be observed that there are some common features which are present in all the resumes in the group and also each resume may possess some special features that could differentiate it from rest of the resumes in the group. The intuition here is that if the special features of each resume are identified, the time required to make a decision for selecting an appropriate resume would be reduced in comparison to the time required by considering all the information.

Normally, each resume is described by a text document and the text in the resume is divided into different sections. A special information of a resume implies special information in each section of a resume. For example, there could be specialty in skills, specialty in achievements, specialty in education etc. The problem here is to identify the special information from each resume. Each section contains different types of text. For example experience section contains long sentences, skills section contains skill type (programming languages) and skill values (c++, java), the development of an approach to process the resume dataset is a complex task, as separate approach have to be developed by identifying special information for each type of text and integrate the same appropriately.

In this paper, we explored only the skill section and develop an approach to identify resumes with 'special skills'. The development of approach to identify special information in other sections of the resume is beyond the scope of this paper and would be considered as a part of the future work.

The main issue here is how to measure the specialness of text in skills section for all the resumes. We extend the notion of special features to propose an effective approach for the resume selection problem.

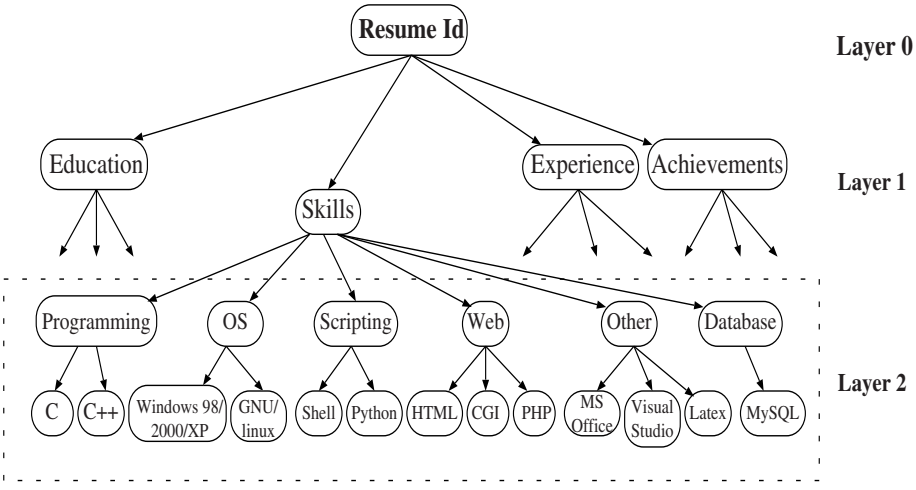

Fig. 3. Hierarchical structure of resume corresponding to Table 1

Table 2. Sample features for skills section

Skill Type features	Skill Values features
1. programming languages	c, c++,java.
2. programming languages	c, c++ java.
3. scripting languages	python, perl, shell
4. scripting languages	python perl shell
5. libraries	opengl, sdl
6. database technologies	mysql mssql
7. operating systems	linux, windows

4.3 Description of Proposed Approach

If we apply the notion of special features on skills section directly the comparison between the features would not be effective. It can be observed that the skill section information (refer Table 1) contains enumerated sequence of text pieces. Each text piece consists of a skill type and its skill values. For example, 'programming languages' is a skill type and 'c, c++, java' are skill values. So, there is a two layer organization in the skill information as shown in Figure 3. The skill information itself forms an hierarchy where skill types form one layer and skill values form another layer. We exploit the inherent organization in the skills information.

We propose an approach by considering that the skill information in the resume is organized into "skill type" and their corresponding "skill values".

Overall the proposed approach consists of a number of steps. First we perform preprocessing on the skill information of the resumes. Then we extract the skill type and skill value features. After extracting the skill type and skill value features, we calculate DS values of the features and organize them.

There are two types of features that could be extracted from skills information. One type is Skill-Type-Feature-Set (STFS) and another is Skill-Value-Feature-Set (SVFSs). Each element in STFS is a two attribute tuple $< ResumeId, Skilltype >$ and each element in SVFS is defined as $< ResumeId, Skilltype, Skillvalue >$. Note that, for a given resume, there exists one STFS and several SVFSs. For the same skill type, same skill values exist, but in different form. For example, in Table 2, the skill values for skill type 'programming languages' are same, except the presence of some special characters (comma in this case). Thus, direct comparison cannot be done. So, both STFS and SVFSs are formed after carrying out the preprocessing steps and then applying the described algorithm (refer Table 3) on the skills information.

Extracting skill types and skill values: The algorithm to extract skill types and skill values is divided into two parts. In the first step, we apply the preprocessing and in the second step we apply an algorithm described in Table 3 to identify the skill type and skill value features. The preprocessing steps are as follows:

(i) Entire input text is converted to lower case and special characters are removed.
(ii) Stop words [1] occurring in general purpose stop words list are removed.
(iii) The skills section in resume is identified with the keyword 'Skills' in the heading irrespective of the position of the Skills section in the resume.
(iv) The skill type and its skill value(s) are identified and separately stored using a delimiter (: in our case).
(v) Skill value(s) corresponding to each skill type are sorted lexically and separated by a comma (,). For a skill value having more than one word, the words are concatenated. For example, the skill values for skill feature 'database technologies: ms sql, postgres sql, mysql' would be changed to skill value string: 'mssql, mysql, postgressql'.
(vi) To resolve human errors like spelling mistakes, typo errors etc., we define a data structure called 'skill values list' with 'skill type' as a hash key and its possible 'skill values' as its values. Each skill value is checked in the skill values list. In case of many partial matches, the skill value is replaced by the skill value from the list with which it has the longest match. In case of no match, the list is manually updated with the skill value after verification. The possible skill values are extracted from the resume dataset.
(vii) A skill value can have more than one name referring to it. For example mssql and microsoft sql refers to same skill. To resolve such ambiguity we identify the various possible ways of redundant occurrences through data analysis and prepare a hash table with the canonical names as the hash key and various possible names as a list of hash values corresponding to the canonical name. All these different names should be replaced by a common name or canonical name to resolve this issue.

The description of algorithm shown in Table 3 is as follows. The input to the second part of the algorithm is set R consisting of 'n' resumes, dictionary S that contains all

[1] Stop words is the name given to words which are filtered out prior to, or after, processing of natural language data (text).

the distinct skill types present in the set R and $|S|$ denotes the cardinality of dictionary S. The output consists of the skill type feature set (STFS) and skill value feature set (SVFS). In STFS each element is a tuple consisting of resume identifier and skill type as its attributes whereas in SVFS each element is a tuple consisting of resume identifier, skill type and skill value. The steps of the algorithm are as follows: We take each resume and repeat the following steps for each resume. (i) Identify the skill section of the resume using the 'Skills' tag. (ii) Process each line of the skill section to identify the skill type and corresponding skill value. (iii) The resume id (r_i) and skill type is stored in $STFS_i$ index of the array of SVFS where as resume id (r_i), skill type (s_j) and skill value is stored are the index $SVFS_{ij}$ of the array SVFS.

Thus after performing the preprocessing steps and applying the above described algorithm we get STFS and SVFSs. The next task is to calculate the specialness values of all the features in STFS and SVFSs and organize the same.

Table 3. Algorithm to calculate STFS and SVFS

Input: R: Set of 'n' resumes; F: set of features for all 'n' resumes; S: dictionary for all the skill types and $\|S\|$ is number of distinct skill types **Output:** STFS and SVFS
1. Notations used: i, j: integers; S_{r_i}: skills information for resume r_i $STFS_i$: array for skill types for resume r_i, where each tuple contain $< r_i, Skilltype >$ $SVFS_{ij}$: array of skill values for resume r_i and skill type s_j, where each tuple contain $< r_i, s_j, Skillvalue >$ 2. **for** i=1 to n 3. Get the skill section features for resume r_i in S_{r_i} 4. **for** each s_j in S 5. **if** $s_j \in S_{r_i}$ 6. store the tuple $< r_i, s_j >$ in $STFS_i$ 7. store the tuple $< r_i, s_j, skillvalue >$ in $SVFS_{ij}$ 8. end 9. end

Calculating DS Value and Organizing the Special Skill Types: Given the STFS, the problem is to identify the specialness value and then on the basis of specialness value organize all the features in the set.

Computing Specialness Value for STFS: Let R be a set of 'n' similar resumes, where resume $r_i \in$ R. Each resume r_i possess some set of features. Let f(r_i) be set of skill type features for resume r_i. Let F be set of all skill type features for all resumes such that F = $\cup_{i=1}^{n} f(r_i)$. Each feature F is denoted by f_j where $0 \leq j \leq |F|$ and n(f_j) denote the number of resumes to which feature f_j belongs. The DS value for each feature in STFS is calculated as defined in Equation 1. The input consists of the feature set F and output consists of Feature set F along with the DS values for all the features in the set.

Organization of STFS: We apply the above described three-level organization algorithm on the features in STFS and organize the features as shown in Figure 2. The input to the algorithm consists of feature set F which contains all the features in STFS along with their DS values, threshold ST and set of resumes R and the output of the algorithm consists of three-level organization of STFS features.

Calculating DS Value and Organizing Special Skill Values: Given the skill types and SVFS, the problem is to identify the specialness value and then on the basis of specialness value organize all the features.

Computing Specialness Value for SVFSs: Let S be a set containing distinct skill type features from all the resumes and $s_j \in$ S denotes a particular skill type. Let f(s_{ij}) denotes the skill value features for skill type $s_j \in$ S and resume $r_i \in$ R. Let F(s_j) be set of all skill value features for skill type s_j for all the resumes such that F(s_j) = $\cup_{i=1}^{n} f(s_{ij})$. The DS value of each feature in SVFSs is calculated using Equation 1. The input consists of the feature set F(s_j) for all $s_j \in$ S and output consists of Feature sets along with the DS values for all the features for each of skill type.

Organization of SVFSs Features: We apply the above described three-level organization algorithm on the features in SVFSs and organize the features as shown in Figure 2. The three-level organization algorithm is run for each distinct skill type $s_j \in$ S. The input to the algorithm consists of feature set F(s_j) that consists of features for skill type s_j along with their DS values, threshold ST and set of resumes R and the output of the algorithm consists of three-level organization of skill value features for each skill type s_j.

4.4 Overall Framework

In this section we explain the overall framework. The input to the proposed approach are the resumes stored as text documents where each document contains different sections along with their descriptions (refer Table 1). The steps of proposed framework are discussed below (refer Figure 4).

1. **Identification of features from skills section:** We extract Skill Type Feature Set and Skill value Feature set from skills information for all the resumes.
   - **Identifying Skill Type Feature Set (STFS):** We identify the skill type features for all the resumes and form an STFS.
   - **Identifying Skill Value Feature Set (SVFSs):** We identify the skill value features for each of the skill type and form SVFSs for all the skill types.
2. **Calculating DS Value and Organizing Special Skill Type Features:** We compute the DS value for skill type features on the basis of DS values we organize the skill type features.
   - **Computing DS Value for Skill Type Features:** We compute the DS value for skill type features based on the notion of degree of specialness as defined in Equation 1.
   - **Organization of Skill Type Features:** We organize the skill type features using the three-level organization approach described in Section 3.2.

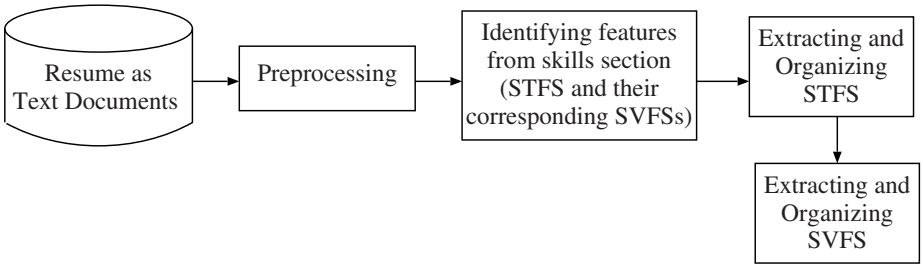

Fig. 4. Flow diagram of the overall framework

3. **Calculating DS Value and Organizing Special Skill Value Features:** We compute the DS value for skill value features for each of skill type and on the basis of DS values we organize the skill value features for each of the skill type.
   - **Computing DS Value for Skill Value Features:** We compute the DS value for skill value features for each of skill type based on the notion of degree of specialness defined in Equation 1.
   - **Organization of Skill Value Features:** We organize the skill value features for each of skill type using the three-level organization approach described in Section 3.2.

4.5 About Using the Proposed Framework

The user can request the system to organize the resumes according to skill type or skill values for the selected skill type. The resumes are organized firstly on the basis of skill type forming the first layer and second layer consists of tables for skill value of each skill type. Now, if the recruiter wants to select a resume only on the basis of skill type, he/she can give only the skill type information as input and browse through the output containing the special skill type information only. And if he/she wants to select a resume based on the skill value for a respective skill type he/she can do so by giving the skill type and skill value information as input and browsing through the special skill types and then special skill values for a respective skill type.

5 Experimental Results

To evaluate the performance, we have applied the proposed framework on real world data-set of resumes. Data-set contains 100 resumes from undergraduate students of computer science department in a University. All the resumes are available in the same format as shown in Table 1. Total number of features in skills section were 643. The skill types present in the data-set are ‘programming languages’, ‘scripting languages’, ‘operating systems’, ‘web technologies’, ‘database technologies’, ‘libraries’, ‘other tools’, ‘compiler tools’, ‘mobile platforms’ and ‘middleware technologies’.

We define the performance metric called ‘reduction factor’ (rf) to measure the performance improvement. The rf denotes the reduction in the number of features that the recruiter needs to browse to select a resume from set of ‘n’ similar resumes.

Table 4. Organization of skill type features

Common Features (I-level)		
programming languages, operating systems, web technologies, database technologies		
Common Cluster Features (II-level)	**Resume Identifier**	**Special Features (III-level)**
other tools, scripting languages	78	libraries, mobile platforms middleware technologies
	83, 13, 91, 114, 112, 52, 67, 54, 15, 108, 105, 109, 107	compilers, mobile platforms
	25, 5	libraries, mobile platforms
	36, 77, 15	Compiler tools
	43, 44, 70, 71, 76, 69, 40, 39, 50, 57, 62, 59, 101, 95, 65, 68, 84, 90, 110 88, 19, 113, 6, 3, 32, 92, 29, 104, 93	libraries
	9, 98, 86, 73, 79, 8, 66, 28, 115, 103 106, 11, 37, 24, 53, 55, 47, 1, 41	None
libraries, other tools	63, 35	compiler tools
	89, 94, 17, 49	None
scripting languages, libraries	75, 96, 45, 2, 58, 33, 14, 10, 51	None
other tools	82, 27, 99, 38	None
scripting languages	111, 4, 22, 23, 21, 46, 42, 16	None

Table 5. Reduction Factor value for STFS

Feature Type	$\|F\|$	$\sum_{i=1}^{L} F(i)$	**rf**
Skill type	643	79	0.88

Let ‘F’ denote the total number of features for all the resumes, F(i) denote the number of features in ‘i’-level and ‘L’ denotes the number of levels. The ‘rf’ is defined as,

$$rf = 1 - \frac{\sum_{i=1}^{L} F(i)}{F}$$

Results for Skill Type Features (STFS): The Table 4 shows the reduction factor for skill type features. It can be seen that rf value comes to 88%. The results indicate that 88% reduction in the effort could be achieved in resume selection process. The total numbers of skill types features present were 643 and numbers of features being displayed to user are only 79. The Table 5 shows the organization of corresponding skill type features (set F) using three-level approach. The I-level shows the common skill type, the II-level shows the common skill types for each cluster of resumes and the III-level shows the special skill types for each resume. The resumes can be classified based on their skill type in one click. Since number of resumes share same special feature we have mentioned them in same row separated by delimiter comma (,) for user convenience as well as to reduce space. There is such large reduction in the number of features displayed as large number of features are present as common features so

instead of displaying them for each resume, it's been displayed only once. Similarly the cluster features are displayed once for all the resumes present in a cluster instead of separately displaying for each one.

Results for Skill Value Features (SVFSs): The reduction factor in the skill value features for each of the skill type is shown in Table 6. It can be observed that rf values for SVFSs for some skill types is very high, for few skill types low and in some cases medium. The reason for high reduction factor for some skill types is that there are number of resumes that share common skill values for these skill types and thus the clusters formed are uniform. The reason for low reduction factor for skill type such as middleware technologies or mobile platforms is because the number of features in these sets

Table 6. Reduction Factor values for SVFSs

Feature Type	$\|F\|$	$\sum_{i=1}^{L} F(i)$	**rf**
database technologies	148	10	0.93
programming languages	283	38	0.86
scripting languages	150	66	0.56
compiler tools	41	7	0.83
mobile platforms	4	3	0.25
middleware technologies	3	3	0
libraries	170	91	0.56
web technologies	354	154	0.34
operating systems	271	16	0.94
other tools	302	208	0.31

Table 7. Organization of skill values for skill type "database technologies"

Common Features (I-level)		
None		
Common Cluster Features (II-level)	**Resume Identifier**	**Special Features (III-level)**
mssql, mysql	10, 101, 105, 107, 114, 115, 14, 16, 19, 24, 25, 29, 3, 32, 33, 35, 36, 39, 4, 40, 41, 42, 46, 47, 5, 50, 53, 54, 55, 57, 58, 59, 6, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 73, 75, 76, 77, 78, 84, 86, 88, 90, 92, 93, 98	none
	103, 112	postgresql
	51	oracle
mysql	1, 108, 109, 11, 110, 111, 15, 17, 2, 21, 22, 23, 26, 27, 28, 37, 43, 44, 45, 79, 8, 87, 89, 9, 91, 95, 96	none
	104	msaccess
	113	jdbc, postgresql
mssql	38, 49, 94	none

Table 8. Organization of skill values for skill type "compiler tools"

Common Features (I-level)		
None		
Common Cluster Features (II-level)	**Resume Identifier**	**Special Features (III-level)**
lex, yacc	105, 107, 108, 109, 112, 114, 13, 15, 36, 52, 54, 64, 67, 77, 83	none
	26, 91	phoenix
	35	phoenix, rdk
	63	phoenix

Table 9. Organization of skill values for skill type "programming languages"

Common Features (I-level)		
c, c++		
Common Cluster Features (II-level)	**ResumeId**	**Special Features (III-level)**
javascript, msil	101, 25	none
	112	j2ee
javascript, python	17	latex
	19	.net, vc++
	35	batchscripting, shellscripting
matlab, python	38	perl, php
	62	none
javascript, vb	47	none
	87	symbian
python	1, 27, 68, 82, 89	none
	63	shellscripting
javascript	10, 103, 104, 111, 113, 21, 24, 3, 33, 37, 39, 43, 49, 5, 51, 53, 6, 75, 78, 94, 98	none
	11	perl
	54	j2me, socketprogramming
	99	lisp, vhdl
Nasm	106, 108	none
	109	msil, oz
matlab	14, 4, 90	none
Mips	69, 76, 86, 92	none
	36	socketprogramming
	55	actionscript, mxml
	57	openc++, symbian
	77	prolog

are very less. Thus their is very little scope of clustering the resumes based on common features. Though in most of the cases reduction factor is above 50%. Thus we can say that on average there is 50% reduction in the efforts of HR managers in resume selection process.

For each skill type, its respective skill value features are organized using three-level feature organization. Table 7 shows the organization of skill value for skill type 'database technologies'. The I-level shows the common skill value in 'database technologies', the II-level shows the common skill value in 'database technologies' for each cluster of resumes and the III-level shows special skill value in 'database technologies' for each resume. Table 8 and 9 shows the organization of skill value for skill type 'compiler tools' and 'programming languages' respectively. Similarly skill value features for other skill type like 'scripting languages', 'mobile platform', 'middleware technologies', 'libraries', 'operating systems', 'web technologies' and 'other tools' can be organized in the similar manner.

6 Discussion

There has been little research work related to the issue of resume extraction and selection. Most of the work has been focused on information extraction from resumes. Accordingly, various approaches have been proposed to extract structured information from a given set of resumes. Also most of the companies use a resume management system that helps them to get the selected resumes based on the user query. The resume management systems give hundreds of results for which again user has to manually browse each of the resumes. Secondly the system is dependent on user query.

In this paper we have made an effort to develop an approach to reduce the task of manually browsing each resume by discovering the special features and organizing them in an efficient manner. Also the system is independent of user query and helps the user to discover important information that user might be unaware of.

The evaluation of the proposed approach mentioned in the paper in the form of reduction factor indicates the benefit in the information reduction by comparing the text. But the real evaluation of the proposed approach is yet to be done by observing how it can help the enterprises.

The problem of resume selection and extraction is a problem that has not received much attention so far. The approach proposed in this paper is just the beginning and provides a direction towards the problem of resume selection. More research work is required to address the problem of resume selection and extraction.

7 Summary and Conclusion

Selecting appropriate resume from a group of similar resumes is one of the problems faced by the recruiters in most of the companies. We have extended the notion of special features to extract the special skills from given set of similar resumes. We have proposed an approach to identify resumes by analyzing "skill" related information. The proposed approach has the potential to improve the performance of resume processing

by extracting both special skills type and special skill values. With the help of the experimental results we have shown that there is 50-94% reduction in the number of features that the recruiter needs to browse through to select appropriate resumes.

The resume selection process is a more complex problem as resume contains free-form texts which are difficult to compare and also has an hierarchical structure containing different sections. As each resume contains different sections with each section containing different types of text, an integrated approach has to be developed by considering information in each section. In this paper, we have developed an approach by considering only the skills related information of the resume. As a part of future work, we are planning to investigate approaches to extract special information from other sections of the resume and develop an integrated approach for resume processing.

Acknowledgements

This work has been carried out with the support from Nokia Global University Grant.

References

1. Times business solutions limited, July 24 (2000) http://www.timesjobs.com/
2. Flagship brand of monster worldwide, inc., July 27 (2009) http://www.monsterindia.com/
3. Info edge (india) ltd., July 30 (2008) http://www.naukri.com/
4. Agyemang, M., Barker, K., Alhajj, R.S.: Mining web content outliers using structure oriented weighting techniques and n-grams. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 482–487. Springer, Heidelberg (2006)
5. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. SIGMOD Rec. 29(2), 93–104 (2000)
6. Ciravegna, F., Lavelli, A.: Learningpinocchio: adaptive information extraction for real world applications. Nat. Lang. Eng. 10(2), 145–165 (2004)
7. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: VLDB 1998: Proceedings of the 24rd International Conference on Very Large Data Bases, pp. 392–403. Morgan Kaufmann Publishers Inc., San Francisco (1998)
8. Liu, B., Ma, Y., Yu, P.S.: Discovering unexpected information from your competitors' web sites. In: KDD 2001: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 144–153. ACM, New York (2001)
9. Maheshwari, S., Reddy, P.: Discovering special product features for improving the process of product selection in e-commerce environment. In: ICEC 2009: Proceedings of the 11th international conference on Electronic commerce, Taipei, Taiwan. ACM, New York (2009)
10. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. SIGMOD Rec. 29(2), 427–438 (2000)
11. Yu, K., Guan, G., Zhou, M.: Resume information extraction with cascaded hybrid model. In: ACL 2005: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 499–506. Association for Computational Linguistics (2005)