A New Feature Extraction Approach to Extract Area of Expertise from Resumes to Augment the Hiring Process

Dr. K. Subramanian,
M. Sc., M.Phil, Ph.D.,Assistant Professor,
Dept of Computer Science,
Government Arts College,
Kulithalai,
subjjcit@gmail.com

M. Latha,
Research Scholar,
J. J. College for Arts and Science (Autonomous),
Pudukkottai,
lathaaravind2011@gmail.com

Abstract:-Text Feature extraction is a process of detecting and discovering promising data from a large unordered textual data set. The main objective of the feature extraction process is to unearth the promising data and transmit them in to acceptable format to help in decision making. With the ever evolving digital technology number of resumes posted everyday seeking for a job increases steeply and this voluminous data intricate the recruitment firms to identify the right candidate for the right job. The main objective of this paper is to deals with a new feature extraction approach using ranking based frequent text occurrences to extract promising texts from the resume dataset and reduces the hiring agencies manual work considerably, reduces the dimensionality of the data to a larger extent and thereby reduces the running or execution time and memory footprints required largely when compared with the existing approaches.

Keywords:-*Feature extraction, Text mining, Resume selection, promising text extraction, frequent item sets.*

*****

1. INTRODUCTION

Automatic text feature extraction is an element alglitch in text data mining. Even a reasonably midsize document may involve several relatively independent sections, texts and parts. Such adverse heterogeneity in text documents can exceptionally affect the performance of text mining to a greater extent. .In this paper a new approach to extract important keyword texts from the resume dataset is proposed and analyzed. The structure of a resume generally comprises of two parts, *Section Head* and *Section data*. Both these parts are interrelated and appear in the same block or table. Mostly resume consists of multiple sections of two-layered architecture. Educational Details, Experience summary, skills and personal details, project handled are all examples of a resume section.

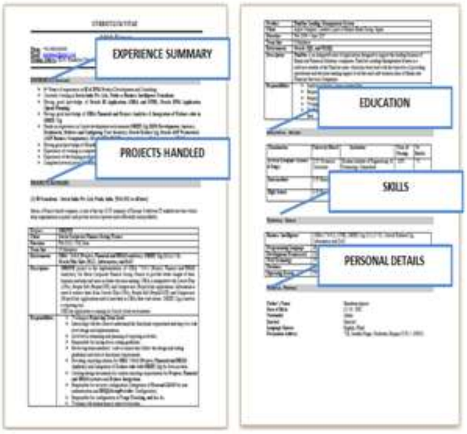


Figure 1.1: Sample resume

A resume is a brief document about an individual trying to market him/her to the industry. They usually are structured and hierarchical documents but contain unstructured data too [10]. In a resume, the format is not predetermined and it is based on the author's thinking, which makes the information extraction, comparison, and selection a daunting task. Each resume is unique in its own way as it contains words and sentences as features [8]. It can also be viewed as a multi section document, with description of each section, which highlights the different aspects of an individual's professional career [9].

RELATED WORK

Most of the existing algorithms heavily rely on one concept: the quantification of lexical cohesion between different parts of a document. Lexical cohesion can be defined as *"the cohesive effect achieved by the selection of vocabulary"* [1] which intuitively means that changes in the vocabulary accompany topic shifts. A number of linguistic structures create lexical cohesion, such as word repetitions, synonyms and pronouns. However, those structures can be difficult to detect due to language ambiguities, often leading to the quantification of lexical cohesion solely based on some measure of word repetitions.

There are many commercial products on resume data extraction and retrieval, but very few research works has been carried out in this area. Some of the commercial products include: Daxtra CVX [2],ResumeGrabber Suite [5], ALEX Resume parsing [4], Akken Staffing [6], andCV Parser [3]. The product specification, algorithms and methods used in them for resume information extraction are not available completely.

PROBLEM DEFINATION

Reputed firms receive several hundreds of resumes from job seekers every day. But generally there is no specific standard formats in which a resume can be drafted and written. To implement a uniform standard so that the resumes can be electronically classified and searched companies force job seekers to fill an online template. A major problem associated with this approach is that the applicant is forced to modify their resume to match the style of the template which might be a burden to the applicant to display all their credentials.

So most of the applicant sends their resumes in document format to the hiring companies to showcase their expertise but to mine and fetch the prospective candidates from this colossal volume of unordered text resume file is a tedious and a cumbersome task. Even though the resume selection from large resume dataset collection is a new vertical, many research works are carried out. The hiring managers/human resource (HR) business partners obtain the segregated set of résumés which are similar to each other. Next, a manual analysis of each résumé is required to mine for the best candidates. This is referred to as “Problem of Resume Selection” [7].

BASIC IDEA

The problem here is to develop an approach to select the prospective resumes efficiently which nearly matches with the hiring agent’s requirement using some important unique features present in the resumes. The problem is that most of the resumes will contain many common features along with some special unique features that could differentiate it from rest of the resumes in the collection.

The idea here is to identify and extract the unique features from the resumes, classify them according to the rank and enable the recruiter to make a decision for selecting the prospective candidates. This research paper aims to address a simple question “How to unearth prospective incumbents from a colossal volume of resumes?” This question motivates to propose a new approach based on frequent text to mine resumes and present the hiring agents with the best possible resume matches in-line with their requirements.

PROPOSED APPROACH

The proposed approach comprises of three phases namely, Preprocessing, Extraction and Ranking.

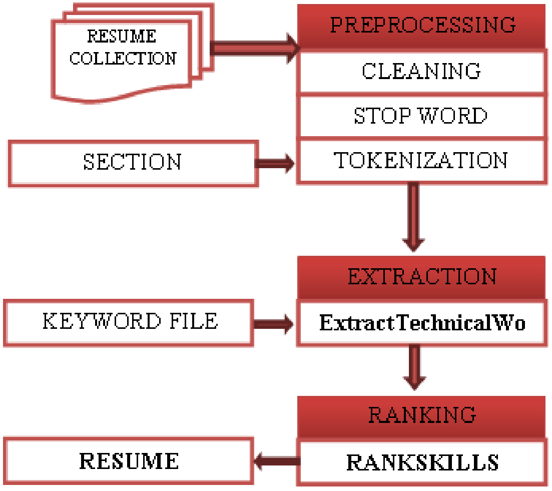


Figure 1.2: Proposed architecture

1. PREPROCESSING

In the preprocessing phase procedure named ProcessResume is proposed and this procedure cleans and tokenizes the text present in the resume based on the section reference file shown in figure 1.4 to categorize the tokens into the appropriate sections. The pseudo code of ProcessResume is shown in the figure 1.3.

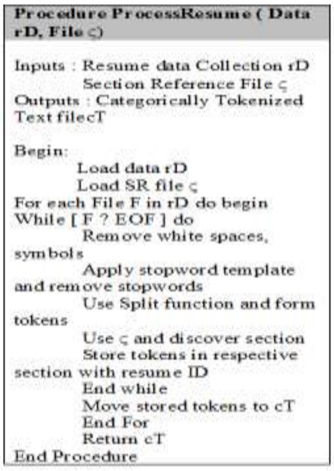
Cleaning

The preprocessing comprises of many procedures like cleaning the data (i.e.) removing the unwanted blank spaces/white spaces.

Stop words removalStop words are frequently occurring, insignificant words. This step filters out the common words used in most of the documents to facilitate phrase search.

Tokenization

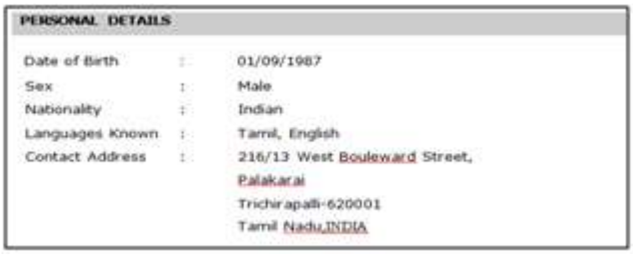
A document is treated as a string or set of words, and then partitioned into a list of tokens. This process breaks the stream of texts into words, phrases and symbols.



```
Procedure ProcessResume ( Data
rD, File ς)

Inputs : Resume data Collection rD
         Section Reference File ς
Outputs : Categorically Tokenized
Text filecT

Begin:
        Load data rD
        Load SR file ς
For each File F in rD do begin
While [ F ? EOF ] do
         Remove white spaces,
symbols
         Apply stopword template
and remove stopwords
         Use Split function and form
tokens
         Use ς and discover section
         Store tokens in respective
section with resume ID
         End while
         Move stored tokens to cT
         End For
         Return cT
End Procedure
```

Figure 1.3: Pseudo code of ProcessResume procedure

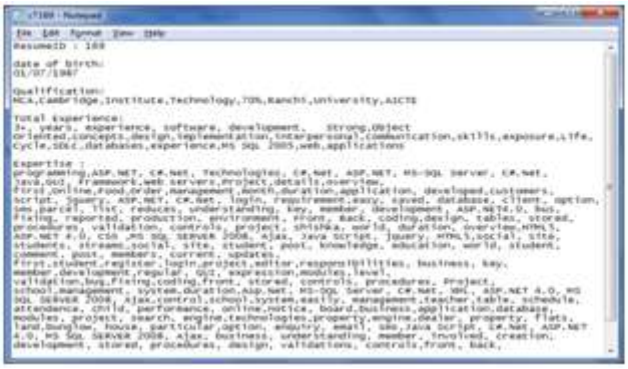
Total Experience	Summary, Experience Summary, Experience synopsis, Professional summary, Experience, Summary of Experience
Age	Age, dob, date of birth, DOB
Qualification	Qualification, Academic, Education, Educational qualification
Expertise	Expertise, System expertise, Projects, Academic chronicle, Work profile, projects handled, project profile, project details, technical profile, specialization, skill set


PERSONAL DETAILS

Date of Birth : 01/09/1987
Sex : Male
Nationality : Indian
Languages Known : Tamil, English
Contact Address : 216/13 West Boulevard Street,
Palakarai
Trichirapalli-620001
Tamil Nadu,INDIA

figure 1.4: Section reference file

The categorically tokenized text files are used to extract the important features present in the resumes. These file are very small in size as almost 80 to 90 % of the unpromising texts present are removed and the overall dimension of the data is reduced considerably. Sample categorically tokenized text file after preprocessing is shown in the Figure 1.5.



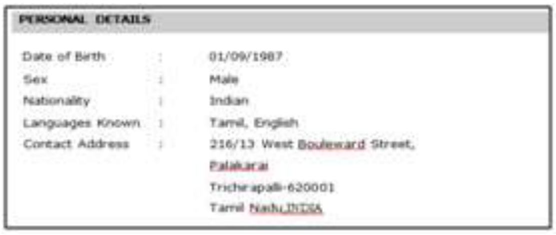
```
ResumeID : 169

date of birth:
01/07/1987

Qualification:
MCA,Cambridge,Institute,Technology,70%,Ranchi,University,AICTE

Total Experience:
3+, years, experience, software, development,   Strong,Object
oriented,concepts,design,implementation,interpersonal,communication,skills,exposure,life,
cycle,SDLC,databases,experience,MS SQL 2005,web,applications

Expertise :
programming,ASP.NET, C#.Net, Technologies, C#.Net, ASP.NET, MS-SQL Server, C#.Net,
Java,GUI, Framework,web servers,Project,details,overview,
first,Online,Food,Order,management,month,duration,application, developed,customers,
script, jquery, ASP.NET, C#.Net, login, requirement,easy, saved, database, client, option,
SMS,parcel, list, reduces, understanding, key, member, development, ASP.NET4.0, bug,
fixing, reported, production, environment, front, back, coding,design, tables, stored,
procedures, validation, controls, project, Shishka, world, duration, overview,HTML5,
ASP.NET 4.0, CSS ,MS SQL SERVER 2008, Ajax, Java Script, jquery, HTML5,social, site,
students, streams,social, site, student, post, knowledge, education, world, student,
comment, post, members, current, updates,
first,student,register,login,project,editor,responsibilities, business, key,
member,development,regular, GUI, expression,modules,level,
validation,bug,fixing,coding,front, stored, controls, procedures, Project,
school,management, system,duration,ASP.NET, MS-SQL Server, C#.Net, XML, ASP.NET 4.0, MS
SQL SERVER 2008, Ajax,control,school,system,easily, management,teacher,table, schedule,
attendance, child, performance, online,notice, board,business,application,database,
modules, project, search, engine,technologies,property,engine,dealer, property, flats,
land,bunglow, house, particular,option, enquiry, email, SMS,Java Script, C#.Net, ASP.NET
4.0, MS SQL SERVER 2008, Ajax, business, understanding, member, involved, creation,
development, stored, procedures, design, validations, controls,front, back,
```

Figure1.5: Categorically tokenized file memory size of the actual resume cT169 is 67 KB and when the preprocessing is carried out the dimension of the data is reduced greatly and the cT169 categorically tokenized file size decreases to2.02 KB, since except the promising texts all other texts are ignored. Right from the phase one, the dimension reduction is effectively handled and this will largely reduce the overheads like memory footprints and execution time.

2. EXTRACTION

A procedure namedExtractTechnicalWords is proposed and the promising texts present in the categorically tokenized text files are fetched. The promising objects or texts extracted from this proposed procedure are age, total experience, qualification and expertise. The age is explicitly provided in the resumes or it is extracted from the date of birth of the candidates using regular expression.

Regular Expression = "\Date of Birth\:\ [0-9]{6-8}[/]{2}" is TRUE if the text Date of Birth followed by 6 to 8 numbers and two "/".


PERSONAL DETAILS

Date of Birth : 01/09/1987
Sex : Male
Nationality : Indian
Languages Known : Tamil, English
Contact Address : 216/13 West Boulevard Street,
Palakarai
Trichirapalli-620001
Tamil Nadu,INDIA

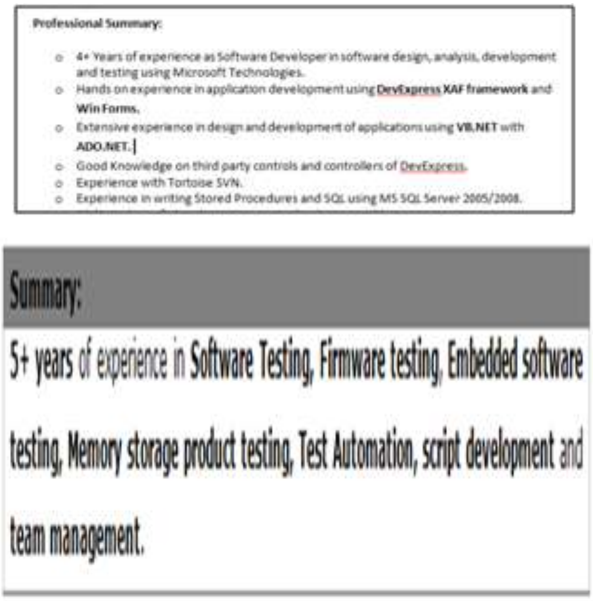
Figure 1.6: Sample resumes

In section to extract DOBIn the above sample The personal section will either contain the age or date of birth. This date of birth is extracted and then converted into age by separating the year from the text. From The extracted "01/07/1987" text, "1987" is separated and subtracted with the current year to get the age. [2015 – 1987] = 28 years.

The total experience is searched in the overall summary section, experience summary section or professional summary section to

fetch it out. Most of the resume will contain an explicit overall total experience in the aforementioned sections and it is extracted using regular expression.

Regular Expression = "\[0-9]\+\" is TRUE when a number followed by a "+" sign.


Professional Summary:

- 4+ Years of experience as Software Developer in software design, analysis, development and testing using Microsoft Technologies.
- Hands on experience in application development using DevExpress XAF framework and Win Forms.
- Extensive experience in design and development of applications using VB.NET with ADO.NET.
- Good Knowledge on third party controls and controllers of DevExpress.
- Experience with Tortoise SVN.
- Experience in writing Stored Procedures and SQL using MS SQL Server 2005/2008.

Summary:

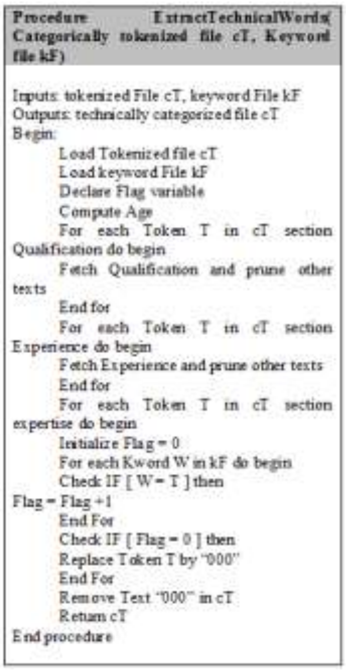
5+ years of experience in Software Testing, Firmware testing, Embedded software testing, Memory storage product testing, Test Automation, script development and team management.

Figure 1.7: Sample resumes with summary of experience section

From the above figure 1.7 two sample resume sections with summary are shown and the extracted total experiences values will be utilized for ranking purpose.

The qualification is searched in the academic section and the highest qualifications of the candidates are fetched and will be used for ranking purpose.

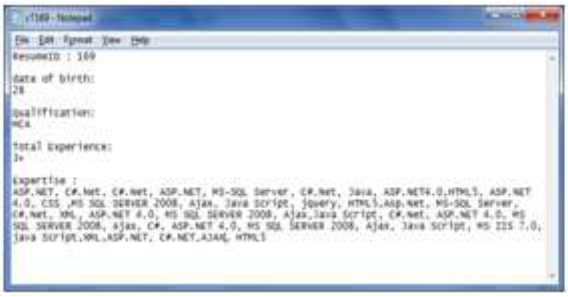
The skillsets of the candidates is extracted using keyword reference file where the unpromising texts are pruned away and the following procedure fetches the promising technical words from the categorically tokenized files.



```
Procedure ExtractTechnicalWords( Categorically tokenized file cT, Keyword file kF)

Inputs: tokenized File cT, keyword File kF
Outputs: technically categorized file cT
Begin:
        Load Tokenized file cT
        Load keyword File kF
        Declare Flag variable
        Compute Age
        For each Token T in cT section Qualification do begin
        Fetch Qualification and prune other texts
        End for
        For each Token T in cT section Experience do begin
        Fetch Experience and prune other texts
        End for
        For each Token T in cT section expertise do begin
        Initialize Flag = 0
        For each Kword W in kF do begin
        Check IF [ W = T ] then
Flag = Flag +1
        End For
        Check IF [ Flag = 0 ] then
        Replace Token T by "000"
        End For
        Remove Text "000" in cT
        Return cT
End procedure
```

Figure 1.8: Pseudo code of ExtractTechnicalWords

The keyword reference file is technical dictionary file comprises of the technical jargons, software names, tools and technologies present in the information technology verticals and this technical keyword file further reduces the categorically tokenized file size considerably and this output file is used to fetch the area of expertise using frequent text mining. The technical output text file is shown in the figure 1.9.



```
ResumeID : 169

date of birth:
28

qualification:
MCA

total Experience:
3+

Expertise :
ASP.NET, C#.Net, C#.Net, ASP.NET, MS-SQL Server, C#.Net, Java, ASP.NET4.0,HTML5, ASP.NET
4.0, CSS ,MS SQL SERVER 2008, Ajax, Java Script, jquery, HTML5,Asp.Net, MS-SQL Server,
C#.Net, XML, ASP.NET 4.0, MS SQL SERVER 2008, Ajax,Java Script, C#.Net, ASP.NET 4.0, MS
SQL SERVER 2008, Ajax, C#, ASP.NET 4.0, MS SQL SERVER 2008, Ajax, Java Script, MS IIS 7.0,
java Script,XML,ASP.NET, C#.NET,AJAX, HTML5
```

Figure 1.9: technically categorized file

The dimension of the file is further reduced from 2.02 KB to 496 bytes. The dimension of the text data is reduced from 67 KB to 496 bytes and the reduction ratio is almost above 99 %. The reduction ratio is calculated using the following formula,

$$Reduction\ Ratio = Actual\ File\ Size \\ /Reduction\ \ File\ \ Size^{\times}100$$

$$\left(\frac{[67000 - 496]}{67000}\right)^{\times}100 = 99.25\%$$

This huge amount of dimension reduction enhances the speed of execution, reduces the memory usage of the algorithm considerably.

3. RANKING

The skillset ranking is found based on the frequent technical words present in the resume and the area of expertise is found here. The procedure RankSkills is proposed to find the skillset ranking of the candidates and shown in the figure 1.10. The frequent text mining concept is utilized here to discover the area of expertise and skills strengths of the candidate.

```
Procedure RankSkills( Technically Categorized File Collection cT)
Output: Rank based mined File rF

Begin:
        Load File Collection cT
        For each File F in Collection cT do begin
Calculate total number of Tokens totTokens in F
For val = 1 to totTokens
        Initialize Count =0
        For val1 = val + 1 to totTokens
If [ Token[val] = Token[val1] and Token[val] ? "Found"] then
        Count = Count +1
        Replace Token[val1] by "Found"
        End IF
        End For
        Insert Token[val],Count into rF
        End For
        Find the largest Count value
        Area of Expertise = Token with largest Count
        Return rF
End Procedure
```

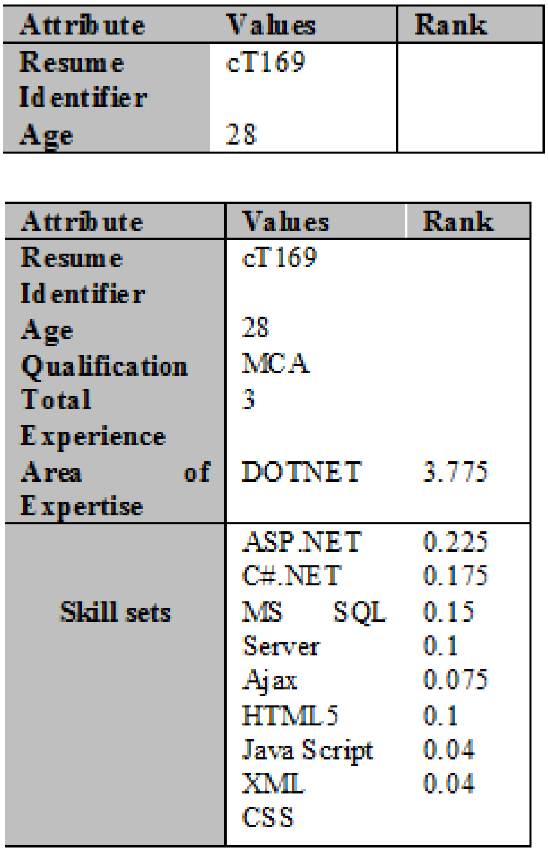
Figure 1.10: Pseudo code of RankSkills

From the sample ct169 file, the area of expertise is found out to be ASP.NET as the support count value for ASP.NET is 9, support count value of C#.Net is found to be 7. The area of expertise is discovered to be DOTNET. Remaining skills are ranked with the corresponding support count values as shown in the figure 1.11.

The ranking for the expertise is found using the following formula,

$$Expertise\ Rank = -1\ maximum\ \ support\ count \backslash total\ count^{\times} total\ experience$$

$$\left[1-\left(\frac{9}{40}\right)+3\right]=3.75$$



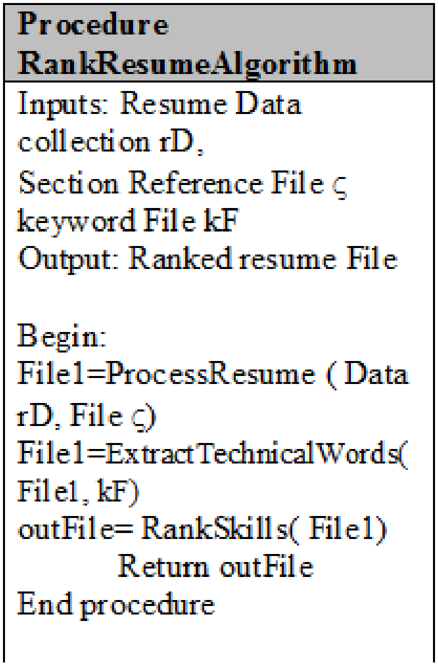
Attribute	Values	Rank
Resume Identifier	cT169	
Age	28	

Attribute	Values	Rank
Resume Identifier	cT169	
Age	28	
Qualification	MCA	
Total Experience	3	
Area of Expertise	DOTNET	3.775
Skill sets	ASP.NET	0.225
	C#.NET	0.175
	MS SQL	0.15
	Server	0.1
	Ajax	0.075
	HTML5	0.1
	Java Script	0.04
	XML	0.04
	CSS	

Figure 1.11: Extracted features for cT169 sample file

PROPOSED ALGORITHM

The proposed algorithm comprises of many sub procedures and the sub procedures are clearly enumerated in the previous sections and the proposed algorithm Rank Resume Algorithm RRA effectively combines the operations of these sub procedures to extract the features from the resume data collection and rank the resumes according to the weightage of experience and skills acquired by the candidates.

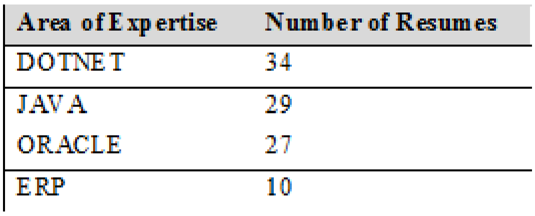


```
Procedure
RankResumeAlgorithm
Inputs: Resume Data
collection rD,
Section Reference File ς
keyword File kF
Output: Ranked resume File

Begin:
File1=ProcessResume ( Data
rD, File ς)
File1=ExtractTechnicalWords(
File1, kF)
outFile= RankSkills( File1)
        Return outFile
End procedure
```

Figure 1.12: Proposed RRA algorithm

EXPERIMENTAL SETUP

The proposed algorithm RRA is implemented using Visual Basic and the system configuration used is dual core 2.6GHz Processor with 1GB RAM. A set of 100 resumes are used for the evaluation purpose and the training resume collection is shown in the table 1.1.



Area of Expertise	Number of Resumes
DOTNET	34
JAVA	29
ORACLE	27
ERP	10

Table 1.1: Trained resume data collection

The genuineness of the texts extracted from the resume is measured by two parameters called, Precision and Recall. Evaluation methods are useful in evaluating the usefulness and trustfulness of the extracted texts from the resumes. Two main criterion for evaluating the proficiency of a system is precision and recall which are used for specifying the similarity between the objects which is generated by the system versus the one generated by human. The terms are defined in the following equations.

$$precision = (Actual)/(Actual + wrong$$

$$Recall = (Actual)/(Actual + missed)$$

To test the precision and recall of the proposed algorithm RRA, 100 resumes are taken and evaluated. The total number of resumes with DOTNET expertise discovered by the proposed algorithm was 32, whereas the trained resume contains 34 DOTNET experts but the proposed algorithm wrongly identified 2 resumes as DOTNET and missed 4 resumes.

$$Precision = \frac{30}{30+2} = 0.93$$

$$Recall = \frac{30}{30+4} = 0.88$$

The correct texts are the number of objects or texts produced by the system and the human, wrong is the number of objects or texts produced by the system alone. Therefore the precision is helpful to find the number of texts the system extracts and the recall is helpful in finding the number of texts the system misses during feature extraction. The precision and recall calculated for the sample resumes is shown in table 1.2.

$$over\ precision\ for\ 100\ resumes = \frac{91}{91+8} = 0.91$$

$$overall\ recall\ for\ resumes = 91(91+16) = 0.85$$

EXPERTISE	MRF	ARF
DOTNET	0.79	0.84
JAVA	0.81	0.93
ORACLE	0.77	0.81
ERP	0.62	0.73

Table 1.2: Precision and recall calculation

The relevancy factor of the resumes extracted is calculated using two factors, namely Machine Relevancy Factor and Actual Relevancy Factor.

$$MRF = total\ number\ of\ relevant \frac{resumes}{total\ resumes\ in\ collection}$$

$$ARF = actual\ number\ of\ relevant \frac{resumes}{total\ resumes\ in\ collection}$$

The total number of resumes discovered is equal to 84.

MRF = 84 / 100 = 0.84

The actual number of resumes discovered manually is 91 and the ARF is calculated as,

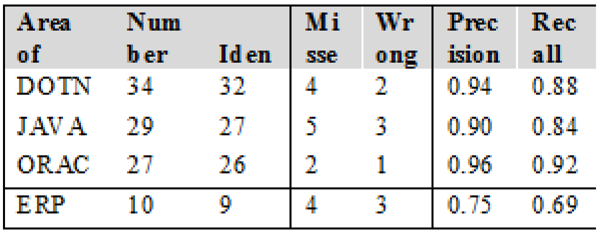
ARF = 91 / 100 = 0.91

Hence the Accuracy = MRF/ ARFx 100= (0.84/0.91)*100 = 92.30%

The accuracy of the proposed algorithm when 100 numbers of resumes are tested is 92.30 %.

Relevancy factor calculated The calculated machine relevancy and actual relevancy are noted and shown in the table 1.3

Area	Resume ID	Common	Unique Skills
DOT NET	3 – Score 5.778 12 – Score 5.67 19 – Score 5.22 23- Score 4.88 81 – Score 4.72 7 – Score 4.49	ASP.NET MS SQL Server Java Script	C#.NET AJAX MVC Tortoise SVN Silverlight Agile
	6 – Score 5.83 4 – Score 5.67 88 – Score 5.16 96 – Score 4.75 16 – Score 4.19 67 – Score		C#.NET CSS WCF HTML5 ADO.NET Linq
	17 – Score 5.69 12- Score 5.51 2 – Score 4.81 90 – Score 4.58 72 – Score 4.18 25 – Score 4.08		VB.NET Crystal Reports Oracle 10 IIS 7.0
ORA CLE	14 – Score 5.61 95 – Score 4.38 35 – Score 3.89 8 – Score 3.51 28 – Score 4.86 79 – Score 4.31 37 – Score 3.25	ORACLE 10G PL/SQL	Clearcase SSIS Goldengate
	13- Score 5.77 4 – Score 5.32 48 – Score 4.57 39 – Score 4.38 16 – Score 3.88		DATA Pump RMAN Solaris



Area of	Number	Iden	Misse	Wrong	Precision	Recall
DOTN	34	32	4	2	0.94	0.88
JAVA	29	27	5	3	0.90	0.84
ORAC	27	26	2	1	0.96	0.92
ERP	10	9	4	3	0.75	0.69

Table 1.3: Relevancy factor for 100 resume sample

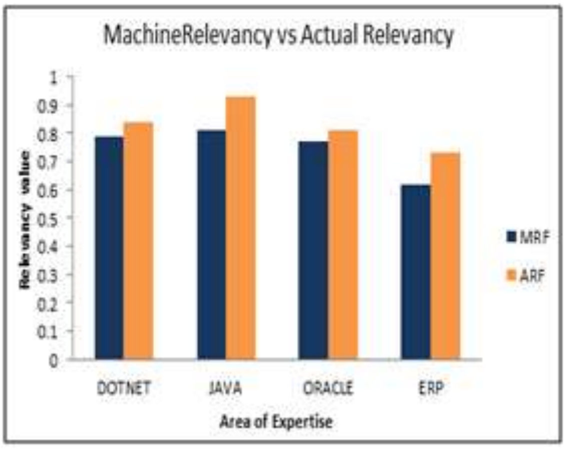


Figure 1.12: Relevancy graphs for 100 resume sample

From the figure 1.12 and table 1.3, it is quite evident that the proposed algorithm has a good accuracy level and the performance of the proposed algorithm RRA is excellent.

The memory reduction of the proposed algorithm is calculated for these 100 numbers of resumes. The actual memory size of the resume. collection is 69.5 MB and the proposed algorithm reduces this to a very small size of 1.31 MB. The memory footprints related to the proposed algorithm is very low and this enhances the speed of the execution. The overall extracted result for the 100 trained resumes is shown in the table 1.4. Table 1.4: Sample Resumes categorized according to area of expertise

FUTURE WORK AND CONCLUSION

The proposed approach proves to be efficient related to run time and memory usage but there is always room for further research and improvement can include other features that appear in a resume like certifications, co-curricular activities, interests, present work location, gender etc. The scores for unique skillset calculated can also be extended to combine with the scores or weightage of education, weightage related to companies worked earlier, weightage related to the complexity of the projects handled. Clustering approaches can be employed to classify the resume according to the categories and distance metrics can be applied to find the similarities between the resumes.

The new proposed approach has numerous facets that are still subject to further investigation and possible improvements. The robustness of the proposed approach can be further evaluated by running further tests on new and large datasets.

There are problems in resume extraction and the selection of appropriate resumes from a huge collection of resumes. An attempt has been made to pull out the appropriate resumes by choosing them based on area of expertise and then highlighting their unique skills. The proposed approach utilizes a scoring technique to select the best high scoring or ranked resumes from the resume set collection. The ranking score based area of expertise and the unique skills of a resume determine the uniqueness of a resume. This enhances the recruiters work to read through a set of resumes with their scores, unique skillset specialties to decide on prospective candidates for hiring.

REFERENCES

[1] M.A.K. Halliday and R. Hasan, *"Cohesion in English"*, Longman, 1976.

[2] Daxtra CVX. http://www.daxtra.com/

[3] Sovren Résumé/CV Parser. http://www. sovren.com/

[4] ALEX resume parsing http://www.hierarchy.

[5] RésuméGrabber http://www.egrabber.com/résumégrabbersuite/.

[6] Akken Stuffing. http://www.akken.com/.

[7] SumitMaheshwari "Mining Special Features to Improve the Performance of Product Selection in E-commerce Environment and Résumé Extraction System" (ICEC 2009) Taipei, Taiwan, August 2009, Published by ACM.

[8] SumitMaheshwari, AbhishekSainani, and P. Krishna Reddy, "An Approach to Extract Special Skills to Improve the Performance of Résumé Selection". In Proceedings of DNIS. 2010, 256-273.

[9] AbhishekSainani, "Extracting Special Information to Improve the efficiency of Résumé Selection Process," June 2011.

[10] Ming "Résumé information extraction with cascaded hybmodel".'05:of the 43rd Annual. .Meeting on Association for Computational Linguistics, pages 499–506, Morristown, NJ, USA, 2005.