JOB RECOMMENDATION SYSTEM USING RESUME DATA EXTRACTION

Vijay Gaikwad, Siddhesh Girase, Akhilesh Belanke, Shubham Sapkal,
Department of Electronics Engineering,
Vishwakarma Institute of Technology, Pune, INDIA

Abstract: In this project we have addressed need of students, where this project helps the final year students to find the correct job of them require skill-set which also matches with the company profile. Students just need to upload their resumes in pdf or image form. Our interface will extract the data from the resumes using Optical Character Recognition (OCR). This data is then processed and compared with the existing database and the companies are recommended to the students. This application will save the time of the students which they earlier spent on searching the companies. Thus, will increase the productivity and efficiency of the overall process. The model we proposed is verified through experiments study which is using actual data. The recommended results can achieve higher score of precision and recall, and they are more relevant with users' preferences before.

I. INTRODUCTION

In India there are about 13 lakh university students graduating every year, dealing with the enormous amount of recruiting information on the Internet, a job seeker always spends hours to find useful ones. To reduce this laborious work, we design and implement a recommendation system. The latest technology designed to fight information overload is the recommender systems that originated from cognitive science, approximation theory, information retrieval, forecasting theories and also related to management science and to consumer choice modelling in marketing. The recommender systems used to determine the interested items for a specific user by employing a variety of information resources that is related to users and items. Many researches in industry and academic areas have been known to develop new approaches for recommender systems in the last decade. The interest in this area still remains high because it is composed of a problem-rich research area and has a wealth of practical applications. Recommender systems are being broadly accepted in various applications to suggest products, services, and information items to latent customers. Many e-commerce applications join recommender systems in order to expand customer services, increase selling rates and decrease customers search time. For example, a wide range of companies such as the online book retailer Amazon.com, books, and news articles. Additionally, Microsoft provides users many recommendations such as the free download products, bug fixes and so forth. All these companies have successfully set up commercial recommender systems and have increased web sales and improved customer fidelity. Moreover, many software developers provide stand-alone generic recommendation technologies. The top providers include Net Perceptions, Epiphany, Art Technology Group, Broad Vision, and Blue Martini Software.

II. RESEARCH METHODOLOGY

The paper is divided into two parts first is image processing and data extraction and second is data base creation and data matching

1. Image processing:

The image processing part is further divided into three parts namely, image pre-processing, OCR and natural language processing. All the parts are done using python 3.5.

We start with loading the image using PILLOW or OPENCV 3 library. When the image gets loaded it is in the form of matrix. This matrix is stored in the variable. Now this variable will act as an image, which is further used to carry out the required operations. The values inside this matrix are the pixel intensity at a particular point. The particular pixel intensity has 3 channels namely - BLUE, GREEN, RED. This is because every colour in the world can be represented by combination of these 3 colours. It is observed from our experiments that PILLOW gives results that are more reliable. So we chose PILLOW in our case. We get this image in BGR, three channelled, format, which doesn't have high contrast. To solve this problem we first convert it to a grayscale image that is a single channel image and apply histogram equalisation to get high contrast. We then apply threshold and blurring techniques to reduce the noise. Then use a standard kernel to filter and in process sharpen the image. After all these steps, our image is ready for OCR detection. The below diagram clearly shows the pipeline of formation of an image and how the image is stored. Thus, all the operations and manipulations that take place on an image are done pixel by pixel. Where each pixel value is taken and passed through the different mathematical equation to get the desired output. This in turn solves another problem where the resume can have different coloured texts in it, by thresholding process we decrease possible probability of failure due to colour mismatch.

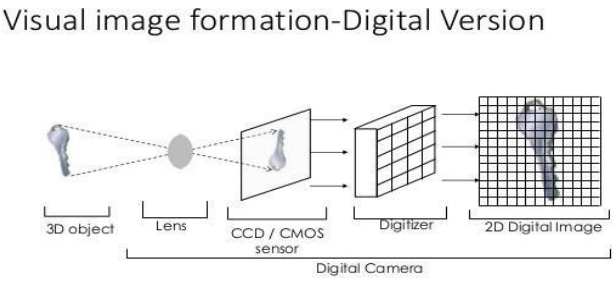


Figure 1: Formation of a Digital Image

Figure 2: Comparison of different grayscale methods

For the second part we use pytesseract library which is an open-source OCR library developed by UB-Mannheim. Originally, Hewlett and Packard developed Tesseract OCR. After downloading the necessary dependencies, we can easily run OCR on our filtered image. OCR means optical character recognition, which helps us mine the words written in a scanned image. This process returns us a string format, which contains all the text information in the scanned file. Although we get all the letters in the image, the formatting information is lost. To solve this problem we detect all characters that match with '\n' and replace it with a full stop. This helps get complete sentences instead of senseless, unformatted data. We now create another array where we have all the keywords that needs to be found in this mined data. A good way to do that is to run the string in a 'for loop' and detect each word individually. If the word is present we change the status of that position from 0 to 1. This string will later be used to inform our data base about the matches made successfully.

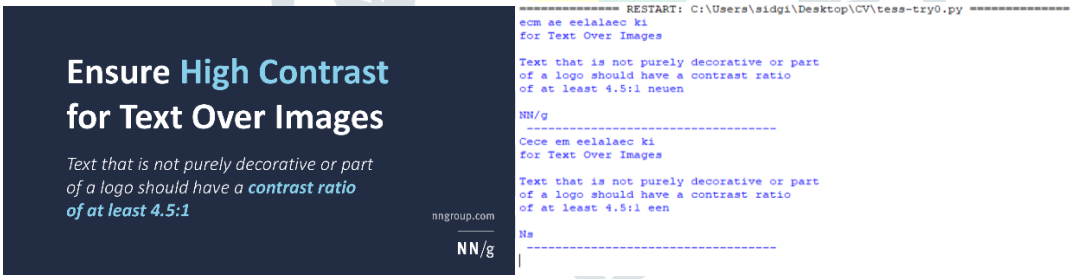


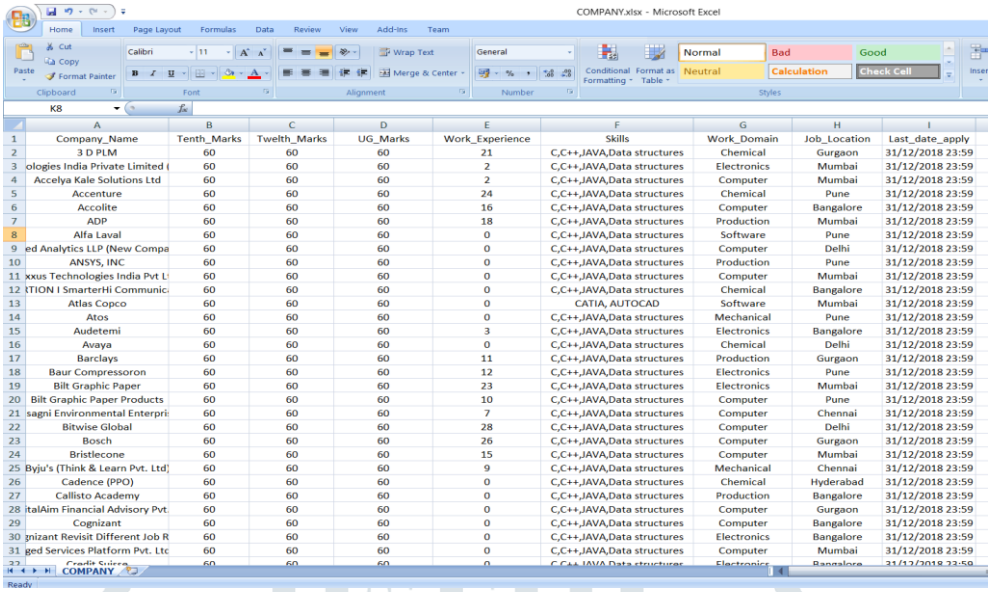
Figure 3: Input grayscale image and different OCR outputs using PILLOW and OpenCV.

For the last part in image processing section we use nltk library which is a Natural Language Processing library. This tool is used to get the tokenised version of our detected string and reduce the redundant data. Redundant data means non useful or repetitive words like articles or prepositions that are not useful for our text mining. With this step we extract all the Nouns, Noun phrases, Verbs, Verb phrases, Adjectives, Number data to get the exact information about skills, work experience, marks, grades etc. This helps us reduce the number of iterations to find a match and generalized keywords in data becomes smaller.

2. Database:

The methodology section outline the plan and method that how the study is conducted. This includes University of the study, sample of the study, Data and Sources of Data, study's variables and analytical framework. The details are as follows; This is one of the challenging task of the project where we compiled the data of different companies on the basis of their job requirement. The data comprises of the company name, job description and the skill required to that specific job. This collection of data from the companies was survey conducted as part of the project. Where the different companies were searched online for their requirements, and the data was recorded. The skills required list of different companies was

represented in the tabular form and afterwards it was passed to MySQL for the purpose of database handling and database management.



	A	B	C	D	E	F	G	H	I
1	Company_Name	Tenth_Marks	Twelth_Marks	UG_Marks	Work_Experience	Skills	Work_Domain	Job_Location	Last_date_apply
2	3 D PLM	60	60	60	21	C,C++,JAVA,Data structures	Chemical	Gurgaon	31/12/2018 23:59
3	ologies India Private Limited (	60	60	60	2	C,C++,JAVA,Data structures	Electronics	Mumbai	31/12/2018 23:59
4	Accelya Kale Solutions Ltd	60	60	60	2	C,C++,JAVA,Data structures	Computer	Mumbai	31/12/2018 23:59
5	Accenture	60	60	60	24	C,C++,JAVA,Data structures	Chemical	Pune	31/12/2018 23:59
6	Accolite	60	60	60	16	C,C++,JAVA,Data structures	Computer	Bangalore	31/12/2018 23:59
7	ADP	60	60	60	18	C,C++,JAVA,Data structures	Production	Mumbai	31/12/2018 23:59
8	Alfa Laval	60	60	60	0	C,C++,JAVA,Data structures	Software	Pune	31/12/2018 23:59
9	ed Analytics LLP (New Compa	60	60	60	0	C,C++,JAVA,Data structures	Computer	Delhi	31/12/2018 23:59
10	ANSYS, INC	60	60	60	0	C,C++,JAVA,Data structures	Production	Pune	31/12/2018 23:59
11	xxus Technologies India Pvt L	60	60	60	0	C,C++,JAVA,Data structures	Computer	Mumbai	31/12/2018 23:59
12	ITION I SmarterHi Communic	60	60	60	0	C,C++,JAVA,Data structures	Chemical	Bangalore	31/12/2018 23:59
13	Atlas Copco	60	60	60	0	CATIA, AUTOCAD	Software	Mumbai	31/12/2018 23:59
14	Atos	60	60	60	0	C,C++,JAVA,Data structures	Mechanical	Pune	31/12/2018 23:59
15	Audetemi	60	60	60	3	C,C++,JAVA,Data structures	Electronics	Bangalore	31/12/2018 23:59
16	Avaya	60	60	60	0	C,C++,JAVA,Data structures	Chemical	Delhi	31/12/2018 23:59
17	Barclays	60	60	60	11	C,C++,JAVA,Data structures	Production	Gurgaon	31/12/2018 23:59
18	Baur Compressoron	60	60	60	12	C,C++,JAVA,Data structures	Electronics	Pune	31/12/2018 23:59
19	Bilt Graphic Paper	60	60	60	23	C,C++,JAVA,Data structures	Electronics	Mumbai	31/12/2018 23:59
20	Bilt Graphic Paper Products	60	60	60	10	C,C++,JAVA,Data structures	Computer	Pune	31/12/2018 23:59
21	sagni Environmental Enterpri	60	60	60	7	C,C++,JAVA,Data structures	Computer	Chennai	31/12/2018 23:59
22	Bitwise Global	60	60	60	28	C,C++,JAVA,Data structures	Computer	Delhi	31/12/2018 23:59
23	Bosch	60	60	60	26	C,C++,JAVA,Data structures	Computer	Gurgaon	31/12/2018 23:59
24	Bristlecone	60	60	60	15	C,C++,JAVA,Data structures	Computer	Mumbai	31/12/2018 23:59
25	Byju's (Think & Learn Pvt. Ltd	60	60	60	9	C,C++,JAVA,Data structures	Mechanical	Chennai	31/12/2018 23:59
26	Cadence (PPO)	60	60	60	0	C,C++,JAVA,Data structures	Chemical	Hyderabad	31/12/2018 23:59
27	Callisto Academy	60	60	60	0	C,C++,JAVA,Data structures	Production	Bangalore	31/12/2018 23:59
28	talAim Financial Advisory Pvt	60	60	60	0	C,C++,JAVA,Data structures	Computer	Gurgaon	31/12/2018 23:59
29	Cognizant	60	60	60	0	C,C++,JAVA,Data structures	Computer	Bangalore	31/12/2018 23:59
30	gnizant Revisit Different Job R	60	60	60	0	C,C++,JAVA,Data structures	Electronics	Bangalore	31/12/2018 23:59
31	ged Services Platform Pvt. Ltc	60	60	60	0	C,C++,JAVA,Data structures	Computer	Mumbai	31/12/2018 23:59

Figure 8: Database of Companies

Creating SQLite database

Company is recommended to the candidate as final output recommendation. This recommendation is based on the different parameter such as skills, marks, work experience which are automatically extracted from the resume of the candidate. And based on information extracted from the candidates resume the company is recommended based on the following parameters.

```
sqlite3.connect('Type your DataBase name here.db')
```

Creating a database in Python using sqlite3:

(1) Create a new database called 'CompanyDB.db'
(2) Create a table: Company
(3) Import CSV file : "Company.csv"
(4) The imported client data from the CSV file will be stored in the Company table.
(5) Use an INSERT INTO statement to store the linked data.
(6) Utilize a WHERE condition to display the records those satisfies predicate condition.

```
import csv, sqlite3

con = sqlite3.connect("CompanyDb.db")
cur = con.cursor()
cur.execute("""CREATE TABLE IF NOT EXISTS Company (
                                    Company_Name text NOT NULL PRIMARY KEY,
                                    Tenth_Marks integer  NOT NULL,
                                    Twelth_Marks integer  NOT NULL,
                                    UG_Marks integer NOT NULL,
                                    Work_Experience integer,
                                    Skills text NOT NULL,
                                    Work_Domain text NOT NULL,
                                    Job_Location text NOT NULL,
                                    Last_date_apply DATETIME
                                );""") # use your column names here

with open('COMPANY.csv','rt') as fin: # `with` statement available in 2.5+
    # csv.DictReader uses first line in file for column headings by default
    dr = csv.DictReader(fin) # comma is default delimiter
    to_db = [(i['Company_Name'], i['Tenth_Marks'], i['Twelth_Marks'],i['UG_Marks'], i['Work_Experience'], i['Skills'

cur.executemany("INSERT INTO Company (Company_Name, Tenth_Marks, Twelth_Marks, UG_Marks, Work_Experience, Skills, Wo
con.commit()
con.close()
```

Figure 9: Program for importing CSV file and storing its data in database

Queries for selecting required companies.

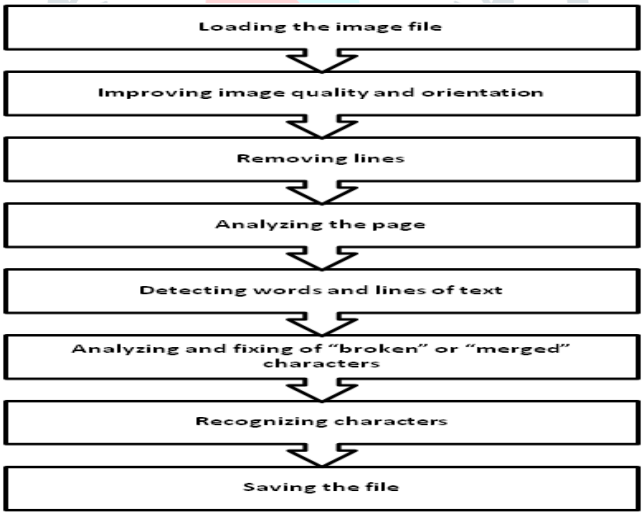
Select query to retrieve data from SQLite database from Python, you use these steps:

1. First, establish a connection to the the SQLite database by creating a Connection object.
2. Next, create a Cursor object using the cursor method of the Connection object.
3. Then, execute the SELECT statement.
4. After that, call the fetchall() method of the cursor object to fetch the data.
5. Finally, loop the cursor and process each row individually.

Where Clause is used with SELECT statement to filter data returned by the query. The WHERE clause is also known as a set of conditions or a predicate list.
When evaluating a SELECT statement with a WHERE clause, SQLite uses the following steps:
1) First, check the table in the FROM clause.
2) Second, evaluate the conditions in the WHERE clause to get the rows that met the conditions.
3) Third, make the final result set based on the rows in the previous step with columns in the SELECT clause.
cur=conn.cursor()
cursor = cur.execute("**SELECT** Company_Name, Last_date_apply, Job_Location from Company **where** Tenth_Marks<="+res[0]+"andTwelth_Marks<="+res[1]+"andUG_Marks<="+res[2]+"and Work_Experience<="+res[3]+"andLast_date_apply<=date('now');")
Here, We are using WHERE clause for sorting out the companies those have satisfied the conditions like Tenth marks, Twelve marks, Graduation Marks, Work experience, Skills, Work domain, job location and last date to apply. The student details are extracted from the *list* which was generated by executing Computer vision, OCR functions and then by tokenizing the text strings.

```
cursor = cur.execute("SELECT Company_Name, Last_date_apply, Job_Location from Company where Last_date_app
print(cursor)
print("You can apply for following companies :")
for row in cursor:
  print("\nCompany name: ",row[0],"\nLast date to apply :",row[1],"Job location :",row[2])
  print("_____________________________________________________________________")
conn.close()
```

Figure 10: Select and Where Clause
In addition to sqlite3, There is need to import CSV and sqlite3 package. The CSV package is an essential component that will be used to:
Import the CSV files using the “open('COMPANY.csv','rt') ” command
Assign the values imported from the CSV files into the tables, so that we can then execute the SQL queries.
Assign the SQL fields into the dataframe.
The above steps can be summarized using the following flowchart.



IV. RESULTS AND DISCUSSION

Resulting Product

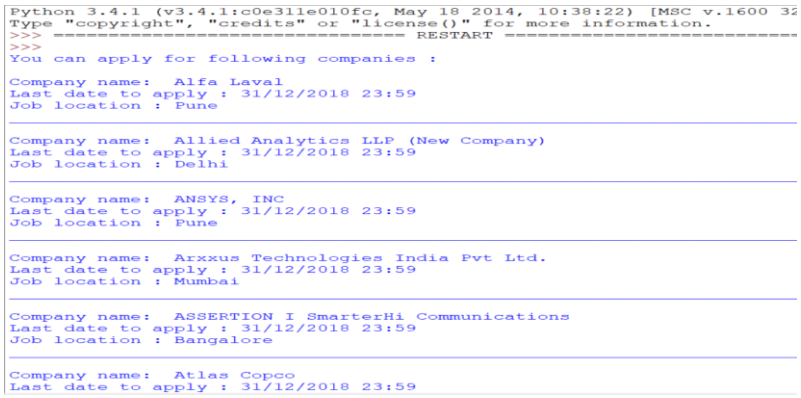


Figure 11: Displaying Recommended Companies

Conclusion

In this project, we have seen from our literature review and from the challenges that faced the holistic e-recruiting platforms, an increased need for enhancing the quality of candidates/job matching. The recommender system technologies accomplished significant success in a broad range of applications and potentially a powerful searching and recommending techniques. Consequently, there is a great opportunity for applying these technologies in recruitment environment to improve the matching quality. This survey shows that several approaches for job recommendation have been proposed, and many techniques combined in order to produce the best fit between jobs and candidates. We presented state of the art of job recommendation as well as, a comparative study for its approaches that proposed by literatures. Additionally, we reviewed typical recommender system techniques and the recruiting process related issues. We conclude that the field of job recommendations is still unripe and require further improvements.

The company which comes for the interview, post it requirement on the mail and students then start to look for the company profile and the job description. They then have to analyze whether the company is suitable for them or not. This consumes lot of time of the students to arrive at the conclusion. Instead our project will help all the students. We already have database of huge amount of companies. Students will upload their resume on our interface and our system will extract the information from their resume automatically. Make a comparison with the company requirements and candidate skill, and recommend the student the company which he/she should be applying for. Thus the candidate will apply for the recommended companies on high priority.

REFERENCES

[1] Abhishek Sainani, "Extracting Special Information to Improve the efficiency of Resume Selection Process," June 2011.
[2] B. L. Hawkins, J. A. Rudy and W. H. Wallace "Recruiting, Retaining, and Reskilling Campus IT Professionals. Technology Everywhere: A Campus Agenda for Educating and Managing Workers in the Digital Age". Dolan, A. F. 2004. Jossey-Bass: 75-- 91.
[3] Baeza Yates, Ricardo, and Berthier Ribeiro-Neto. Modern information retrieval (1999). New York: ACM press, 463.
[4] Dorre, Jochen, Peter Gerstl, and Roland Seiffert (1999). Text mining: finding nuggets in mountains of textual data. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 398-401. ACM, 1999.
[5] Fan, Weiguo, Linda Wallace, Stephanie Rich, and Zhongju Zhang (2006). Tapping the power of text mining. Communications of the ACM, 49(9): 76-82.
[6] Jayaraj, V., and V. Mahalakshmi. "Augmenting Efficiency of Recruitment Process using IRCF text mining Algorithm." Indian Journal of Science and Technology 8.16 (2015).
[7] Jayaraj, V., and V. Mahalakshmi. "Information Retrieval Configuration File Text Categorization Algorithm for Improving Business Intelligence." International Journal Of Computational Engineering And Management"(IJCEM), ISSN:2230-7893, January 2015.
[8] Kun Yu, Gang Guan, and Ming Zhou, "Resume information extraction with cascaded hybrid model". In ACL "05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 499–506, Morristown, NJ, USA, 2005.
[9] Lecerf, Loic, and Boris Chidlovskii (2009). Scalable Feature Extraction from Noisy Documents. Document Analysis and Recognition, ICDAR'09. 10th International Conference on. IEEE.
[10] Liu, Luying, Jianchu Kang, Jing Yu, and Zhongliang Wang. A comparative study on unsupervised feature selection methods for text clustering (2005). In Natural Language Processing and Knowledge Engineering, 2005. IEEE NLP-KE'05. Proceedings of 2005 IEEE International Conference, 597-601.
[11] Norm Schneider, "Job Hunters: Resume Filters May Help or Hinder Your Job Search," Published on August 23, 2011 Employment http://bizcovering.com/employment/job- hunters-resume-filters-may-help-or-hinder-your-jobsearch/#VNEJ2.

[12] Rathi, VP Gladis Pushpa, and S. Palani (2012). A novel approach for feature extraction and Selection on mri images for brain tumor Classification. CCSEA, SEA, CLOUD, DKMP, CS & IT 5, 225-234.
[13] Sumit Maheshwari, Abhishek Sainani, and P. Krishna Reddy, "An Approach to Extract Special Skills to Improve the Performance of Resume Selection". In Proceedings of DNIS. 2010, 256-273.
[14] V.Jayaraj and V.Mahalakshmi, "Improving Text Categorization Using Configuration File" American International Journal of Research in Science, Technology, Engineering & Mathematics September 2014.
[15] Xing Yi, James Allan and W. Bruce Croft, "Matching Resumes and Jobs Based on Relevance Models". In Proceedings, SIGIR 2007.
[16] H. T. Yu, C. R. Liu and F. Z. Zhang, "Reciprocal recommendation algorithm for the field of recruitment," Journal of Information & Computational Science, vol. 8(16), pp. 4061-4068, 2011.
[17] J. Malinowski, T. Keim, O. Wendt and T. Weitzel, "Matching people and jobs: a bilateral recommendation approach," In Proceedings of The 39th Hawaii International Conference on System Sciences, pp. 1-9, Hawaii, USA, 2006.
[18] L. Li and T. Li, "MEET: a generalized framework for reciprocal recommender systems," In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 35-44, Hawaii, USA, 2012.
[19] R. Burke, "Hybrid web recommender systems," The Adaptive Web, vol. 4321, pp. 377-408, 2007.
[20] T. Keim, "Extending the applicability of recommender systems: a multilayer framework for matching human resources," In Proceedings of 40th Annual Hawaii International Conference on System Sciences, pp. 169-178, January, 2007.
[21] M. Fazel-Zarandi and M. S. Fox, "Semantic matchmaking for job recruitment an ontolgy based hybrid approach," In Proceedings of the 3rd International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web at the 8th International Semantic Web Conference, Washington D. C., USA, 2010.
[22] R. Rafter, K. Bradley and B. Smyth, "Personalised retrieval for online recruitment services," In Proceedings of the 22nd Annual Colloquium on IR Research, Cambridge, UK, 2000.
[23] A. Singh, C. Rose, K. Visweswariah, V. Chenthamarakshan and N. Kambhatla, "PROSPECT: a system for screening candidates for recruitment," In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 659-668, Toronto, Canada, 2010.
[24] M. Hutterer, "Enhancing a job recommender with implicit user feedback," In Fakultät für Informatik, Technischen Universität Wien, 2011.
[25] B. Keith and S. Barry, "Personalized information ordering: a case study in online recruitment," Knowledge Based Systems, vol. 15(5-6), pp. 269-275, 2003.