Developing a Framework for Potential Candidate Selection

Farzana Yasmin[1], Mohammad Imtiaz Nur[2], Mohammad Shamsul Arefin[3]
Computer Science and Engineering, Chittagong University of Engineering and Technology
Chattogram, Bangladesh

***Abstract*—Recruitment is the process of hiring the right person for the right job. In the current competitive world, recruiting the right person from thousands of applicants is a tedious work. In addition, analyzing these huge numbers of applications manually might result into biased and erroneous output which may eventually cause problems for the companies. If these pools of resumes can be analyzed automatically and presented to the employers in a systematic way for choosing the appropriate person for their company, it may help the applicants and the employers as well. So in order to solve this need, we have developed a framework that takes the resume of the candidates, pull out information from them by recognizing the named entities using machine learning and score the applicants according to some predefined rules and employer requirements. Furthermore, employers can select the best suited candidates for their jobs from these scores by using skyline filtering.**

Keywords—Information extraction; named entity recognition; machine learning; skyline queries

I. Introduction

Information extraction (IE) infers the process of automatically gisting of information in a strucutred way from unstructured and/or semi-structured machine-readable documents. The task involves the utilization of natural language processing (NLP). The present purpose of IE refers to the growing amount of information available in unstructured form [1].

Nowadays huge volume of documents are found online and offline. Extracting information from these vast volumes of data manually is time consuming. Moreover generating some pattern from the extracted information has recently been a new challenge and prime concern of the modern technological era.

Recruitment is the process of searching and selecting best candidates for filling the vacant positions of an organization. Recruitment process requires planning, requirements setup strategy, searching candidates, screening the candidates according to the requirements and evaluation of the candidates. These steps are usually conducted by the Human Resource (HR) department of any company. Whenever there is a job opening for the vacant positions, large amount of applications are dropped. On the other hand, the recruiters may search applicants from a job portal placing their requirements. In both cases, searching and screening the best candidates from these applicants after assessing the ablites and qualifications manually, takes huge amount of time, cost and effort of the HR department as the volume of data are big. If we can develop an efficient system for extracting information from the resumes and process these information in an automated way so that only the relevant applications are presented to the recruiters, it will ease the work of the HR management. An automated system for choosing the potential candidates that best suits the position's requirements can increase the efficiency of the HR agencies greatly.

Therefore, in order to make the recruitment process easy, effective and automated, we have developed a framework of potential candidate selection system. To perform this task we have chosen a domain of document information extraction which can be helpful in choosing the best potential candidates for any job openings i.e. CV/resume document. This development task involves the information extraction based on natural language processing i.e. tokenization, parsing, named entity recognizer (NER) and utilizes skyline query processing which works well in filtering the non-dominating objects from database and also makes a new addition to this domain.

So the objectives of the system development can be summerized as follows: 1) To design an efficient information extraction system from documents like curriculum vitae, 2) To generate scores on different features based on extracted information, 3) To perform appropriate filtering of information using skyline queries and 4) To generate proper ranking system for candidate selection.

The rest of the paper is presented as follows: In Section II related works of the candidate ranking system development has been portrayed. The system architecture and design is elaborated in Section III. Section IV represents the implementation of our work with some experimental results. And finally, a conclusion over the work has been drawn in Section V.

II. Related Work

D. Celik [2] proposed an information extraction system for candidate selection where the information extraction was based on ontology. The proposed methodology used Ontology-based Resume Parser(ORP) to convert English and Turkish documents into ontological format. The proposed method constructed seven reference ontologies to extract the information and categorize them into one of these ontologies. Though the methodology worked good on information extraction but it did not describe any score generation mechanism to rank the candidates.

Another form of candidate selection was proposed by S. Kumari et. al. [3] where candidate selection was done by using

Naïve Bayes algorithm for classifying the candidate profiles. They also considered employers importance criteria. No description given of how the information extraction are done. Also it requires GPRS connection every time as it is online based.

R. Farkas et al. [4] worked on a method of extracting information for career portal where the information of applicants' are stored in a uniform data structure named HR-XML format. They used a CV parser to automatically extract data from the CV. It is basically template specific method and doesn't work for all formats of documents.

In [5], the authors used a hybrid cascade model for information extraction from CVs. In the first pass, the proposed method segments resume using Hidden Markov Model. The second pass uses HMM and SVM to extract further detailed information. The cascaded pipeline suffers from error propagation i.e. errors from first step are passed in the second pass and the precision and recall value decreases subsequently.

Information is extracted from resumes using basic techniques of NLP like word parsing, chunking, reg ex parser in [6]. Information like name, email, phone, address, education qualification and experience are extracted using pattern matching in this work. Some other online resume parsers are found in [7] & [8].

An algorithm for CV information extraction is developed in [9] which works in two step. In the first step, raw texts are retrieved as resume blocks. Then in the next step they developed a mechanism to identify the fact information from the resume like named entities.

There also have been developed some works using skyline queries. [10], [11] & [12] describes some algorithms for processing skyline queries with their implementation.

S. Patil et al. [13] developed a method for learning to rank resumes with the help of SVM rank algorithm. In [14], X. Yi et. al. applied a Structured Relevance Model to select resumes for a given post or to choose the best jobs for a given candidate based on their CV. İn [15] job narration are transformed into queries and lookup in database is performed. The top-ranked candidates get selected automatically from these queries. Some authors exploit additional information like social media information along with information gained directly from resumes [16]. Moreover, [17] takes consideration of data collected from the LinkedIn profile and personality traits from the personal blogs of the candidates. In [18], digital resumes of candidates are generated by extracting data from social networking sites like Facebook, Twitter and LinkedIn. Candidates are evaluated based on their digital resume and ranked accordingly. In [19], CVs are filled in a predefined format and the scoring and ranking process is based on Analytic Hierarchy Process (AHP).

Though many works have been developed for candidate recruitment, the use of skyline query in this sceneraio is relatively new approach and we have implemented this novel approach in our framework.

III. System Architecture and Design

The proposed framework works in 4 modules according to Fig. 1: Document processing module, Query Execution Module, Analysis & Output module and Storage module.

A. Processing Module

1) CV upload: First, the candidates may upload their resumes in the interface. After resumes are uploaded to the system it is considered as input for the processing module.Then information extraction process begins and we used a NLP module named spaCy [20] for the rest of the processing steps. Suppose, sample resumes as Fig. 2(a), (b) and (c) are uploaded in the system.

2) Conversion to text: The standard format of resumes for our system is considered english resumes in PDF format. At first, we need to convert the pdf into plain text using UTF-8 encoding. From [21], UTF stands for Unicode Transformation Format and '8' means it uses 8-bit blocks to represent a character. The number of blocks needed to represent a character varies from 1 to 4. UTF-8 is a compromise character encoding that can be as compact as ASCII but can also contain any Unicode characters.

3) Tokenization: After conversion to text, now we have our necessary text file. We start reading the text file and tokenize the whole document. Tokenization is the process of splitting a document into its smallest meaningful pieces named tokens. Tokenization is done using the language rule i.e. removing the white space, checking the exception rules like punctuation checking, abbreviation rules, etc.

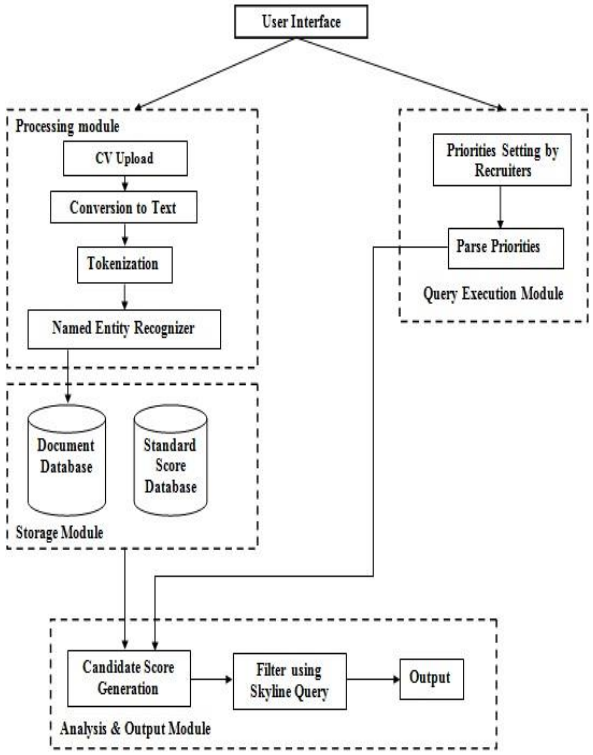
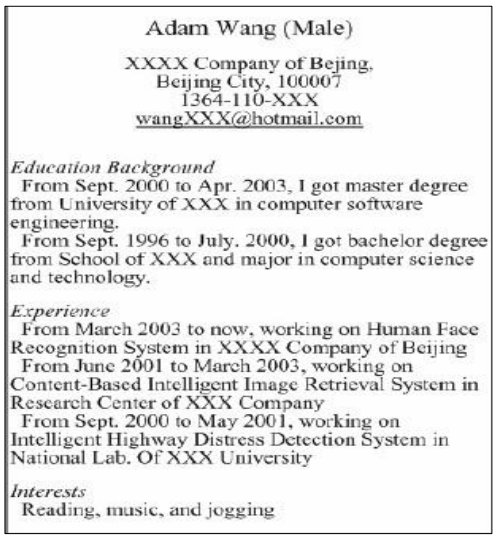


Fig. 1. System Architecture of Potential Candidate Selection System.



Adam Wang (Male)

XXXX Company of Bejing,
Beijing City, 100007
1364-110-XXX
wangXXX@hotmail.com

Education Background

From Sept. 2000 to Apr. 2003, I got master degree from University of XXX in computer software engineering.

From Sept. 1996 to July. 2000, I got bachelor degree from School of XXX and major in computer science and technology.

Experience

From March 2003 to now, working on Human Face Recognition System in XXXX Company of Beijing

From June 2001 to March 2003, working on Content-Based Intelligent Image Retrieval System in Research Center of XXX Company

From Sept. 2000 to May 2001, working on Intelligent Highway Distress Detection System in National Lab. Of XXX University

Interests

Reading, music, and jogging

(a)

ABC

Cell: +880 1680671851
E-mail: abc@gmail.com,

RESEARCH INTEREST

Data Mining, Artificial Intelligence, Machine Learning, Algorithm Design, Data Structure.

EDUCATIONAL QUALIFICATIONS

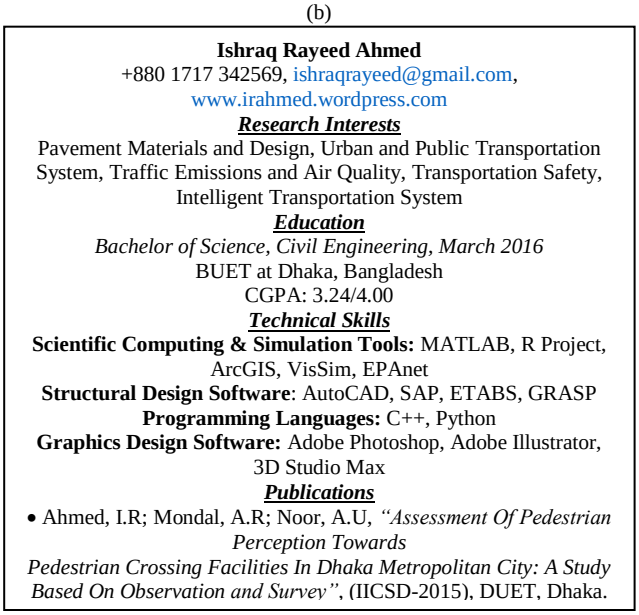
B.Sc. in CSE- October 2015
CUET, Bangladesh.
Result: CGPA 3.81 (out of 4.00)
Class Position: 2nd out of 113 students

WORK EXPERIENCES

Chittagong University of Engineering & Technology, Chittagong-4349, Bangladesh

- Joined as ***"Lecturer"*** as on 2016
- Duration : 2016- Present

(b)



Ishraq Rayeed Ahmed
+880 1717 342569, ishraqrayeed@gmail.com,
www.irahmed.wordpress.com

Research Interests

Pavement Materials and Design, Urban and Public Transportation System, Traffic Emissions and Air Quality, Transportation Safety, Intelligent Transportation System

Education

Bachelor of Science, Civil Engineering, March 2016
BUET at Dhaka, Bangladesh
CGPA: 3.24/4.00

Technical Skills

Scientific Computing & Simulation Tools: MATLAB, R Project, ArcGIS, VisSim, EPAnet
Structural Design Software: AutoCAD, SAP, ETABS, GRASP
Programming Languages: C++, Python
Graphics Design Software: Adobe Photoshop, Adobe Illustrator, 3D Studio Max

Publications

- Ahmed, I.R; Mondal, A.R; Noor, A.U, *"Assessment Of Pedestrian Perception Towards Pedestrian Crossing Facilities In Dhaka Metropolitan City: A Study Based On Observation and Survey"*, (IICSD-2015), DUET, Dhaka.

(c)

Fig. 2. (a), (b) and (c) Sample Resumes.

4) Named entity recognition: Named entity recognition (NER) is the most important task to do next. The success of the extraction process mainly depends on the accurately recognized entities from a resume. The subtask of information extraction that seeks to locate and classify named entity mentions in unstructured text into pre-defined categories such as the person names, organizations, email, phone, address, time, quantities, numeric values, etc. can be defined as Named entity recognition [22]. A statistical model is used to classify our desired entities in a standard resume like name, date of birth, email, phone number, university, education, major, publications, experience, skills, etc. The NER training model is designed using incremental parsing and residual CNNs. In case of training our model (Fig. 3) with the desired annotation we used resumes in JSON format.

The adapted algorithm of spaCy's NER training module is provided below:

Algorithm 1: Named Entity Recognition Training
Input: Tokens of the resumes
Goal: To identify the named entities required for information extraction

1. **Begin**
2. Annotate the training data manually
3. Initialize the annotated model, no. of iterations, output directory path
4. **If** model not loaded **do**
5. Load the initialized model
6. **End if**
7. **If** ner pipeline is not set **do**
8. Create *ner pipe*
9. Add the *ner pipe*
10. **Else** get *ner pipe*
11. **For** *annotations* in training data **do**
12. **For** *entities* in *annotations* **do**
13. Add *labels* of *entities*
14. **End for**
15. **End for**
16. Disabling other pipeline, begin the training
17. **For** iterations in range **do**
18. Shuffle the examples in batches
19. For each example update the model
20. **End for**
21. Save the model in the output directory
22. Test the model with the test data

At first, we have to manually annotate our training data in JSON format (2). Then we load or build the NER model (step 4-6). For training the NER model with our custom entities, now we add the labels for each annotations (step 11-15). For starting the training of our NER model, we must disable other pipeline components like tokenizer, tagger of spaCy (step 16). Then we shuffle and loop over our training examples (step 18). At each word the model makes a prediction. It then consults the annotations to see whether it was right. If it was wrong, it makes adjustment of the weight so that the correct action will score higher next time (step 19). Then we save the model (step 21) and test it to make sure the entities in the test data are recognized correctly (step 22).

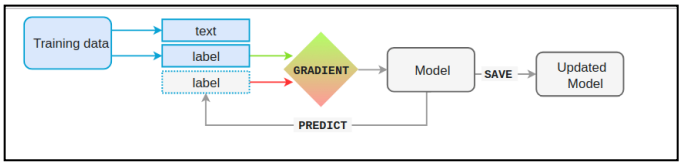


Fig. 3. SpaCy's NER Model Training Process (Source: [23]).

After the validation of the training of the NER model, now we use this model to extract the values of the entities from the resumes as blocks. The recognized entity values are stored block-wise in a row of a table for each candidate in the storage module. If we send the resumes in the NER model, the table of the extracted information takes the form like Table I, Table II, Table III, Table IV and Table V.

B. Storage Module

Storage module stores information processed by the processing module. The extracted information table after the entites are recognized are stored in the document database. We set the scores for different criterias based on some predefined hypothetical rules. The total storage is required for the candidate score generation in the analysis and output generation phase. The rules and scores that are set as lookup tables (Tables VI and VII) in the score database are given.

Rule for Experience:

If (*Designation*==*Required experience designation*)
  If (*Experience* == *Required position experience*)
    Score[Experience] = 5
Else
  If (*Total Experience* ≥ *Required experience*)
    Score[Experience] = 2
  Else *Score*[Experience] = 0}

Rule for Skills:

If (Candidate_skill== Required skill)
  Score[skill]= *Score*[skill]+ 1
Else *Score*[skill] = 0

Rule for Certification:

If (Candidate_certification== Required certification)
  Score[Certification]= Score[Certification]+ 1
Else *Score*[Certification] = 0

TABLE. I. EXTRACTED INFORMATION OF PERSONAL INFORMATION BLOCK

Name	Phone	Email	Date of Birth
Farzana Yasmin	01680671851	farzanaefu@gmail.com	30 Sep 1993
Mohammad Imtiaz Nur	01818772617	imti.nur@gmail.com	07-09-1992
Ohidul Islam	01617224955	ohid@gmail.com	14-12-1989
Faisal Karim	01826564578	faisal90@outlook.com	13.05.1994
Md. Intishar ur	01747678878	intishar788@gmail.com	6/1/1983
Ananna Das	01918793180	anannadas@yahoo.com	5-23-1987
Hasibul Haq	01922935210	h.haq602@yahoo.com	3 April 1990

TABLE. II. EXTRACTED INFORMATION OF EDUCATIONAL INFORMATION BLOCK

Name	Degree	Major	Institution	CGPA
Farzana Yasmin	B.Sc	CSE	Chittagong University of Engineering & Technology	3.81
Mohammad Imtiaz Nur	Bachelor in Science	CSE	CUET	3.01
Ohidul Islam	B.Sc., M.Sc.	Electrical Engineering	BUET, Jahangirnagar University	3.72,3.96
Faisal Karim	B.Sc., PhD	CSE	Chittagong University, University of Houston	3.50
Md. Intishar Nur	Bachelor in Science	Computer Science	IUT	3.46
Ananna Das	LLB	Law	Premier University	3.22
Hasibul Haq	Bachelor in Business Administration	Finance	IIUC	2.90

TABLE. III. EXTRACTED INFORMATION OF PUBLICATION BLOCK

Name	Publication
Farzana Yasmin	International Journal, International Conference
Mohammad Imtiaz Nur	--
Ohidul Islam	International Conference
Faisal Karim	International Journal, International Journal, International Journal, International Conference
Md. Intishar Nur	--
Ananna Das	--
Hasibul Haq	--

TABLE. IV. EXTRACTED INFORMATION OF EXPERIENCE BLOCK

Name	Company Worked at	Experience	Designation
Farzana Yasmin	CUET	3 yrs	Lecturer
Mohammad Imtiaz Nur	SAPL	2 yrs	Executive Programmer
Ohidul Islam	PDB	7 yrs	Executive Engineer
Faisal Karim	MatWorks	2.5 yrs	Senior Programmer
Md. Intishar Nur	BSRM	6 months	Assistant Programmer
Ananna Das	--	0	--
Hasibul Haq	Dhaka Bank Ltd.	5 yrs	Senior Officer

TABLE. V. EXTRACTED INFORMATION OF OTHERS BLOCK

Name	Skills	Certification
Farzana Yasmin	C, C++, PHP, Python, HTML, Java Script	CCNA
Mohammad Imtiaz Nur	Angualar, JavaScript, PHP, Java	Mobile Apps Training
Ohidul Islam	Python, Java, Javascript, Rubi on rails	--
Faisal Karim	C#, C++, PHP, Html, CSS, Javascript	--
Md. Intishar Nur	Matlab, C++	--
Ananna Das	MS Word, MS Office	--
Hasibul Haq	MS Word, MS Office, Linux	--

TABLE. VI. SCORE LOOKUP TABLE (PUBLICATION)

International Journal						Conference			
Indexing	Score	Publisher	Score	Impact Factor	Total	International Conference	Score	Others	Score
SCI	1	Nature	1	Value	Sum of Indexing, Publisher and Impact Factor score	Match the keyword 'International Conference'	.2	If doesn't satisfy other criteria of publication	.1
SCIE	.75	Springer, IEEE, Wiley, Elsevier	.5						
SCOPUS	.5	Others	.1						
Others	.4								
Predatory	0								

TABLE. VII. SCORE LOOKUP TABLE (EDUCATION)

Inst. Ranking	***PhD Score***	***M.Sc. Score***	***B.Sc. Score***	***CGPA***		***Major***	
1-200	10	5	2	***Rule***	***Score***	***Rule***	***Score***
201-500	9.5	4.8	1.9	≥ 3.75	4	Keyword Matching with Requirement	2
501-1000	9	4.6	1.8	3.5-3.74	3		
1001-1500	8.5	4.4	1.7	3.0 -3.49	2		
1501-2000	8	4.2	1.6	2.5-2.99	1		
2001-2500	7.5	4	1.5	Others	0		
2501-3000	7	3.8	1.4				
3001-3500	6.5	3.6	1.3				
3501-4000	6	3.4	1.2				
Others	5.5	3.2	1.1				

TABLE. VIII. REQUIREMENT SETTING TABLE

Job_criteria	Keywords
Skills	C++, Java, PHP
Experience	0-3 yrs as Executive/ Senior Programmer
Major	CSE, EEE

C. Query Execution Module

1) Priorities setting by the recruiter: In the UI, employers set the requirements for the vacant positions of their company. For example, for Senior Programmer position, the employer sets the following requirements as Table VIII for each criteria.

2) Parse priorities: The system will then parse these requirements of the employer in the query execution module.

D. Anaysis and Output Module

1) Candidate Score Generation: After parsing the requirement of the employer, the system will start the score table generation of each candidate according to the employer priority and previously set standard score for different categories from the score database. The algorithm of candidate score generation is given below:

Algorithm 2: Candidate Score Generation
Input: Extracted information stored in Excel file
Goal: To generate score of each candidate in each criterion
1. **Begin**
2. Initialize *Scores* object with unique *job_criteria*
3. Initialize an empty *Score_table* list
4. **For** each row in excel **do**
5. Set *Scores* object value to zero
6. **For** each job_info details **do**
7. Find(Excel(column))
8. **If** *job_criteria* == Excel(column) **do**
9. **If** keyword matches with column value **do**
10. Calculate the Scores value as:
Scores [job_criteria] += score set for the criteria in the score database
11. **Else** skip
12. **Else** skip
13. **End For**
14. Push *Scores* values in *Score_table*
15. **End for**
16. Set the mandatory required *job_criteria*
17. If *Scores [mandatory_job_criteria]* = 0 do
18. Delete the score row from the *Score table*

The extracted information stored in the lookup table in document database is retrieved (step 7-8) and matched with the keywords stored in the job_info_details table (step 9). If match found, the corresponding values are calculated as the rules set in the standard score table (step 10).

If multiple keywords are matched for a specific criteria, then they are stored as aggregated sum. For example, if multiple skills match, then all the skill values are added and stored in the skill column for that candidate.

For education score generation, the degrees and the ranking of the institutions they are acquired from are checked and scores for the institution and degree is put on the score table.

For the publication column, internation conference, international journal keywords are searched and matched. İf found, the number of occurences are counted. The lists of SCI, SCIE, SCOPUS and predatory journals are sotred in the database. The journal names are matched with these list. If match found, scores are calculated accordingly.

For the experience column, at first it is checked that the candidate is fulfilling the requirements for the given job. If yes, the experience score is calculated. If relevant experience is not fulfilled according to the requirements, the total experience is checked and given a score.

For skills an certifications score calculation, the requirement of the employers are checked. If matched, the information is given a value for each matched keywords.

If any column information contains missing value, then they are considered as zero in the score calculation. The calculated score is stored in that specific criteria column of the score table. After being scored in each criteria, now a table is generated which is score of each candidate (step 14).

The sample score table of the resumes of Table I are depicted below in Table IX.

TABLE. IX. SAMPLE SCORE TABLE

CV no.	Degree	Publication	CGPA	Skills	Experience	Major	Total
1	1.1	.2	4	1	2	2	10.3
2	1.1	0	2	2	5	2	12.1
3	5	.1	8	1	0	2	16.1
4	10.1	54.2	3	2	5	2	76.3
5	1.1	0	2	0	2	2	7.1
6	0	0	2	0	2	0	4
7	0	0	1	0	0	0	1

The first candidate had a B.Sc degree from Chittagong University of Engineering & Tech. And its ranking goes to institution category others. So the value for degree from the lookup Table VI is 1.1. She also has an international conference publication so the score is .2. The first candidate had the matching skill C++, experience of 3 years but as lecturer and major CSE. So the first candidate get scores according to the rules.

The scores of the other candidates will be calculated as the 1st candidate. The degree of 6th and 7th candidate doesn't match the required degree database as we have only condidered technical degrees and so the missing value is scored as zero. Accordingly, the 5th, 6th and 7th candidates doesn't match the skills requirement and so they get a zero in skills field. Now if we select any field as mandatory, the row containing zero in that field will be deleted.

2) Filter using skyline query: A skyline is defined as those points in a dataset those cannot be worse than any other point. A point dominates other points if it is as greater or equal in all criteria and greater in at least one criterion. A study in [24] states that during the past two decades, skyline queries are applied in several multi-criteria decision support problems. Skyline query utilizes the idea of skyline operator. There are several algorithms for the implementation of skyline operator like using directly in SQL queries, block nested loop, divide

and conquer (D&C), branch and bound, map reduce etc. We have used the block nested loop (BNL) and D&C method. Applying skyline queries on the score table according to employers' priorities, now the dominant applicants will be filtered. We can explain the working procedure of skyline query using Table IX. According to BNL, we compare all the data points with all other points. We keep the points that can dominate other points in all criteria and at least in one dimension. The points dominated are discarded from the list. Those points are considered to be skyline that dominates others or maybe a part of the skyline if they neither dominates nor dominated by others. For performing skyline filtering, the categories are to be selected by the recruiters. As the employer placed the requirements for only skills, major and experience category, so comparing the data points of these categories of Table IX we find that Candidate 2 and 4 dominates the other candidates in the required criteria as they contain either equal or higher value in every criteria than the other five. After applying skyline we get that candidate 2 & 4 best suits for the job and others are discarded from the list as depicted in Table X.

According to D&C, first we will divide the list into m partitions recursively. Then a local skyline is calculated for each partition. Then we merge these local skylines for calculating the global skyline. And finally these global skylines are the best candidates for the job.

3) Output generation: The system output will show the result of the potential candidates after the filtering process. The output will be sorted according to the score obtained and personal details like name, email, phone number of each candidate will be displayed. The sample output generation is shown in Table XI.

The algorithm is depicted below:

Algorithm 3: Filtering Using Skyline Query (BNL method)
Input: Generated *Score_list*, candidate (1), candidate (2),…. candidate(n)
Goal: To filter the total candidate, create the best candidates list and remove the non dominant candidates

1. **Begin**
2. Initialize an empty *best_candidates* list
3. Set flag 'passed'= true for all Candidates
4. **For** i to n of *score_list*
5. **If** (passed==false || candidate(i)== candidate(n))
6. Continue
7. Set *compare_list* = *score_table* filtered by (passed=true && candidate(i)
8. **For** each *job_criteria* do
9. **If** (candidate (i).[criteria] ≥candidate (i+1).[criteria])
10. Candidate (i+1).passed= false
11. **Else if** (candidate (i).[criteria] < candidate (i+1).[criteria])
12. { Candidate (i).passed= false
13. Break}
14. **End If**
15. **End for**
16. Set *best_candidates* list = candidates with (passed= true)
17. **End for**

Algorithm 3: Filtering Using Skyline Query (D&C method)
Input: Generated *Score_list*, candidate (1), candidate (2),…. candidate(n)
Goal: To filter the total candidate, create the best candidates list and remove the non dominant candidates

1. **Begin**
2. Initialize an empty *best_candidates* list
3. Keep the *Score_list* in N
4. Divide N using the median value of each criteria
5. Call BNL for calculating partial skyline of each partition
6. Merge the partial results
7. Call the BNL for calculating global skyline
8. Add the global skylines to *best_candidates* list
9. **End**

TABLE. X. SCORE TABLE AFTER FİLTERİNG USİNG SKYLİNE QUERY

CV no.	Skills	Experience	Major
2	2	5	2
4	2	5	2

TABLE. XI. OUTPUT GENERATİON

CV No.	Name	Phone	Email	Skills	Experience	Major
2	Mohammad Imtiaz Nur	01818772617	imti.nur@gmail.com	2	5	2
4	Faisal Karim	01826564578	faisal90@outlook.com	2	5	2

IV. IMPLEMENTATIONS AND EXPERIMENTS

In this section, we have described the implementation and experimental setup of our system with necessary illustrations.

A. Experimental Setup

Potential candidate selection system has been developed on a machine having Windows 10, 2.50GHz Core i5-3210 processor with 12GB RAM. The system has been developed in Python 3.7.3, Asp.Net Core and Angular5 in the front end and MS SQL Server is used in the back end for storing related data to complete this project.

B. Implementation

At the beginning of our system workflow, resume documents are fed into the system. All the resumes are stored in a file according to the specific job id. These resumes are then converted into text format using UTF-8 encoding and stored in a file named lookup.py.

Once we have found the extracted information table, it is stored in the document database.

On the other hand, employers set the necessary information for setting the requirements of each criteria. Job_info_details table holds the columns like Job_info ID, Keyword, Job Criteria Name i.e. the information set by the recruiters on the requirements setting step. For the specific job position, extracted information table can be uploaded next for score generation.

After scoring according to the rules set, the system generates the score table. This table can be downloaded by the

recruiter. Next the recruiter is given the option to choose the mandatory requirement criteria. If any of the criteria is chosen and candidates holding zero value in that specific criterion is removed before applying skyline query. Applying skyline query on the score table now returns the dominant applicants for the specified job by comparing each data points with each other. Then the unique dominant candidates holding maximum values in any of the criteria are returned.

The best candidates with score and personal details are shown in the output generation page (Fig. 4) in a descending score order.

C. Performance Evaluation

Potential candidate selection system performance is evaluated in two different phases- Information extraction performance, and filtering using skyline queries. We tested the performance of our system using 150 resumes of engineering background.

For the training of our NER model, we used a dataset of 300 manually annotated resumes and validated the model using resumes from the dataset. We found some incorrect values for extracted information and also some missing values. The precision, recall and F-measure of each block of information of the NER model is given below in Tablse XII, XIII and XIV. The extraction time is depicted in Table XV for different number of resumes.

We have tested the candidate filtering using skyline query with three different job criteria- Software Engineer with 2-4 years experience, Research Assistant with CGPA above 3.5 and 2 publications and Assistant Programmer with skills Java, JavaScript, HTML and CSS. We have scored the 150 resumes for these three different job positions. The returned output showed that the top scored candidates were different for the 3 job positions as the requirements were placed different. Comparing with the manual processing, we found that the filtering were accurate with a faster response. So based on the observations we can come to the conclusion that the accuracy of the skyline query depends on the accuracy of the scores generated. If the score generation is accurate, the skyline query returns those candidates that are best for the vacant position quite accurately and within few seconds.

We have also compared the response time of the skyline filtering using BNL and D&C method. The execution time for different number of resume data and with different number of dimensions is given in Table XVI. The table shows that the D&C method performs faster than BNL method because D&C method doesn't compare all the points naively as BNL does. The graphical comparison of each method for different criteria is shown in Fig. 5, 6 and 7.

TABLE. XII. PERSONAL INFORMATION EXTRACTION PERFORMANCE

Entity	Accuracy	Precision	Recall	F- measure
Name	99.77	0.99	0.99	0.99
Email	100.0	1.0	1.0	1.0
Phone	100.0	1.0	1.0	1.0
Date of	99.87	1.0	0.99	0.99

TABLE. XIII. EDUCATIONAL INFORMATION EXTRACTION PERFORMANCE

Entity	Accuracy	Precision	Recall	F- measure
University	99.87	1.0	0.99	0.99
Degree	99.25	0.99	0.99	0.98
Major	98.36	0.99	0.98	0.98
CGPA	100.0	1.0	1.0	1.0

TABLE. XIV. PUBLICATION, SKILLS AND CERTIFICATION INFORMATION EXTRACTION PERFORMANCE

Entity	**Accuracy**	**Precision**	**Recall**	**F- measure**
Publication	98.71	0.98	0.98	0.98
Skills	94.84	0.99	0.94	0.96
Certification	93.21	0.92	0.93	0.93

TABLE. XV. INFORMATION EXTRACTION TIME OF THE SYSTEM

No of CV	Extraction Time (msec)
10	11.08
20	20.53
50	28.5
100	60.64
150	83.81

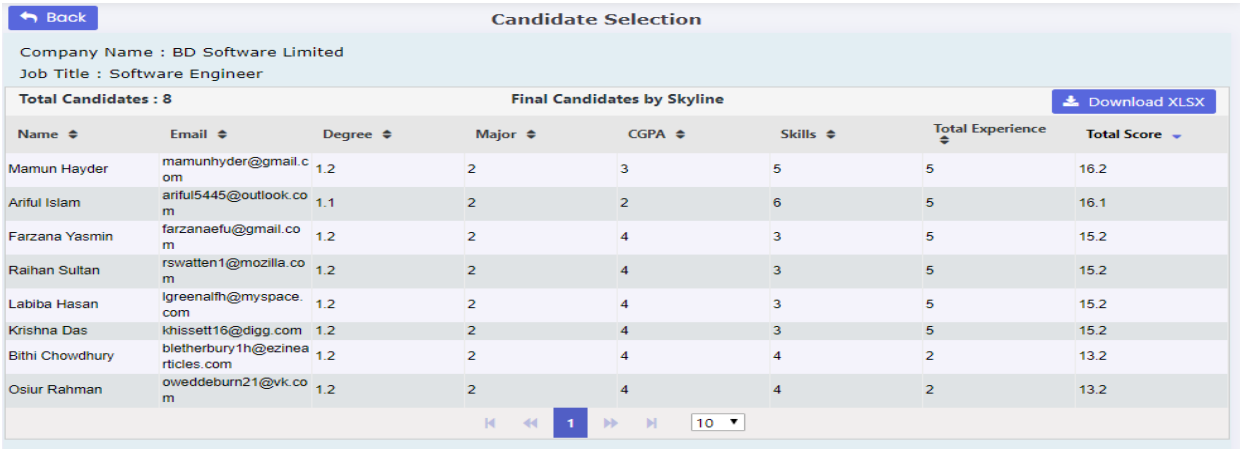


Fig. 4. Output Generation.

TABLE. XVI. RESPONSE TIME OF SKYLINE FILTERING

No of Resume	Response time (msec)					
	5 criteria		7 criteria		9 criteria	
	BNL	D&C	BNL	D&C	BNL	D&C
10	3.09	3.08	6.24	3.98	15.41	6.55
50	13.01	3.73	24.00	5.61	32.69	10.17
100	13.49	4.10	24.08	7.66	40.85	19.26
200	17.33	4.98	24.466	7.77	48.445	22.90

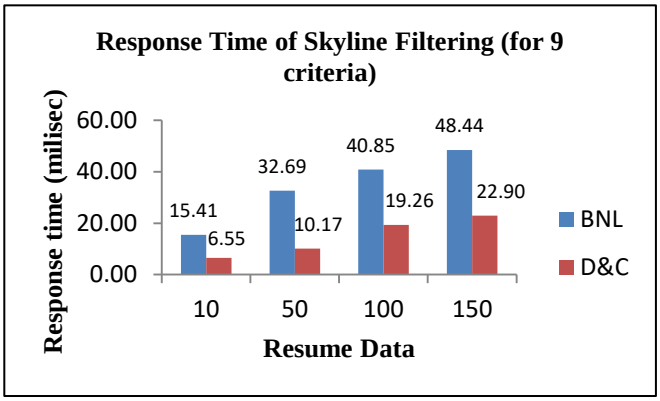


Fig. 5. Response Time of BNL & D and C for 9 Criteria.

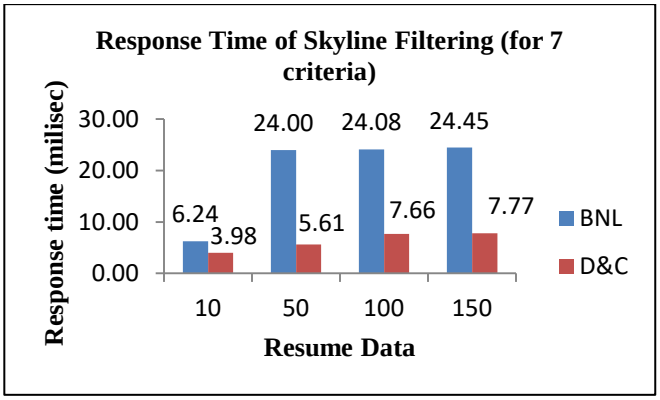


Fig. 6. Response Time of BNL and D and C for 7 Criteria.

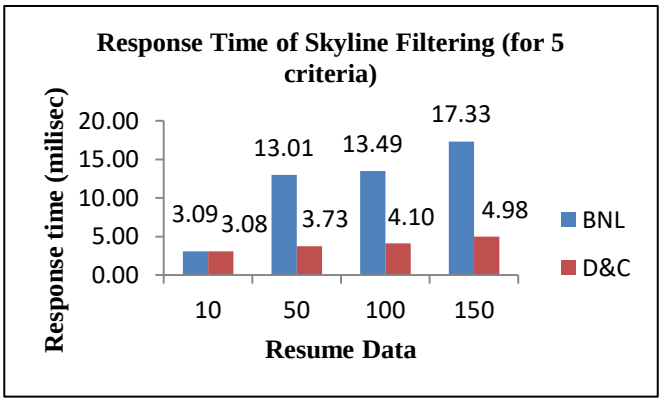


Fig. 7. Response Time of BNL and D and C for 5 Criteria.

V. CONCLUSION

In this paper, we have narrated the idea of a candidate selection system which finds the best potential candidates by extracting information and filtering using skyline query. Automating the total task may help the HR agencies by reducing time, cost and effort of searching and screening the pioneer applicants from vast applications. There are many automated candidate ranking system available online. But we have developed a novel idea of using skyline query in filtering and returning the dominant candidates for the job specified. Skyline queries are mostly applied in multidimensional decision application. In candidate filtering, the implementation of skyline is new and we have applied this novel approach in an efficient manner. In the system performance evaluation, we have used 150 resumes of technical background in testing of the system and found that, the system works in an efficient way of returning best candidates by matching the given requirements with qualifications of the candidates. Altogether the system performs better in filtering the documents as well as the candidates based on the information extracted from the resume documents. We have also compared the performance of two skyline filtering method and found out that D&C method returns faster response than BNL method. Our system works for only English documents currently. In future, we hope to extend it for Bangla resumes as it is fifth most spoken native language in the world by incorporating Bangla Language Processing and test the system performance accordingly.

REFERENCES

[1] Information Extraction, https://en.wikipedia.org/wiki/Information_extraction.

[2] Celik, D., “Towards a semantic-based information extraction system for matching resumes to job openings,” Turkish Journal of Electrical Engineering & Computer Sciences. vol. 24, pp. 141-159 (2016).

[3] Kumari, S., Giri, P., Choudhury, S., Patil, S.R., “Automated resume extraction and candidate selection system,” In: International Journal of Research in Engineering and Technology, e-ISSN. 2319-1163, p-ISSN. 2321-7308, vol. 03, issue. 01 (2014).

[4] Farkas, R., Dobó, A., Kurai, Z., Miklós, I., Nagy, Á., Vincze, V., Zsibrita, J., “Information extraction from hungarian, english and german cvs for a career portal,” In: Prasath R., O’Reilly P., Kathirvalavakumar T. (eds) Mining Intelligence and Knowledge Exploration. Lecture Notes in Computer Science, vol. 8891, Springer, Cham (2014).

[5] K. Yu, G. Guan, M. Zhou, “Resume information extraction with cascaded hybrid model,” In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, pp. 499–506, Ann Arbor, June 2005.

[6] Information Extraction from CV, https://medium.com/@divalicious.priya/information-extraction-from-cv-acec216c3f48.

[7] Writing Your Own Resume Parser, https://www.omkarpathak.in/2018/12/18/writing-your-own-resume-parser/.

[8] Resume Parser, https://github.com/bjherger/ResumeParser.

[9] Chen, J., Zhang, C., Niu, Z., “A two-step resume information extraction algorithm,” Mathematical Problems in Engineering, vol. 2018, Article ID 5761287 (2018).

[10] Shah, S., Thakkar, A., Rami, S., “A survey paper on skyline query using recommendation system,” In: International Journal of Data Mining And Emerging Technologies, vol. 6, issue. 1, pp. 1-6, ISSN. 2249-3212 (2016).

[11] Kalyvas, C., Tzouramanis, T., “A survey of skyline query processing,” 2017.

[12] Papadias, D., Tao, Y., Fu, G., Seeger, B., "An optimal and progressive algorithm for skyline queries," In: ACM SIGMOD International Conference on Management of Data, pp. 467-478 (2003).

[13] Patil, S., Palshikar, G.K., Srivastava, R., Das, I., "Learning to rank resumes," In: FIRE, ISI Kolkata, India (2012).

[14] Yi, X., Allan, J., Croft, W.B., "Matching resumes and jobs based on relevance models" In: SIGIR, Amsterdam, The Netherlands, pp. 809–810 (2007).

[15] Rode, H., Colen, R., Zavrel, J., "Semantic CV search using vacancies as queries," In: 12th Dutch-Belgian Information Retrieval Workshop, Ghent, Belgium, pp. 87–88 (2012).

[16] Bollinger, J., Hardtke, D., Martin, B., "Using social data for resume job matching," In: DUBMMSM, Maui, Hawaii, pp. 27–30 (2012).

[17] Faliagka, E., Ramantas, K., Tsakalidis, A., Tzimas, G., "Application of machine learning algorithms to an online recruitment system," In: Seventh International Conference on Internet and Web Applications and Services, Stuttgart, Germany, pp. 215–220 (2012).

[18] Dandwani, V., Wadhwani, V., Chawla, R., Sachdev, N., Arthi, C.I., "Candidate ranking and evaluation system based on digital footprints," In: IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN. 2278-0661, p-ISSN. 2278-8727, vol. 19, issue. 1, ver. 4, pp. 35-38 (2017).

[19] Faliagka, E., Ramantas, K., Tsakalidis, A., Viennas, M., "An ıntegrated e-recruıtment system for cv rankıng based on ahp," In: 7th International Conference on Web Information Systems and Technologies, Noordwijkerhout, The Netherlands, (2011).

[20] SpaCy, https://spacy.io/.

[21] UTF-8 encoding, https://www.fileformat.info/info/unicode/utf8.htm.

[22] Named Entity Recognition, https://en.wikipedia.org/wiki/Named-entity_recognition.

[23] spaCy NER training model, https://course.spacy.io/chapter4.

[24] Tiakas, E., Papadopoulos, A. N., Manolopoulos, Y., "Skyline queries: An introduction," In: 6th International Conference on Information, Intelligence, Systems and Applications (IISA), DOI: 10.1109/IISA.2015.7388053, E-ISBN: 978-1-4673-9311-9, July (2015).