Potential Candidate selection using Information Extraction and Skyline Queries

Farzana Yasmin, Mohammad Imtiaz Nur and
Mohammad Shamsul Arefin

October 16, 2019

Potential Candidate selection using Information Extraction and Skyline Queries

Abstract. Information extraction is a mechanism for devising an automatic method for text management. In the case of candidate recruitment, nowadays different companies ask the applicants to submit their applications or resumes in the form of electronic documents. In general, there are huge numbers of resumes dropped and therefore the volume of the documents increases. Extracting information and choosing the best candidates from all these documents manually are very difficult and time consuming. In order to make the recruitment process easier for the companies, we have developed a framework that takes the resumes of candidates as well as the priorities of the employer as input, extract information of the candidates using Natural Language Processing (NLP) from the resumes, rank the candidates according to predefined rules and return the list of dominant candidates using skyline filtering.

Keywords: Information Extraction, natural language processing, skyline query, candidate selection.

1 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. 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 of the applicants and process these information in an automated way, 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 ranking 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, named entity recognizer (NER) and utilizes skyline query processing for candidate scoring and ranking 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.

2 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 GRPS 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].

A two step resume information extraction algorithm is developed in [9]. 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 which are then searched in a database of Dutch CVs. The best-ranked candidates gets 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 ranking, the use of skyline query in this sceneraio is relatively new approach and we have implemented this novel approach in our framework.

3 System Architecture and Design

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

3.1 Processing Module

Document Input. First we will need to input the resumes in the interface for a specific job id. After documents are being fed to the system in processing module, information extraction process begins and we used a NLP module named spaCy [20] for the rest of the processing steps. Suppose, we have fed the following resumes in the system:

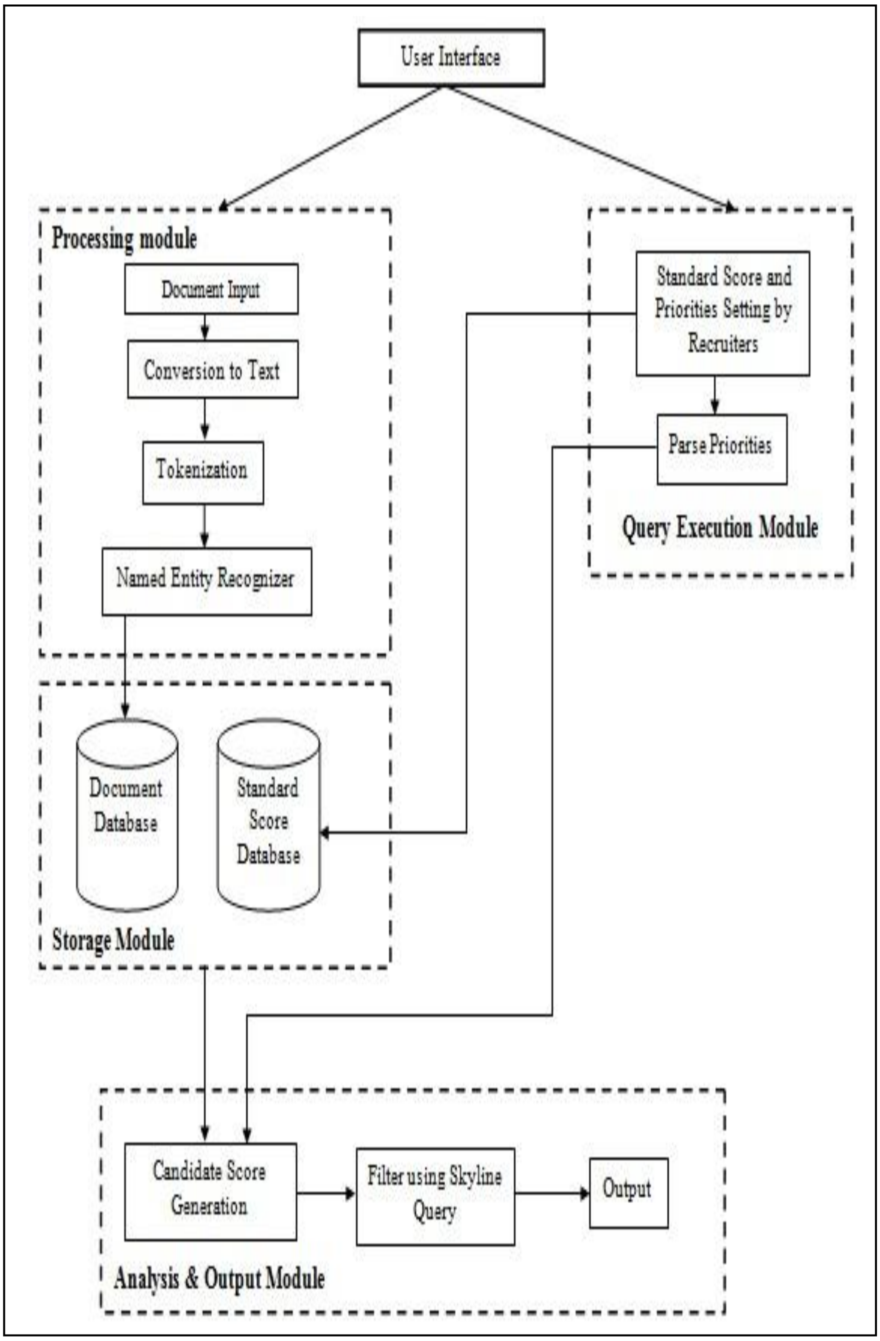


Fig. 1. System Architecture of Potential Candidate Selection

Adam Wang (Male)

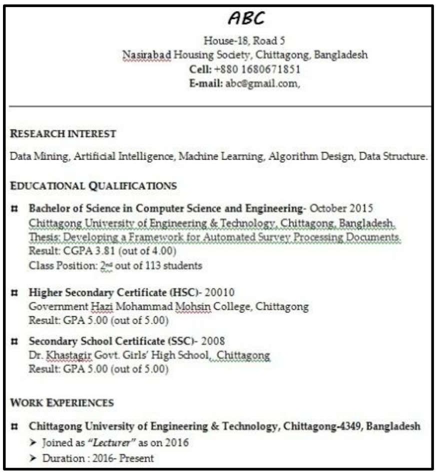
XXXX Company of Bejing,
Beijing City, 100007
1364-110-XXX
wangXXX@hotmail.com

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

House-18, Road 5
Nasirabad Housing Society, Chittagong, Bangladesh
Cell: +880 1680671851
E-mail: abc@gmail.com,

RESEARCH INTEREST

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

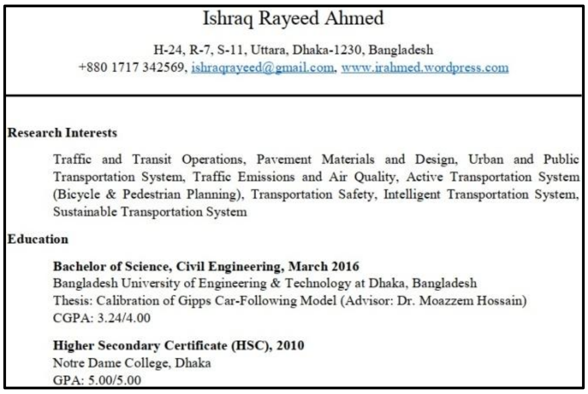
EDUCATIONAL QUALIFICATIONS

- **Bachelor of Science in Computer Science and Engineering**- October 2015
  Chittagong University of Engineering & Technology, Chittagong, Bangladesh.
  Thesis: Developing a Framework for Automated Survey Processing Documents.
  Result: CGPA 3.81 (out of 4.00)
  Class Position: 2nd out of 113 students
- **Higher Secondary Certificate (HSC)**- 20010
  Government Hazi Mohammad Mohsin College, Chittagong
  Result: GPA 5.00 (out of 5.00)
- **Secondary School Certificate (SSC)**- 2008
  Dr. Khastagir Govt. Girls' High School, Chittagong
  Result: GPA 5.00 (out of 5.00)

WORK EXPERIENCES

- **Chittagong University of Engineering & Technology, Chittagong-4349, Bangladesh**
  - Joined as *"Lecturer"* as on 2016
  - Duration : 2016- Present

(b)



Ishraq Rayeed Ahmed

H-24, R-7, S-11, Uttara, Dhaka-1230, Bangladesh
+880 1717 342569, ishraqrayeed@gmail.com, www.irahmed.wordpress.com

Research Interests

Traffic and Transit Operations, Pavement Materials and Design, Urban and Public Transportation System, Traffic Emissions and Air Quality, Active Transportation System (Bicycle & Pedestrian Planning), Transportation Safety, Intelligent Transportation System, Sustainable Transportation System

Education

Bachelor of Science, Civil Engineering, March 2016
Bangladesh University of Engineering & Technology at Dhaka, Bangladesh
Thesis: Calibration of Gipps Car-Following Model (Advisor: Dr. Moazzem Hossain)
CGPA: 3.24/4.00

Higher Secondary Certificate (HSC), 2010
Notre Dame College, Dhaka
GPA: 5.00/5.00

(c)

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

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. UTF-8 is a compromise character encoding that can be as compact as ASCII (if the file is just plain English text) but can also contain any Unicode characters (with some increase in file size). UTF stands for Unicode Transformation Format. The '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 [21].

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.

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]. We are considering 12 criteria for information extraction- university, degree, major, result, experience, publication, skill/others, training/certification and personal information (name, date of birth, email, phone etc.).

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.

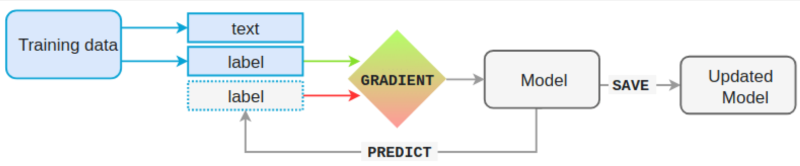


Fig. 3. spaCy's NER model training process (Source: [23])

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).

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

Algorithm 3.1: 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

After the validation of the training of the NER model, now we use this model to extract the values of the entities trained from the resumes. The recognized entity values are stored in a row of a table for each candidate in the storage module. If we send the sample resumes of Fig. 2. in the NER model the table of the extracted information take the form like below:

1	Name	Email	Phone	Date of Birt	University	Degree	Major	CGPA	Skills
2	ABC	abc@gmail.com	1680671851	30-09-1993	Chittagong University of Engineering and Technology	BSc	Computer Science & Engineering	CGPA: 3.81	C++, C#, Python, Htn
3	Adam Wang	wangXXX@hotmail.com	1364110xxx	7-Sep-92	School of XXX	Bachelor	Computer Science & Engineering	_	Image Processsing
4	Ishraq Rayeed Ahmed	ishraqrayeed@gmail.com,	880 1717342569	14-12-1989	Bangladesh University of Engineering and Technology	Bsc	Civil Engineering	CGPA: 3.24	MATLAB, R Project, /

(a)

I	J	K	L	M
Skills	Total Experience	Publications	Certifications	
C++, C#, Python, Html, Javascript	Total Year of Experience : 3.7 Year(s)	Computer Science and Engineering Research Journal (CSERJ), International Conference	__	
Image Processsing	Total Year of Experience : 4 Year(s)	__	__	
MATLAB, R Project, ArcGIS, VisSim, EPAnet,Auto(	__	International Conference, International Conference,	CCNA	

(b)

Fig. 4. (a), (b) Expected extracted information

3.2 Query Execution Module

Standard Scores and priorities setting for each criteria by Recruiter. In the UI, employers set the standard scores required to evaluate the abilities of the candidate according to their job criteria. Each criterion gets a value and a weight for a specific keyword. The weight represents the relative importance or prioritiesof the specific criteria and value represents the variations of the score of each criteria. Keyword gives the matching criteria i.e. which information to be satisfied for scoring. These standard scores are stored in the storage module as a lookup table. For example, for software developer position, the employer sets the following values and weights in the table for each criteria.

Table 1. Standard Score Setting Table

Job_criteria	Keywords	Value	Weight
Skills	C++	10	5
Skills	Java	10	5
Skills	PHP	8	5
Experience	3	5	3
Experience	0	2	3
Major	CSE	10	2
Major	EEE	6	2

Parse the Requirements. The system will then parse these requirements of the employer in the query execution module.

3.3 Storage Module

Storage module stores information processed by the processing and query execution module. The extracted information table after the entites are recognized are stored in the document database. The standard scores set by the recruiters in the query execution phase are stored in the score database. The total storage is required for the candidate score generation in the analysis and output generation phase.

3.4 Analysis and Output Module

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 by the employer for different categories.

The algorithm of candidate score generation is given below:

Algorithm 3.2: 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] += job_details (value)*job_details(weight)*
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 by multiplying the value and weight set in the standard score table (step 10).

If multiple keywords are matched for a specific critria, 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.

The score calculation follows the following formula (1):

$$Score[job\_criteria] = Score[job\_criteria] + (job\_details\ (value) * job\_details\ (weight)) \quad (1)$$

For the result column scoring, extracted result of the applicant matched with the sorted list of previously set result keywords. İf the extracted result is greater or equal to any specified keyword of the result, the score is calculated according to that result keyword. The same goes for the total years of experience column.

For the publication column, internation conference, international journal keywords are searched and matched. İf found, the number of occurences are counted.

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 for the resumes in Fig. 2 are depicted below:

Table 2. Sample Score Table

CV no.	Skills	Experience	Major	Total
1	50	15	20	85
2	0	15	20	35
3	50	6	0	56

The first candidate had the matching skill C++, experience of 3.7 years and major CSE. So the first candidate fulfills all the requirements of the specified job position and get scores according to the rules set as Table 1 i.e. Scores[skill]= value for C++ (10) * weight of C++ (5) = 50. The skills of 2nd candidate doesn't match the required skills and so the missing value is scored as zero. Accordingly, the 3rd candidate's major doesn't match the requirement and so he gets a zero in major field. Now if we select the Major field as mandatory, the row containing zero in this field i.e candidate 3 will be deleted.

Filter using Skyline Query. A skyline is defined as those points in a dataset those are not dominated by any other point. A point dominates other points if it is as good or better in all dimensions and better in at least one dimension. A study in [24] states that during the past two decades, skyline queries are applied in several multi-criteria decision support problems. Given a dominance relationship in a dataset, a skyline query returns the objects that cannot be dominated by any other objects. Skyline query utilizes the idea of skyline operator. There are several algorithms for the implementation of skyline operator like using directly in SQL queries, divide and conquer, branch and bound, map reduce etc. We have used the combination of SQL query and the map reduce method. Applying skyline queries on the score table according to employers' priorities, now the dominant applicants will be filtered. The algorithm is depicted below:

Algorithm 3.3: Filtering Using Skyline Query
Input: Generated *Score_table*
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. **For** each *job_criteria* **do**
4. Find the max value from all the candidates by mapping according to *job_criteria*
5. Filter all the candidates who have the max values in the *job_criteria*
6. Concatenate the candidates in the *best_candidates* list
7. **End for**
8. Remove the duplicate candidates from the *best_candidates* list

We can explain the working procedure of skyline query using Table 3. At first we find the max value for each job criteria (step 4). For example, from Table 3., skills column has the max value 50, experience column has the max value 15 and major column- 20. We map these max values in another list accouding to job criteria at the same time (step-4). Now we filter the candidates holding any of these max values (step 5) because these are the dominant objects as per the skyline filtering and are pushed in the best_candidates list (step 6) i.e. candidate 1 & 2. Then we remove the duplicate candidates from the best_candidates list and make the list unique (step 8). As candidate 1 holds max value in all the 3 criteria, it is pushed 3 times in the list. So to make the list unique we remove the duplicate values of the candidates and just take the row 1 time.

Table 3. Score table after filtering using skyline query

CV no.	Skills	Experience	Major	Total
1	50	15	20	85
2	0	15	20	35

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 Fig. 5.

CV no.	Name	Email	Phone	Skills	Experience	Major	Total
1	ABC	abc@gmail.com	1680671851	50	15	20	85
2	Adam Wang	wangXXX@hotmail.com	1364110xxx	0	15	20	35

Fig. 5. Output generation

4 Implementations and Experiments

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

4.1 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.

4.2 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 (Fig. 6).

```
Imtiaz_Nur_T.txt - Notepad
File  Edit  Format  View  Help
b' \n\nImtiaz Nur \n\n     Date of Birth: 07/09/1993 \n\nNasirabad Housing Society, Chittagong-4209. \n\nCell: +880172
of Engineering & Technology (CUET) \nCSE          \nCGPA 3.11(out of 4.00) \n\nYear: 2015 \n\n:  \n: \n:  \n\n \n\n \n
ertifications  \n\n \n\n  \xe2\x80\x9cAdvance IT Training on Java\xe2\x80\x9d conducted by ICT ministry of Bangladesh
nalysis \n\nreport as a project in Software Engineering course \n\nTools   \n\n: \n\nPHP, HTML, CSS, JS, MySQL, Bootst
Performed Industrial Attachment at W3Eden, Inc. for 2 weeks on January, 2016. \n  Performed System Analysis at Broadca
```

Fig. 6. Snapshot of Resume in Text format

The text files are then called for tokenization and named entity recognition. Next, calling the trained model of NER, we extract the information from the tokenized data of the resumes. We have extracted information of 12 entities- university, degree, major, experience, publication, skill, certification and personal information (name, date of birth, email, phone etc.). The information of these entities are extracted according to the annotation of the trained NER model (Fig. 7).

	A	B	C	D	E	F	G	H	I	J	K	L
1	Name	Email	Phone	Date of Birth	University	Degree	Major	CGPA	Skills	Total Experience	Publications	Certifications
2	Farzana Yasmin	farzanae	168067	31-12-1993	Chittagong Univer	BSc	CSE	CGPA: 3.81	C++, C#, Python, Htn	Total Year of Experience	International Journal, I	__
3	Imtiaz Nur	imti.nur(	880172		/n/n	B.Sc.	__	CGPA: 3.11	@@	Total Year of Experience	__	__
4	Ohidul Islam	ohid@gr	18189	14-12-1989	Bangladesh Unive	Bsc	Computer Science {	__	Python, Java,___, J	__	__	__
5	Faisal Karim	faisal90(	151578	__	International Islar	/n	Computer Science	GPA: 3.78	C#, C++, PHP, Html, (	Total Experience: 1 year	National Conference	
6	Md. Intishar Nur	intishar7	167192	13.05.1994	IUT	Bsc	Mechanical Enginee	CGPA: 3.34	Matlab, C++	__	International Conference, National C	
7	Ananna Das	anannad	177887	6/1/1983	__	__	Eng	CGPA: 2.81	C++, C#, Python, Htn	Total Year of Experience	18. ACM TRANSACTION	__
8	Hasibul Haq	h.haq602	16172		Chittagong Univer	BSc	__	CGPA: 3.21	__	Experience : 2.5 Year(s)	__	__
9	Farid Alam	farid@gr	152287	23-5-87	__	BSC	CS Eng.	3.54	//t/n	__	__	CCNA
10	Masud Habib Sawon	mhabib(	182689	__	International Islar	Bsc	Computer Science	CGPA: 3.43	__	Total Experience: 4.5 ye	National Conference	
11	Shoumik Barua	shoumik	17379	__	Chittagong Univer	Bachelor of Sciences	Computer Science {	CGPA: 2.98	Matlab, C++	2 years	__	
12	Kamruz Zaman	kzaman6	17476	3/4/1990	__	Bscee	__	CGPA: 2.99	AutoCad, Matlab	Total Year of Experience	ACM TRANSACTIONS O	__
13	Saber Hossain	saber354	191287		Chittagong Univer	__	Computer Science {	CGPA: 3.65	As@_, Bootstrap, H	2.5 Year(s)	__	__
14	Touhidul Islam	t.islam9	181899	11.04.1989	Bangladesh Unive	__	EEE	CGPA: 3.74	Python, Java, C++, P	__	__	CCNA
15	Atanu Dey	atanu07(	151578	__	__	Bsc	Computer Science/	CGPA: 3.53	C#, C++, PHP, Html, (	Total Experience: 3 year	National Conference	
16	Shimul Das	shimuld	16700	6.9.1993	Islamic University	B.sc. in	Mechanical Enginee	__	Matlab, C++	__	International Conference, National C	
17	Nusrat Jahan	nusratjal	178000	15.04.87	Chittagong Univer	BSc	__	CGPA: 3.61	__	Total Year of Experience	Computer Science and	__
18	Fariba Hossen	fariba.h9	16172		Chittagong Univer	__	Engineering	CGPA: 3.53	__C#, C++, JavaScri	Total Year of Experience	__	__
19	Saniul Alam	sani455(	15226	27-07-1996	__	Bsc	ETE	: 2.66	Python, Java, C++, P	__	__	CCNA
20	Mahmudul Hasan	mahmud	1826	__	International Islar	Bsc	Computer Science	Result 3.55	//nJQuery	Total Experience: 2 year	National Conference	
21	Abrar Shahriar	abrarcse	1737	__	C.U.E/n	Bachelor of Sciences	__	CGPA: 2.98	Matlab, C++	2 years	__	
22	Mesbah Uddin	mesbahu	168798	17-12-1964	CUET	BSc	Electrical Engineeri	__	/t/tC++, C#, Python,	Total Year of Experience	SIGNAL PROCESSING- ,	__
23	Bashir Mahmood	mahmoo	880156	__	Chittagong Univer	BSc	CS	CGPA: 2.85	__	Total Year : 2.5 Year(s)	__	__

Fig. 7. Snapshot of Extracted Information Table

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 and scores of each criteria. Job_info_details table holds the columns like Job_info ID, Keyword, Value, Weight, Job Criteria Name i.e. the information set by the recruiters on the score setting step (Fig. 8).

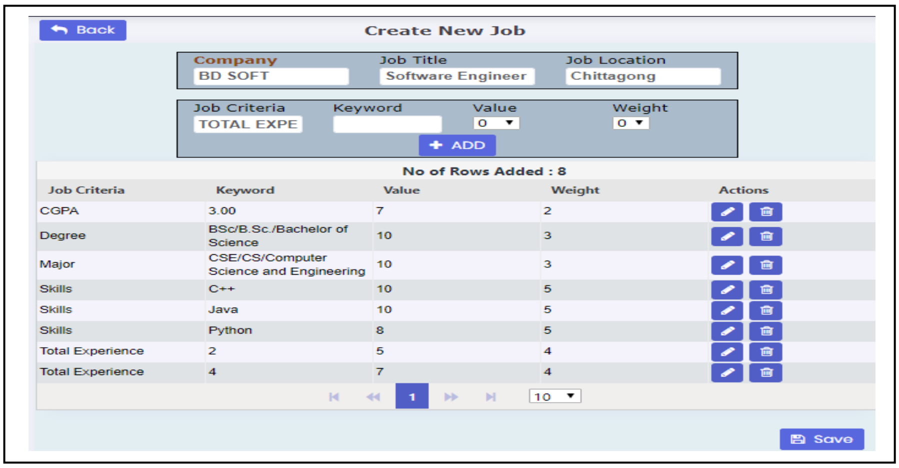


Fig. 8. Snapshot of Requirement Setting by Recruiters

For the specific job position, extracted information table can be uploaded next for score generation (Fig. 9).

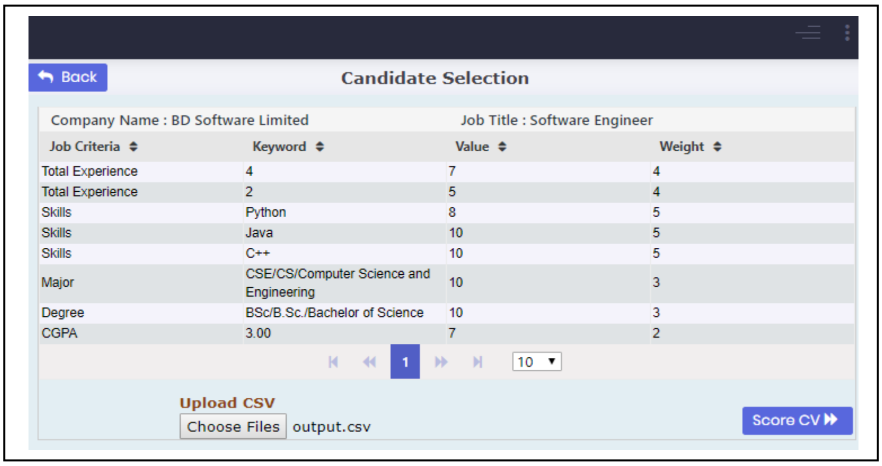
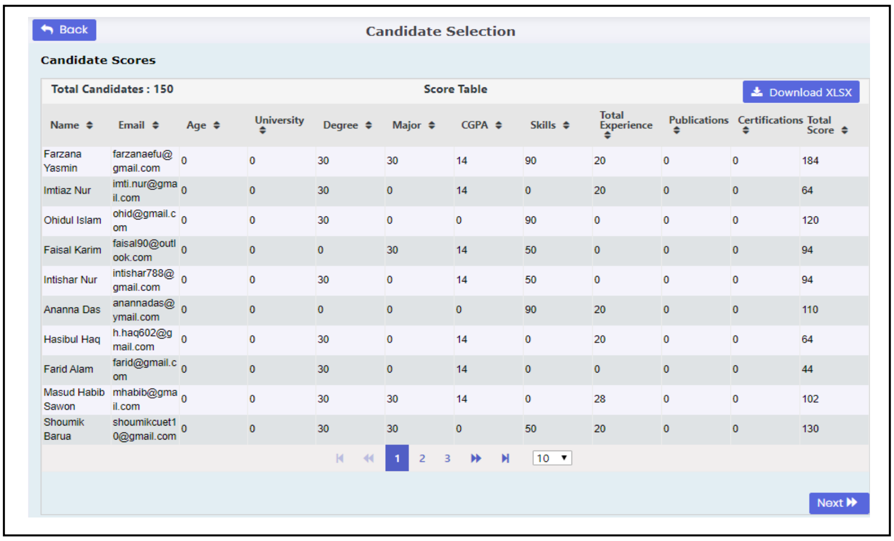


Fig. 9. Snapshot of Extracted Information File Upload

After scoring according to the rules set, the system generates the score table. This table can be downloaded by the recruiter (Fig. 10).



Candidate Selection

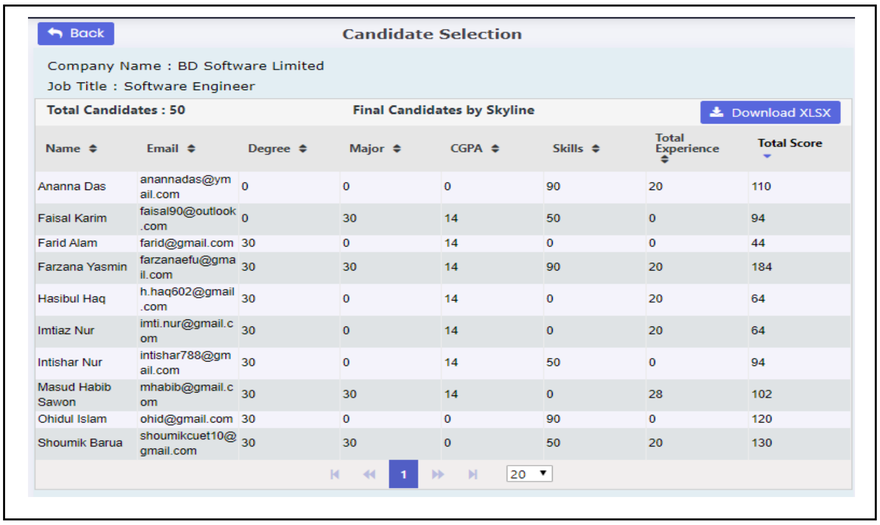
Candidate Scores

Total Candidates : 150 Score Table

Name	Email	Age	University	Degree	Major	CGPA	Skills	Total Experience	Publications	Certifications	Total Score
Farzana Yasmin	farzanaefu@gmail.com	0	0	30	30	14	90	20	0	0	184
Imtiaz Nur	imti.nur@gmail.com	0	0	30	0	14	0	20	0	0	64
Ohidul Islam	ohid@gmail.com	0	0	30	0	0	90	0	0	0	120
Faisal Karim	faisal90@outlook.com	0	0	0	30	14	50	0	0	0	94
Intishar Nur	intishar788@gmail.com	0	0	30	0	14	50	0	0	0	94
Ananna Das	anannadas@ymail.com	0	0	0	0	0	90	20	0	0	110
Hasibul Haq	h.haq602@gmail.com	0	0	30	0	14	0	20	0	0	64
Farid Alam	farid@gmail.com	0	0	30	0	14	0	0	0	0	44
Masud Habib Sawon	mhabib@gmail.com	0	0	30	30	14	0	28	0	0	102
Shoumik Barua	shoumikcuet10@gmail.com	0	0	30	30	0	50	20	0	0	130

Fig. 10. Snapshot of Score Table

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.



Candidate Selection

Company Name : BD Software Limited
Job Title : Software Engineer

Total Candidates : 50 Final Candidates by Skyline

Name	Email	Degree	Major	CGPA	Skills	Total Experience	Total Score
Ananna Das	anannadas@ymail.com	0	0	0	90	20	110
Faisal Karim	faisal90@outlook.com	0	30	14	50	0	94
Farid Alam	farid@gmail.com	30	0	14	0	0	44
Farzana Yasmin	farzanaefu@gmail.com	30	30	14	90	20	184
Hasibul Haq	h.haq602@gmail.com	30	0	14	0	20	64
Imtiaz Nur	imti.nur@gmail.com	30	0	14	0	20	64
Intishar Nur	intishar788@gmail.com	30	0	14	50	0	94
Masud Habib Sawon	mhabib@gmail.com	30	30	14	0	28	102
Ohidul Islam	ohid@gmail.com	30	0	0	90	0	120
Shoumik Barua	shoumikcuet10@gmail.com	30	30	0	50	20	130

Fig. 11. Snapshot of Output Generation

Applying skyline query on the score table now returns the dominant applicants for the specified job by finding the max value and mapping them according to the job criteria. Then the unique 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. 11) in a descending score order.

4.3 Performance Evaluation

We tested the performance of our system using 150 resumes. For the training of our NER model, we used a dataset of 350 annotated resumes and validated the model using 50 resumes from the dataset. Extraction procedure is the toughest task of the whole system. We found some incorrect values for extracted information and also some missing values. The prcision, recall and f-measure of each entity of the NER model is given below:

Table 4. Accuracy, Precision, Recall and F-measure of the Entities Recognized

	Name	Email	Phone	Date of Birth	University
Accuracy (%)	99.76525821596243	100.0	100.0	99.87452948557089	99.87452948557089
Precision	0.9984350547730829	1.0	1.0	1.0	1.0
Recall	0.9976525821596244	1.0	1.0	0.998745294855709	0.998745294855709
F-measure	0.9978859382188446	1.0	1.0	0.9993722536095418	0.9993722536095418

	Degree	Major	Publication	Skills	CGPA
Accuracy (%)	99.24717691342535	98.35680751173709	98.70892018779342	94.83568075117371	100.0
Precision	0.9925285145930405	0.9963484611371936	0.9872584733670198	0.9904364458355068	1.0
Recall	0.9924717691342535	0.9835680751173709	0.9870892018779343	0.9483568075117371	1.0

F-measure	0.98966292 25927619	0.98872802990 97218	0.9852126262 984936	0.9654163803 961983	1.0

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 would be returned by manual filtering.

We have tested the filtering and ranking using skyline query with 3 different job criteria- Software Developer 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 3 different criteria. 3 criteria returned different combinations of candidates as the requirements are different with a 100% accuracy.

We have also tested the skyline filtering with 50,000 synthesized score data. The execution time for different number of data is given in Table 5.

Table 5. Response Time of Skyline Filtering

No. of Data	Response Time (mili sec)
3000	5.299999960698187
6000	9.265000000596046
25000	27.17999997548759
50000	81.80499996524304

The table shows that the skyline query can perform in a very responsive way.

5 Conclusion

In this paper, we have presented a candidate ranking system that 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 ranking and returning the dominant candidates for the job specified. Skyline queries are mostly applied in multidimensional decision application. In candidate ranking, 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 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. The performance of the extraction can be made higher by increasing the training data. Altogether the system performs better in reducing the processing time as skyline query returns the dominant applicants in a very responsive way.

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 Integrated 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)