

PROJECT REPORT

Information Retrieval and Extraction, Spring 2016

Project-11

Semantic Job Candidate Recommendation Engine

Sachin Baldua : 201330082
Sikander Sharda : 201301159
Nausheen Fatma : 201407541

1. Problem Statement

A recruiter wants to select the best CVs which suit his job requirement. To do this, one need to semantically match the words mentioned by candidate in the resume to those mentioned in the job description and get the best candidates suitable for the job. We need to design a search engine which takes the requirements of the job such as the skills and the position as the input and outputs a ranked list of CV in order of their relevance to the opening.

2. Applications

A particular job vacancy may receive huge number of applications. Manually sorting the CVs is practically impossible. Sorting out the suitable CVs from thousands (or possibly lakhs) of applications is a very challenging problem. We aim to build a system in which given a job title/description (query), the system can retrieve the suitable CVs (documents) and rank them according to some relevance measure .

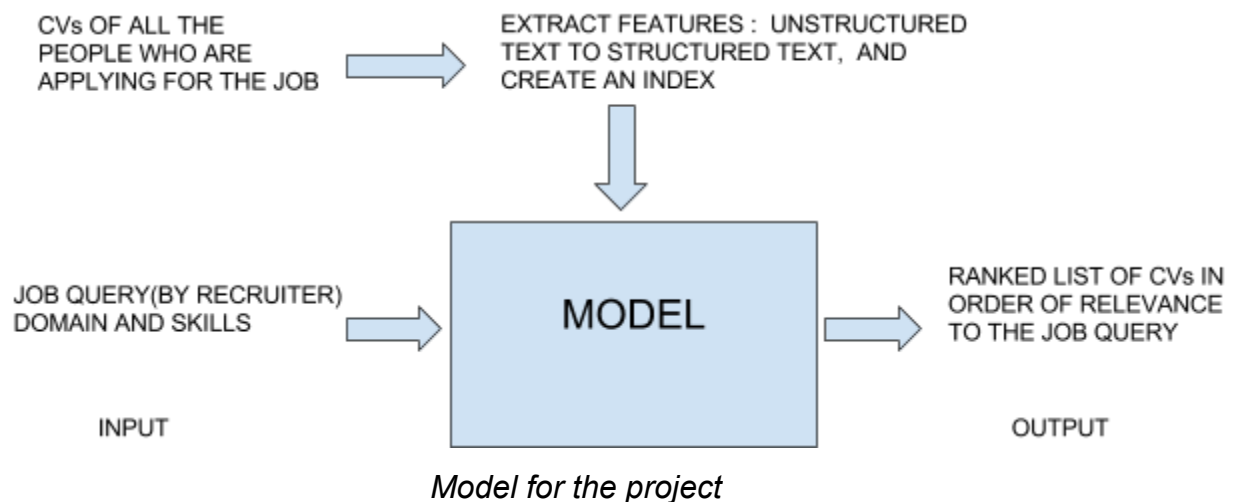
This can be applied in various job portals/websites such as LinkedIn, monster.com, theladder.com, etc. to select CVs for a particular job.

Similar recommendation systems has widely been used in various applications such as music, movie (Youtube), product (Flipkart, Amazon), computational advertisements, news articles recommendation systems, where a new item is recommended by modelling the behaviour of past history of user selections.

3. Challenges

Following are the major challenges we faced in this project:

- Feature identification and extraction from varying CV formats and styles such as tabular data or basic text from both pdfs and markup language format.
- Extracting data related to the person and his skills by suitable regex and other similar techniques.
- Identifying CVs of different domains of job offerings such as android,web development, networking,etc.
- Extracting different forms of technologies like frameworks,languages etc and the corresponding job domains they are primarily used for. This also includes including all naming techniques of these technologies. For example C++ can also be written as cpp or different modules can refer to the same or different languages/domains such as nltk or pygame which refer to different job domains.
- Different ranking algorithms which can be used to rank the shortlisted CVs corresponding to a job requirement.



4.Procedure

- **Extracting the text :**

As most of the CV are in pdf formats, this becomes a major problem. Many modules do not convert the tables in pdf format properly into text files. This results in complication related to data of the individual. The most suitable module

to perform this task is Java's pdfbox which can convert table rows into simple lines in text format.

- **Applying regex to retrieve data from the text files:**

Using regex and simple conventional techniques, we can find out the name, college, highest degree, phone number, email-id , etc of the individual. This knowledge helps us to recognise the individual. We also built a User-Interface in which a resume in pdf format was uploaded and it gave output as the information of the individual. This part was performed in the second deliverable.

- **Extracting list of languages, career domains, modules etc :**

This helps us later to extract the terms using bag of words concepts from the text files of the CVs and can later be used to match them to the most suitable CV according to the job requirement given by the user.

- **Applying tf-idf method on the parsed CVs:**

After extracting all the (personal information and) skills of the individual related to languages,frameworks,jobs, etc we need to apply tf-idf score to each of the terms using Scikit Library thus providing us with a ranked list of all the CVs corresponding to a job requirement query. The functionality is implemented on the terminal itself.This is implemented in the third deliverable.

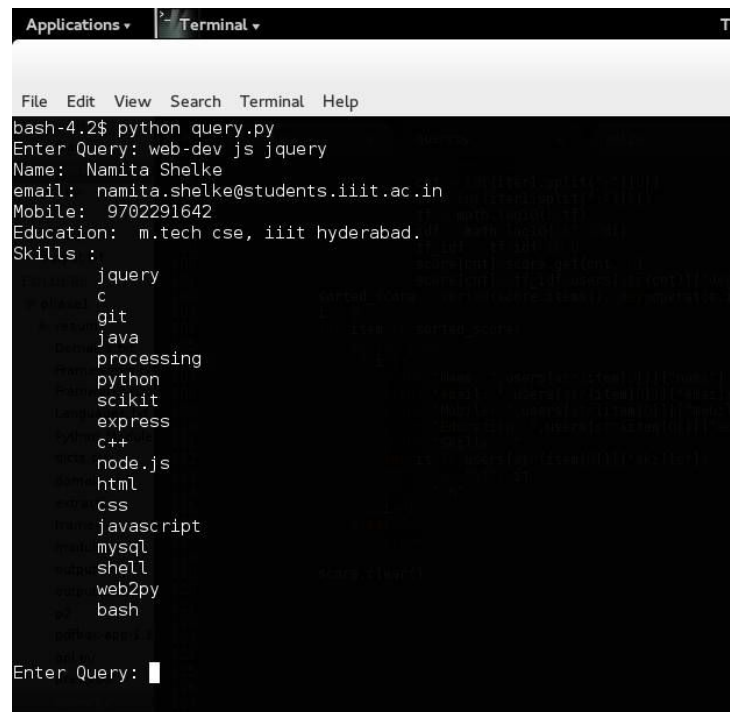


The screenshot shows a web browser window with the address bar displaying 'localhost/ire/fileupload.php'. The page contains a form with the following fields and values:

- Name: soumyajit ganguly
- Mobile: +919830353468
- E-mail: soumyajit.ganguly@gmail.com
- Education: currently pursuing master of science (research) in computer science at
- Experience: computer vision engineer, tech bla solutions pvt. ltd, kolkata, india

A 'Submit' button is located at the bottom of the form.

The output of the UI for a CV for second deliverable



```
Applications ▾ Terminal ▾ Th
File Edit View Search Terminal Help
bash-4.2$ python query.py
Enter Query: web-dev js jquery
Name: Namita Shelke
email: namita.shelke@students.iiit.ac.in
Mobile: 9702291642
Education: m.tech cse, iiit hyderabad.
Skills :
    jquery
    c
    git
    java
    processing
    python
    scikit
    express
    c++
    node.js
    html
    css
    javascript
    mysql
    shell
    web2py
    bash
Enter Query: █
```

The output on terminal for the third deliverable

5.Modules Used

The modules used in the code are Math, operator, sys, re, os and json.

6.Analysis ,Experiments & Results

The major analysis experiments and their different types of results are listed as follows:

- To find a suitable module to convert CV which are mostly in pdf formats to text files. Especially extracting a table in proper form. We used many different python and java modules like pdfMiner, pyPdf, etc before arriving at pdfbox. In the previous modules, extracting tabular data was a major problem as they came in different lines(according to cells) whereas in java's module pdfbox a single row gave text in a single line.
- To extract data properly we have to set a suitable regex which is not too strict so that it may not take some data or too lenient that it takes garbage data. Using different regular operations for regex we arrived at the proper format. For example the regex for telephone numbers initially was this :
“(\\d{3}[-\\.\\s]?\\d{3}[-\\.\\s]?\\d{4}|\\(\\d{3}\\)\\s*\\d{3}[-\\.\\s]?\\d{4}|\\d{3}[-\\.\\s]?\\d{4})”
which was too strict and did not take some of the cases like +91-<10 digit phone number > so we later changed it to this :
“(?:?:\\+|0{0,2})91(\\s*[\\-]\\s*)?[0]?[789]\\d{9} ”.

- The different types of languages that are used in a particular career domain, the modules related to that language and the frameworks that are used to handle a particular job need to be classified properly. Out of the few hundred languages, we need to select the topmost that are used and classify them according to the fields in which they are used most often. For example : Perl is used in Networking or the NLTK module of python is used in natural language processing. Similarly babel.js framework of javascript is used in web development. We need to classify the frameworks and modules according to languages also. Doing this requires manual labour of extracting all these information from the web.
- We need to find an accurate and efficient method to rank the CVs that are shortlisted according to a particular job query. To do this we assign a score to all the degrees(m.tech,b.tech, MS, etc) and the different frameworks and languages that are available in the data. We also need to balance the tf-idf score to give an accurate output as a ranked list of CVs.

7.Further Improvements

This system can be provided with many other features to ensure better results such as :

- Going beyond simple word matching, and doing meaning based search. This is quite challenging as simply adding the synonyms can be disastrous. For example, imagine if we add the synonyms for terms like Spark, Pig, Hive. It can worsen and add incorrect features instead of improving it. Also utility like Wordnet (which gives synsets, hypernyms, hyponyms) is not available for such technical terminologies.
- Modelling concepts for feature representation from words (for example C++ ,Java and Object oriented programming may represent one single concept(therefore one single dimension) rather than 3 different concepts) and identifying hierarchy of concepts (example Neural networks, Nearest neighbours, SVMs belongs to Machine learning concept). We can make better systems if we could model these concepts as features rather than treating each word as a separate feature, and apply similarity measures on these representations.
- Taking the duration of a work experience into account to represent the proficiency of an individual at a particular skill. This also tells us the positions of responsibility the individual has worked at.
- Identifying CVs of different domains such as engineering, arts , commerce, etc.