Job Search

Kechen Qin, Xinyue Wang, Yang Liu

Job Search Part:

We wanted to design a search engine that can help students, especially computer science students, find jobs and internships. I built corpus by using indeed API and NLP technologies. There are nearly 12 thousand instances of job information in my corpus including 6000 computer science related jobs. To show the accuracy of our search engine I stored 6000 other jobs in our corpus as noises. There are twelve fields in my schema including 'text', 'job_type', 'job_title', etc. Some are gotten from indeed API directly and others are extracted from text using NLP technologies. Here is one sample of my data:

```
{"text": "The OpportunityWant the opportunity to work in a startup-like
environment while enjoying the benefits of being part of a large dynamic
company? Looking to work with some of the best minds in the industry,
leveraging the latest frameworks to solve cutting-edge problems? Join us
and help develop the next generation in network control and analytics,
enabling networks to be operated with the agility and elasticity that cloud
infrastructure has brought to software applications. For a motivated,
talented candidate willing to push the envelope, we\u2019re offering a chance
to work in a flexible, team-driven environment with competitive compensation
and fascinating technical challenges.ResponsibilitiesAs a key member of a
small team, you will enable high-performance, scalable functions in a virtual
network control system. As a member of the development scrum team, you
will:Develop a carrier-grade product from proof of concept through
productionParticipate in design and vision evolutionReview codeWrite unit
testsTrack and fix defects as they are foundRequired skills and background3+
years of experience in Agile development environmentDevelopment of
horizontally-scalable softwareExperience with a CI/CD model
(Jenkins)Proficient in software architecture and design (data models,
function definition, etc.)Proficiency in various languages including Python
and JavaExcellent technical analysis and communication skillsCollaborative
personality, able to engage in interactive discussions with the rest of the
teamDesired QualificationsBS/MS degree in Computer Science or related
technical fieldExperience with networking products, particularly in
```

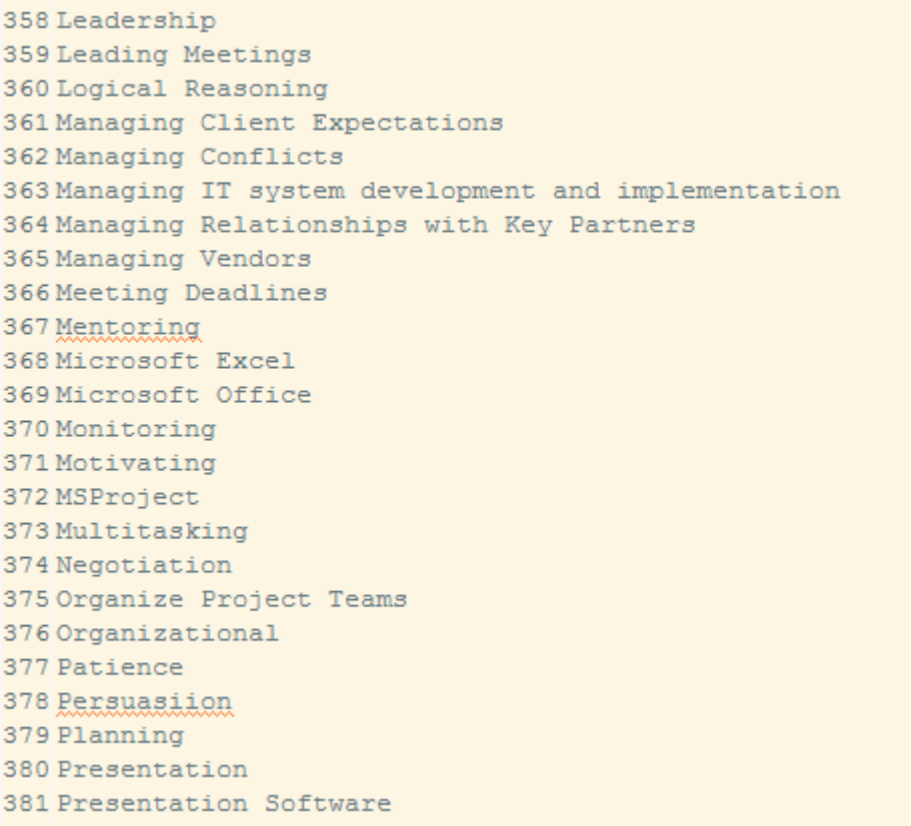
management and controlExtensive experience with software development and the complete software lifecycle (analysis, design, implementation, testing, quality assurance)Worked with NoSQL databases (e.g., Hbase, Cassandra, MongoDB)Implementations based on message queue frameworks (e.g, RabbitMQ, Redis)Experience with OpenStack", "company": "Juniper Networks", "job_type": "fulltime", "date": "Fri, 01 May 2015 17:53:08 GMT", "education": ["master"], "city": "Sunnyvale", "url": "http://www.indeed.com/viewjob?jk=7f153d414237e7c5&qd=NtoM74hyIqUD7jmpokDke51lUoWs1SYrE_M2mnDYjadrfscpxQFzlGrR_AflpbYsCmMTY8e7pqAlO7KpGptniDK76P-yvYWMbE3Ac1oODbQ&indpubnum=7011615221270226&atk=19kj8aig45sole9i", "country": "US", "state": "CA", "plus": ["development", "analytics", "implementation", "communication", "testing", "analysis", "languages", "python", "applications", "computer", "design", "architecture", "databases", "technical", "hbase", "data", "networks", "software"], "location": "37.368134,-122.03297", "job_title": "Software Engineer - Data Structures & Algorithms"}

I designed eight search criteria, such as query statement, location, education level, job skills, etc. Query statement is necessary and cannot be empty. I used both multiword query and bigram phrase query to deal with it. I match query statement with both job_title and text fields and I set a higher weight for job_title field. For education level and plus skills fields I used term query. For location field I used cross_fields to match the search location with city, state and country fields and I used geo_distance_filter to control the search range. For company field I used fuzzy query. I used bool query to combine all these queries.

Besides, I designed a recommendation system for job search. It means that the search result is influenced by user preferences. To implement this, I saved users browsing history first. Then I extracted the useful information from the history data. When user start a new search I will add the useful history data into this search and use rescoring function in elasticsearch to refine results. For instance, I can get user's favorite workplace by finding the most frequently used job location in nearest twenty times search. When a user starts a new search, the job position, which is located in his frequently used place, will rank top.

There are five extra packages in my project. I used indeedclient to download job information from indeed. I used nltk, beautifulsoup and urllib for text processing. I used getcoders to get geographical coordinates based on location.

There are two modules in my part. One is jobsearch.py and the other is util.py. Jobsearch.py is used for implementing elasticsearch part. There are many query functions and schema standard in this module. Util.py is used for data extraction and text analysis, which includes nltk tools and other tools. Besides I have a skills list which is used for skills extraction. Here is a small piece of skills list:



```
Leadership
Leading Meetings
Logical Reasoning
Managing Client Expectations
Managing Conflicts
Managing IT system development and implementation
Managing Relationships with Key Partners
Managing Vendors
Meeting Deadlines
Mentoring
Microsoft Excel
Microsoft Office
Monitoring
Motivating
MSProject
Multitasking
Negotiation
Organize Project Teams
Organizational
Patience
Persuasiion
Planning
Presentation
Presentation Software
```

This part is done by Kechen Qin.

Talent Search Part:

External library used:

Use faker in Python to generate artificial name, email and phone number.

Code list:

search_candidates.py includes two classes: SearchCandidates and CandidateDataGenerator. The first class is used for search and the second is used for generating 1000 records of candidates.

Data sample:

```
{
"data": [
    {
        "name": "Efrem Schulist",
        "job_exp": 16,
        "skills": [
            "D3.js",
                "Analytical Tools",
                "Windows",
                "Conducting Statistical Analyses",
                "Planning"
            ],
            "id": "0",
            "phone": "(795)791-7655x73362",
            "education": "master",
            "email": "chantal91@yahoo.com"
        },
        {
            "name": "Lennie Littel MD",
            "job_exp": 3,
            "skills": [
                "C"
            ],
            "id": "1",
            "phone": "+56(5)7645647348",
            "education": "phd",
            "email": "leila.legros@runolfsdottirgottlieb.com"
        },
        {
            "name": "Bryan Donnelly",
            "job_exp": 2,
            "skills": [
                "Installation",
                "Data Interpretation",
                "Release Scheduling",
                "Written Communications"
            ],
            "id": "2",
            "phone": "587-839-6164x229",
            "education": "phd",
            "email": "olson.tiana@yahoo.com"
        },
        {
            "name": "Nanette Eichmann",
            "job_exp": 8,
            "skills": [
                "Observational",
                "Adaptability",
                "Risk Management",
                "Analytics",
                "Python",
                "Data Science Tools",
                "Languages",
                "Operating Electron Microscopes"
            ],
            "id": "3",
            "phone": "997-290-8537",
            "education": "bachelor",
            "email": "runte.gaither@gmail.com"
        },
}
```

Scenarios:

Use query java and PhD corresponding to skills and education level and then filter job experience on 1-4 years. The return is as follows:

```
[{'name': u'Lennie Littel MD', 'job_exp': 3, 'skills': [u'C'], 'phone':
u'+56(5)7645647348', 'education': u'phd', 'email':
u'leila.legros@runolfsdottirgottlieb.com'}, {'name': u'Hobson Predovic',
'job_exp': 1, 'skills': [u'Embedded Hardware', u'Interviewing', u'Planning',
u'Code', u'Presentations'], 'phone': u'(834)334-9259x1875', 'education': u'phd',
'email': u'vandervort.arielle@wiegand.biz'}, {'name': u'Harley Wyman', 'job_exp':
4, 'skills': [u'Oracle', u'Constructing Surveys and Questionnaires',
u'Troubleshooting', u'Quality Control', u'Structured Query Language (SQL)',
u'Software', u'Analytical'], 'phone': u'646.948.0038x1939', 'education': u'phd',
'email': u'mraz.lum@tillmanjacobson.com'}, {'name': u'Ramona Sporer', 'job_exp':
2, 'skills': [u'Modeling Data', u'Matlab', u'Quantitative ', u'Application
Development', u'Change Management', u'Email', u'Collaborating', u'Data Analytics',
u'Project Management'], 'phone': u'795-978-0569x246', 'education': u'phd',
'email': u'schaefer.ruie@bednardouglas.biz'}, {'name': u'Florian Brekke II',
'job_exp': 1, 'skills': [u'SPSS'], 'phone': u'096-544-3994x500', 'education':
u'phd', 'email': u'versa15@hotmail.com'}, {'name': u'Dr. Adan Cronin', 'job_exp':
3, 'skills': [u'AppEngine', u'C', u'Code'], 'phone': u'1-301-181-3239x0927',
'education': u'phd', 'email': u'pryor.cormier@bruen.org'}, {'name': u'Derrick
Flatley', 'job_exp': 4, 'skills': [u'Process Development', u'Physics', u'Code',
u'Organize Project Teams', u'Analytical', u'Certifications', u'Composing an
Abstract Summarizing Studies', u'Software'], 'phone': u'057-061-8475',
```

'education': u'phd', 'email': u'ishanahan@sanfordmertz.com'}, {'name': u'Bennie Schmitt', 'job_exp': 2, 'skills': [u'Python', u'Graphic Design', u'Diagnostics', u'Computer Programming', u'AutoCAD'], 'phone': u'296-081-6558', 'education': u'phd', 'email': u'lizette29@gmail.com'}, {'name': u'Gladstone Quigley', 'job_exp': 2, 'skills': [u'Data Management', u'Machine Learning Models', u'Diagnostics', u'Data Interpretation'], 'phone': u'243.097.3010', 'education': u'phd', 'email': u'hobert55@hotmail.com'}, {'name': u'Joretta Howell', 'job_exp': 2, 'skills': [u'Development', u'Aspect Oriented Programming', u'Manual Dexterity', u'Spreadsheet', u'AWS', u'Meeting Deadlines', u'Creativity ', u'Assertiveness', u'Java'], 'phone': u'1-363-094-8203', 'education': u'phd', 'email': u'celestia91@gmail.com'}, {'name': u'Coleman Swift MD', 'job_exp': 4, 'skills': [u'Adobe Illustrator', u'Managing Client Expectations', u'Microsoft Publisher', u'Data Analytics', u'Interviewing', u'Programming Methodologies', u'Desktop Publishing', u'Javascript', u'PowerPoint'], 'phone': u'02517068513', 'education': u'phd', 'email': u'romie.hamill@yahoo.com'}, {'name': u'Shellie Murray', 'job_exp': 4, 'skills': [u'Data Analytics', u'Instructing', u'Resource Allocation', u'Matlab', u'System Design', u'Adobe InDesign'], 'phone': u'09381204573', 'education': u'phd', 'email': u'bryana.turner@schowalter.com'}, {'name': u'Demian Dooley', 'job_exp': 3, 'skills': [u'Choosing the Right Keywords for Online Searches', u'Logic'], 'phone': u'354.849.0599', 'education': u'phd', 'email': u'bridgette80@stiedemann.com'}, {'name': u'Frederick Barton', 'job_exp': 4, 'skills': [u'Perl', u'Data Manipulation', u'Interpersonal', u'Google Visualization API'], 'phone': u'450.618.0686x738', 'education': u'phd', 'email': u'lpaucek@kirlin.org'}, {'name': u'Dr. Lone Muller PhD', 'job_exp': 2, 'skills': [u'Revision Control', u'Debugging', u'FileMaker Pro'], 'phone': u'+82(2)2307185774', 'education': u'phd', 'email': u'fjones@gmail.com'}, {'name': u'Mr. Matias Schumm', 'job_exp': 3, 'skills': [u'Operating Systems', u'Maya', u'Databases', u'Estimating Costs', u'Web Applications', u'Producing Data Visualizations', u'Managing Conflicts', u'Industry Systems'], 'phone': u'1-766-934-1598', 'education': u'phd', 'email': u'azariah21@robel.com'}, {'name': u'Jane Windler PhD', 'job_exp': 1, 'skills': [u'Scheduling'], 'phone': u'07921699548', 'education': u'phd', 'email': u'val73@yahoo.com'}, {'name': u'Raekwon Bins DDS', 'job_exp': 1, 'skills': [u'Presentation Software', u'Planning', u'Microsoft Office', u'Web Platforms', u'PT Modeler', u'Release Planning'], 'phone': u'04929099379', 'education': u'phd', 'email': u'abb.ondricka@swiftbins.info'}, {'name': u'Buna Bahringer', 'job_exp': 3, 'skills': [u'Patience', u'Reporting', u'Reporting Tool Software', u'Evaluating the Validity and Reliability of Related Research Studies', u'Maintain Databases', u'Prioritizing', u'Third Generation Languages', u'D3.js', u'AppEngine'], 'phone':

```
u'719-704-1967x33783', 'education': u'phd', 'email': u'yratke@kutch.com'},
{'name': u'Luverne Collins', 'job_exp': 4, 'skills': [u'Certifications'], 'phone':
u'1-085-858-0662', 'education': u'phd', 'email': u'jailyn.leannon@gmail.com'}]
```

Part Description:

As an extension of this project, we implemented another direction of search, which is used by companies in search of potential candidates.

The first difficulty of this part is to get the data since it involves personal information that may lead to privacy concerns. For example, it is not real to get the resume of a job applicant as well as his private data such as phone number, address, etc.

There is one way to handle this problem. First is to specify several fields of a candidate's information. In this project, we have defined some fields in candidates as follows: id, name, email, phone, skills, education, job_exp. Job_exp is the number of years that a person has been working. Next, we implemented a generator to randomly generate candidates of different backgrounds like name, email, phone, skills, education level(Bachelor, Masters and PhD) and job experience. Take skills as an example, we specified a skill list that contains 408 kinds of abilities for job hunting. For every candidate, a random number is used for how many different skills he/she has. Then we pick this number of skills in the list stochastically. For education level, the procedure is mostly the same. For personal information such as name, phone and email, we use faker library in Python, which randomly generates artificial record for a person. In this project, we have 1000 records for search query. And all of them are signed up users in the system in which usernames and passwords are their emails by default.

After getting the data, it is necessary to do the query according to it. For candidate searching, we use skills and education level as multi field match as well as job experience for filter. Searches will be based on skills and education level and the results will be constrained in the range of job experience.

Future work can be on the basis of recommendation. That is to recommend candidates with certain abilities to companies, which facilitates the progress of seeking ideal job applicants. Also, query can be refined to match more complicated needs of clients and add more fields into search such as location, preferred job type, etc.

This part is done by Xinyue Wang.

Website:

In this part, I used Django to create the framework of our website. The main idea is to demonstrate our search functions so it only contains plain html. This website contains two applications, one for job search and one for talent search.

For job search part, I added a user login system to keep track of each user's search history and store their information in our database, not only for optimizing the search result but also provide data for the other part: talent search. And it could remember the user's input if he/she comes back later. This function has two views: one is for user's log in, and another is for creating new user. Both of these views have the ability to handle some errors, such as username or password incorrect, or username already exists etc.

After user log in his/her account, he/she would be able to do the search job as well as edit his/her profile. This profile is mainly used for talent searching part, which means in that part the search engine will help companies to search people based on this data. We have design several query request, based on different type of these requests, I used different form of input method for users to put in what they want to search. All of these could be blank except for "search your job".

By hitting the search button, results will be displayed. Thought there are many information could be used, we choose several information that we think are most helpful for the user to decide whether they should hit the link to review more detailed information or not. The title of each job is an anchor text which is linked to the real address of this job, through which user would be able to apply to this position.

The talent search part is aimed at searching for users that registered in our website. We choose three fields to do the search, and based on their types, I choose different forms of input on the web page. After hit the search button, the user's information could be displayed in order.

If we have time, we would add the company's register and login system to keep track of each company and their job positions, so that the job information could also acquired in our database. Each company could have the information of what positions they provided and how many people applied for this position etc.

Since our user data is generated by ourselves, I added a method to upload these data to our website database. Then another method could be used to upload these data from the website database to elasticsearch database. For the job search part, we directly used the data from file instead of uploading them to our website first. Both of the date loading processes is in __init__.py in each application.

In this part, I used Django to create the framework. If a test is needed, first run the elasticsearch sever, then just cd to the fold of this project in terminal and run: "python manage.py runserver ---noreload" to test the website. After that, open a browser and type "localhost:8000", you will be looking at our website.

This part is done by Yang Liu.