Job Metric

Qihang Mao
Computer Science
University of Colorado Boulder
Boulder Colorado US
Qihang.Mao@colorado.edu

Shanli Ding
Computer Science
University of Colorado Boulder
Boulder Colorado US
Shanli.Ding@colorado.edu

Qiuyang Fu
Computer Science
University of Colorado Boulder
Boulder Colorado US
Qiuyang.Fu@colorado.edu

ABSTRACT

Since every undergraduate considers graduating, each of them either will choose to pursue higher education, Master-degree or PhD-degree, or will dive straight into the industry's field. As for the part of students who goes into the work area, it is always requiring a bold heart and solid knowledge base in the brain. Nowadays, there is a prevalent trend for software engineer's job positions or other programming-related jobs. By standing a little further from the job recruitment area, it is surprising that there is a traditional skill requirement which will not change often. We believe that there should be at least one relationship between traditional skills and job positions. Therefore, our team came up with the idea that the data contains job information that may have an implicit relationship between skills and job. By looking for such implicit relationships we focus on data mining strategy which can help us visualize and conclude our speculation. The experimental results of job information mining clearly demonstrate the relationship which describes what the important skill applicants are supposed to obtain if they apply for the specific job position.

CCS CONCEPTS

• Information systems ~ Information systems applications ~ Data mining ~ Data stream mining

KEYWORDS

data mining; data visualization; job position; job average salary; preferred skills; scrapy

ACM Reference format:

Qihang Mao, Shanli Ding and Qiuyang Fu. 2020. Job Metric. In *Proceedings of ACM Woodstock conference (WOODSTOCK'18). ACM, New York, NY, USA, 3 pages.* https://doi.org/10.1145/1234567890

1 Introduction

According to studying some big data analysis about students who go into industry, more specifically, the paper titled Campus to Career is composed by Glenda Young and David B. Knight[1] and article Career Choice Prediction done by Min Nie, Zhaohui Xiong, Ruiyang Zhong, et al[2]. Both studies concentrate on the students' intrinsic professional skills and the future career pathways which can be construed as jobs engineering students are more likely to choose. As long as the outstanding speed of technology development, technological industries are always required job applicants be more prepared and competitive. Our team thus consolidates the idea about campus-to-career analysis that there may be much deeper relationships within specific job areas. And we will put efforts to mine out available relationship connections as much as possible. Therefore, we formed our base questions for solving data relationships, and questions can be divided into different three parts: what kind of feature data we want to use(data preprocess); how can we get known with features and make visualization analysis(data mining); and, lastly, what can we learn from all data visualization analysis.

First of all, efforts to shrink data scope are as indispensable as the first step into analyzing discrete data from all around the internet. Due to getting data by spidering the website, we have put the security check at the very first place before we start the experiment. And after we have affirmation from the website's owner or host, and then we will take another step into the research. To be specific, we are supposed to finish researching and discriminating the clusters of different kinds of data. Hence, we preprocess the raw data under consensus of the team that some of the most crucial factors are job position, experience requirement, expectation from job recruiter, and job earnings separately.

Secondly, one of the most critical visualizations our experiment needs to create is the correlation matrix. Because after classifying the importance of data by human, then we need to dig into these data by applying data visualization strategy. Once we have our correlation map, subsequently, we can determine the level between most related and least related kinds of data. Moreover, we are focusing on hot-key skill names as well, so as long as we have the relation map and hot-key skills we can then form up rudimentary thoughts about hidden relationships among the dataset.

Lastly, following track to make conclusions about our experiment has highlighted converging all emerging conjectures into one conclusion about job analysis. Starting from massive and mixed data to clearly depicting data visualizations. The purpose of mining out hidden possible relationships between job applicants and job requirements can be corroborated by efforts to analyze every feature we have extracted. In the nutshell, we try to contribute to the relationship between skills and job positions which may help applicants know much better about the job they apply for.

2 Milestones

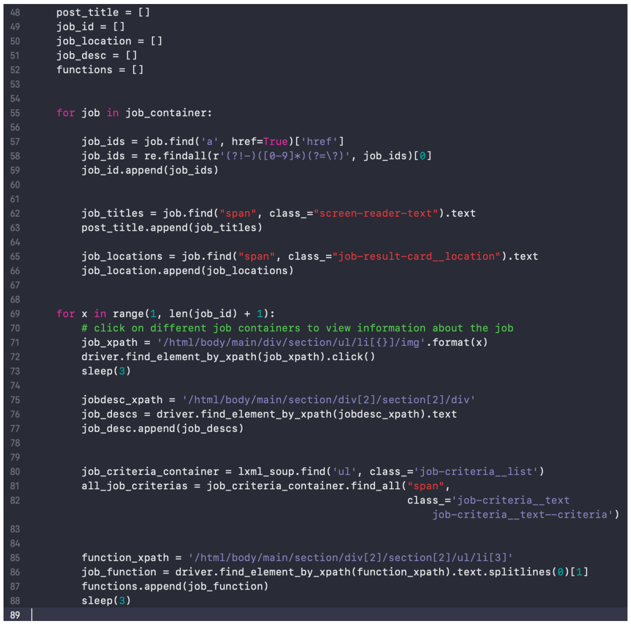
2.1 Data collection

We collect data from open source, more likely from websites linkedin[3], handshake[4]. Before the data collection process, we need to finish the security checking which can assure our behavior is legal and we can continue collecting data safely. During the data collection process, the main strategy we will apply is crawling, more specifically, we try to spider information from websites' job pages via python. We search for keywords; check the frequency of these skill names appearing on the website, and record them as raw data. The data collected by this method is relatively rough and needs to be further screened because we crawl down all available information provided by websites. Moreover, We have also considered using existing datasets that were collected by other data analysis projects, but there is not a condign dataset we can use for our particular research purpose. Therefore, we agreed to collect the data ourselves. Although the data is viewable and readable on websites for people's innate brain system, we are supposed to convert the information which is readable by humans to the data which is readable by machines.

2.1.1 Data Scraping

Before we program our scrapers, it is essential to closely inspect the specific website's overall structure using the console in Chrome. For instance, we can't scrape the data off the sites without looking into the HTML elements because they basically tell us what information is stored in each tag. In our project, the Linkedin jobs' data are usually stored in "<span></span>" inside the "classes." And there are several notable tools that can help us collect the data: BeautifulSoup (parses HTML and XML documents), Pandas (data manipulation and analysis), and Selenium (automates browser activities).

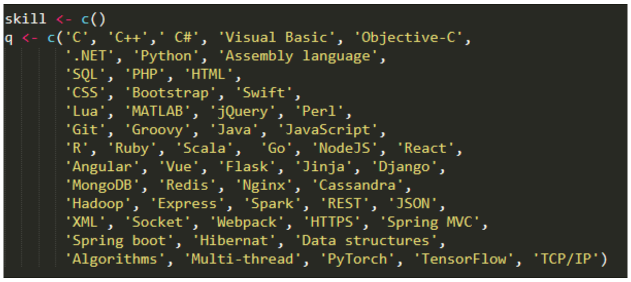
Take joinhandshake.com as an example, we found out that it might be necessary to configure the headers in our code, so the script is able to access the website more smoothly. Basically, we manually assign the desired values to the attributes to allow the bot to "open the gate" of the website. After that, we program the scraper and enable it to inspect the HTML elements, then parse the data using BeautifulSoup, and finally store each piece of data collectively in arrays.



```
    post_title = []
    job_id = []
    job_location = []
    job_desc = []
    functions = []


    for job in job_container:

        job_ids = job.find('a', href=True)['href']
        job_ids = re.findall(r'(?!-)([0-9]*)(?=\?)', job_ids)[0]
        job_id.append(job_ids)


        job_titles = job.find("span", class_="screen-reader-text").text
        post_title.append(job_titles)

        job_locations = job.find("span", class_="job-result-card__location").text
        job_location.append(job_locations)


    for x in range(1, len(job_id) + 1):
        # click on different job containers to view information about the job
        job_xpath = '/html/body/main/div/section/ul/li[{}]/img'.format(x)
        driver.find_element_by_xpath(job_xpath).click()
        sleep(3)

        jobdesc_xpath = '/html/body/main/section/div[2]/section[2]/div'
        job_descs = driver.find_element_by_xpath(jobdesc_xpath).text
        job_desc.append(job_descs)


        job_criteria_container = lxml_soup.find('ul', class_='job-criteria__list')
        all_job_criterias = job_criteria_container.find_all("span",
                                                            class_='job-criteria__text
                                                                job-criteria__text--criteria')


        function_xpath = '/html/body/main/section/div[2]/section[2]/ul/li[3]'
        job_function = driver.find_element_by_xpath(function_xpath).text.splitlines(0)[1]
        functions.append(job_function)
        sleep(3)

```

Figure 2.1.1.1: Code Snippet

Note that since bots can perform numerous operations in one second and humans cannot compare this in any way. Thus, we have included the "sleep()" function to simulate a real person looking for job information on the website. It does slow down the scraping process a lot, but in exchange, it can successfully avoid being detected by the "robot check" system. After we have gathered the information we need and exported them to CSV files, we would like to further analyze the data: filter the data and extract the technical skills and the frequency of how often they get mentioned.



```
skill <- c()
q <- c('C', 'C++',' C#', 'Visual Basic', 'Objective-C',
         '.NET', 'Python', 'Assembly language',
         'SQL', 'PHP', 'HTML',
         'CSS', 'Bootstrap', 'Swift',
         'Lua', 'MATLAB', 'jQuery', 'Perl',
         'Git', 'Groovy', 'Java', 'JavaScript',
         'R', 'Ruby', 'Scala',  'Go', 'NodeJS', 'React',
         'Angular', 'Vue', 'Flask', 'Jinja', 'Django',
         'MongoDB', 'Redis', 'Nginx', 'Cassandra',
         'Hadoop', 'Express', 'Spark', 'REST', 'JSON',
         'XML', 'Socket', 'Webpack', 'HTTPS', 'Spring MVC',
         'Spring boot', 'Hibernat', 'Data structures',
         'Algorithms', 'Multi-thread', 'PyTorch', 'TensorFlow', 'TCP/IP')
```

Figure 2.1.1.2: Filter Script

2.2 Data cleaning

Since we get the dataset, we need to do data preprocess, clean it up, before we use it. Because the data we collect is manually collected, there may be various problems like the feature in data is totally not related to our research, etc. Therefore, data cleaning is as essential as choosing the place to build a skyscraper. The specific cleaning method should be determined according to different datasets. Most of the cases are to delete the outliers to reduce the impact from them on the dataset and supplanted by the considerable value, *Nan* or *Zero*. Plus, the missed value can be selected from the average, mode and median which depends on the data's type. And as for our experiment, the average is preferred. After cleaning the dataset more aggregation (the degree of dispersion will be much smaller than the original data), which will facilitate our use and analysis. According to these datasets, we can find obvious patterns reflecting its general trend. Data cleaning helps us get more accurate conclusions.

2.2.1 Valid Features Sanity Check

As when we get the raw data set from the data collection process, we realize that there is some unrelated information. All feature data types include Role Name, Location, Description, Function, Salary, Job Type, and Employee Size. However, it is obvious that Function, Job Type and Employee Size are features that are unrelated, so we dropped these three columns' data. And then, we looked through the raw data set, and we found out a problem that Salary's column contains many either NaN or N/A. Therefore, by consolidating our speculation, we display a pie-chart from the Salary column. And not surprisingly that valid salary data only has less than 20% out of all data. Hence, we dropped Salary column as well. And finally our modified data set contains features of Role Name, Location, and Description.

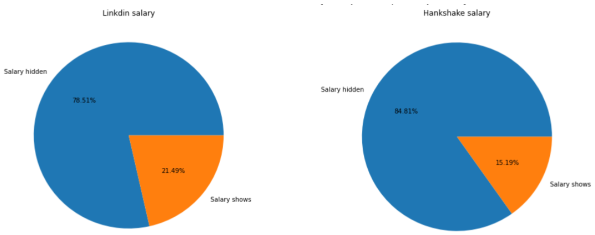


Figure 2.2.1.1: Pie-chart of Salary Data

Subsequently, because we are still doubtful about the Location column containing some disruption data like places where are not in the United States. And truly, there are locations in China, Ireland, India, etc. However, the good thing is that these disruption information only include less than 50 cases out of our 7732 total cases. Therefore, we have done one more step of the data cleaning process on the Location column. Finally, our final output data structure is in correct form which contains useful features and it has a specific geographical area in the United States. Plus, we merged both Linked in and Handshake's data set together, so we can have a full analysis on the modified data set.

2.3 Data mining

According to the data we collected, we can use models and formulas learned in class to understand the relationship between these different data, that is, data mining. Through the seemingly unrelated situation of these data, we can dig out their inner deeper connection and find some laws that are hard to find. These laws are very critical and important, but also often ignored by people, through data mining, we can see clearly. Data mining is not only to find the deeper data, but the rules they contain are the most valuable things.

Graphics is an important and useful tool for data mining. As we learned in class, different graphs can do different jobs, such as the histogram, which can reflect the number of data in the dataset. We can find the most popular variable or the least variable through this graph, and at the same time, we can see the size relationship of the two variables; the line chart can reflect the trend of the data, through which we can have the trend of the data change forecast, the variable of upward trend will become larger in the future, the downward trend will become smaller, and the stable trend will remain unchanged. With the prediction, we can easily carry out the future plan; box chart, which can reflect the maximum value, minimum value and average value of the dataset in one graph. According to this graph, we can easily see the range, discrete degree to analyze whether the variables represented by this dataset will have large fluctuations, so as to make prediction

2.3.1 Word Frequency Ranking: (Word Cloud)

After we get the required dataset from the data cleaning part, we try to mine the information hidden in the dataset, we decided to use an interesting graph way to show the conclusion -- Word Cloud. Firstly, we decide to find the locations ranking. This describes which state has the most posts, so that employees can have a higher chance to find internships there. We pick all state names from the working location to draw this word cloud about the location ranking. From this graph, we can find California, Texas, Colorado, New York can be the top ranking in location.

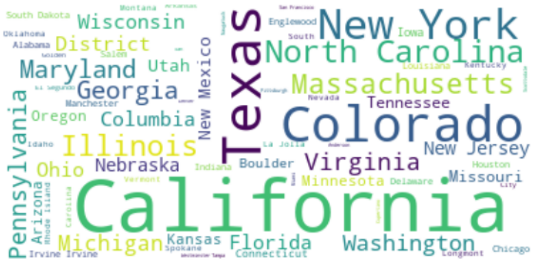


Figure 2.3.1.1: Location Ranking Word-Cloud

Secondly, we decide to draw a word cloud about keywords in description. These words are most often mentioned in description. According to this word cloud, users can find the most popular words shown in the role description, which can help them make a good impression when introducing yourself to the interviewer. We can see “experience”, “work”, “team”, “support” are top words, and this is meaningful. In the future work space, employers prefer an experienced employee, and they like those who can cooperate with teammates and support each other.

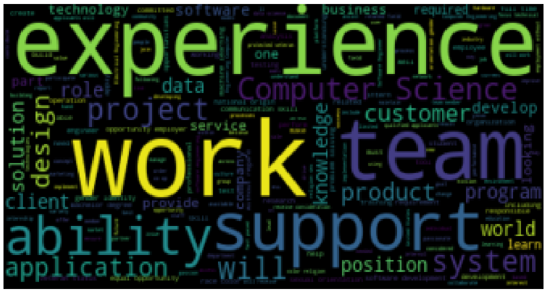


Figure 2.3.1.2: Description Keywords Word-Cloud

Moreover, our team also made another word cloud for visualizing the popular job roles. We took out a single column named “Post” to calculate the word frequency. As figure 2.3.1.3 shows, we can find out some conspicuous job roles in the word cloud. Because we only want to consider the job role types, we take out words such as “Intern”, “Internship”, “Summer”, and “Grad”. Additionally, the word cloud computing only considers words one-by-one, we thus concatenate words that are at the same quantity level in human logic. Therefore, there are three outstanding job roles among our 7211 cases: software engineer, data analyst, and machine learning.

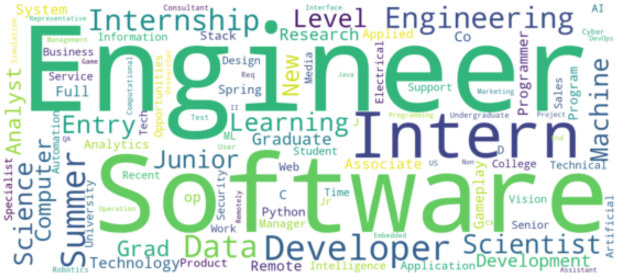


Figure 2.3.1.3: Role Types Word-Cloud

Lastly, we have filtered the information in “Description” that contains detailed job descriptions for professional skills ranking. Although there are some difficulties on both unknown characters and none skills are described, we processed back to data cleaning, finally we got a reasonable outcome. As figure 2.3.1.4 shows, you can find the top-ranked required skills. Besides, as we want to prove our finding’s correctness, we referenced *TIOBE Index* [6]. As the rankings shown in *TIOBE Index*, because the ranking depends on the number of search hits, we only consider the growth of the popularity of the skill. Therefore, the largest growth rate belongs to Python, which is 1.90%. And also, look back to our mined out information, it is corresponding to the high growth rate. And the necessity in Python ranked in the first place.

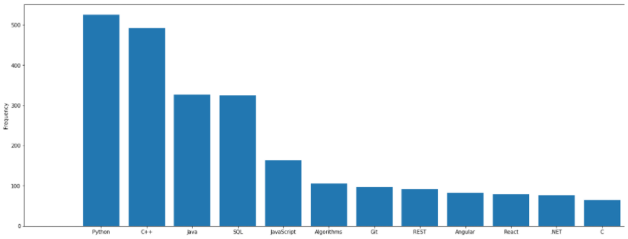


Figure 2.3.1.4: Skill Ranking
(Python, C++, Java, SQL, JavaScript: First 4 skills)

2.4 Form Conclusions

We get conclusions and visualizations from the data mining process, and then analyze these conclusions. In order to achieve our ultimate goal: how to provide the applicants more accurate information about the specific job position, and bring them more advantage to win a job or internship. Walking through data collection, cleaning, mining and final analyzing, the conclusion is as solid to draw. Through this experiment, users can find out which position is suitable for with their own professional skills. More specifically, according to the analysis of skills ranking, people can have a clearer sense of which skill is in high demand.Python, C++, Java and SQL are popular programming languages in the IT area. If users want to improve their skills or learn a new programming language, focusing on these languages can be a good choice. Moreover, due to geographical location analysis, we can find out the first five cities in the US that IT workers are in high demand. These states are California, Texas, Colorado, Florida, and New York. Therefore, users can have a good sense for which state is growing a technical working environment recently. In addition, the more cities grow up, the more opportunities people can get. Finally, as we analyzed the keywords in each job's description, it is obvious that every employer needs an applicant who can work as a team, support other colleagues, and have some experiences for the job that applicants apply to. In a nutshell, users can gain merits if they learn the conclusions of our project. And also, the source code can be updated as time goes, so this project can be used for analyzing future trends of the working environment. By all means, people can be more prepared to apply, rather than aimless large-scale delivering resumes.

2.5 Improvements

For the data we collected, we may meet data localization problems, which means the data we collect can be influenced by the location of data collecting. We found out that since websites like Linkedin and JoinHandShake use users' personal information and customize search results; for instance, at JoinHandShake.com, since the website specifically links CU Boulder students, many of the job positions are located in Colorado. One of the potential solutions might be clearing out all personal settings so it prevents the website from customizing search results. Additionally in the description part, the content is too long and there is lots of unhelpful information. If we can come up with a solution to parse the long paragraphs and sort them all into short, accurate and more helpful information. Not only it greatly reduces the length of time for reading the content, but also significantly improves user experience.

3 Related Work

3.1 Analysis of campus to career future pathway

Our project focuses on mining job keywords on websites and surveys to get relationships between them. And this can be a popular area, and many projects can be similar with our project. But their data is massive and mixed, our data is spidering specific data, which means our project is more concentrated on graduate students. We recommend graduates who face finding jobs and internships use our project. From campus to career can be a difficult task for graduates, and we try to make the process easier. But for adults looking for jobs, our program may not be ideal. Our project is not to help users find jobs, but to provide users with information to help them make the best choice.

4 REFERENCES

[1] G. Young and D. B. Knight, "Campus to career, understanding how engineering student skill perceptions link to future career pathways," 2015 IEEE Frontiers in Education Conference (FIE), El Paso, TX, 2015, pp. 1-4, doi: 10.1109/FIE.2015.7344258.

[2] Nie, M.; Xiong, Z.; Zhong, R.; Deng, W.; Yang, G. Career Choice Prediction Based on Campus Big Data—Mining the Potential Behavior of College Students. Appl. Sci. 2020, 10, 2841.

[3] linkedin: https://www.linkedin.com

[4] handshake: https://www.colorado.edu/career/students/find-job-or-internship/handshake

[5] Class ppt: https://canvas.colorado.edu/courses/65838/files/22995836?module_item_id=2305283

[6] TIOBE Index: https://www.tiobe.com/tiobe-index/