Job Site Aggregator
Rashmi Laddha

Problem Statement:

According to US department of labor statistics, number of job seekers in USA in the month of July were over 100 million. Count of job seekers is more than 100 million in the world. A job seeking candidate will use channels like web search, personal contacts, consulting agencies etc to find the job. However, web search is the most widely used tool.

There are many job-sites that post the job vacancies for companies. However there is one problem with searching jobs on multiple web sites. All jobs are not posted on one site. A job seeking candidate will not want to lose even a single opportunity. It may lead to loss of time and opportunity. There may be same vacancy posted on multiple sites. This leads to duplication. Therefore, I seek to develop a job aggregator tool. It will take the required job field from user, collect the job options from multiple web-sites, remove duplicates and present the result to end user.

Executive Summary:

This report presents the status of a job aggregator tool. Data is being acquired from job sites viz. SimplyHired, Monster,Indeed, glassdoor and CareerBuilder. For the purpose of this report, I am using "data scientist" as the input field.

Below is the status report:
It shows the details of jobs found on various web sites.

Website	Total Jobs	Duplicates Found	Details Missing	Distinct States
Monster.com	127	0	0	29
Careerbuilder.com	93	0	0	26
Simplyhired.com	2396	0	34	32
Indeed.com	990	0	0	21
Glassdoor.com	521	204	218	28

When I divide all the available jobs according to US-States in which the position needs to be filled, I found that California has more job positions than any other state in USA. I have presented the detailed statistics in appendix.

This job aggregator tool will enable job seekers to view the available job positions for Data scientist that are posted on all the five web-sites without repetition.

Recommendations:

A job seeker would want to have as many options as they are available. This tool strives to achieve that goal. Tool can be further enhanced to grab job postings from other forums like stack overflow. I recommend, as next step:
Developing an automatic cover letter creator according to job description and keywords in job posting.

Data Acquisition Experiments:

I grabbed job postings by decoying our program as web browser and by giving delay in accessing the web pages. The collected data was stored in dictionary in a program. The program was written in Python. Key for dictionary is a string concatenated with state, company name, job title, job location. This key will help in removing duplicates. If duplicates are found. latest job post is retained. Value for each key is job title, company name, job location and link to the web site from which the posting was scrapped. There is a separate dictionary for each web site. The dictionaries are converted to json format. Subsequently all json files are aggregated to make a single dictionary without duplicates.
Program also has a dictionary that contains company information and state information. These dictionaries can be used to derive statistics at company level and state level.

The table given below shows the summary of list of tools used for web sites and the difficulty level in obtaining the data.

Parameter	**Monster**	**Career-builder**	**Simply-hired**	**Indeed**	**Glassdoor**
Structure of Data	1	3	4	4	2
Ease of scrapping	1	5	4	2	1
Tools Used	lxml	API	lxml	beautiful soup with regex	beautifulsoup

Structure of Data scale : 1 to 5 with 1 being very difficult to comprehend and 5 very easy to understand.
Ease of scrapping scale : 1 to 5 with 1 being very difficult and 5 being very easy to scrape.

Data Challenges:

One of the biggest challenge was to get the required data from company provided APIs. APIs only pre-specified data. For example, LinkedIn has an API. It is easy to get data from the API. But the data didn't contain the required details. Similarly, glass door jobs API was not giving the required information. Therefore, I tried to use SELENIUM package to login and extract sites. However, our attempts were still not successful.
Our attempt with beautifulsoup to extract details from glassdoor was successful but not with LinkedIn. Therefore, I am sadly excluding linkedin from our Job-Aggregator and using data from 5 web sites.

Solution Approach:

On all the 5 job websites, major challenge was to understand the data structure and write specific python code to grab data.
Next challenge was to clean up the data. I used tools like lxml and beautifulsoup. All the job postings do not have all the details. And, sometimes there are duplications in job postings. Therefore, I created separate dictionaries for each site to collect the details regarding duplicates, and incomplete data. I am presenting statistics for each site so that error can be isolated easily.

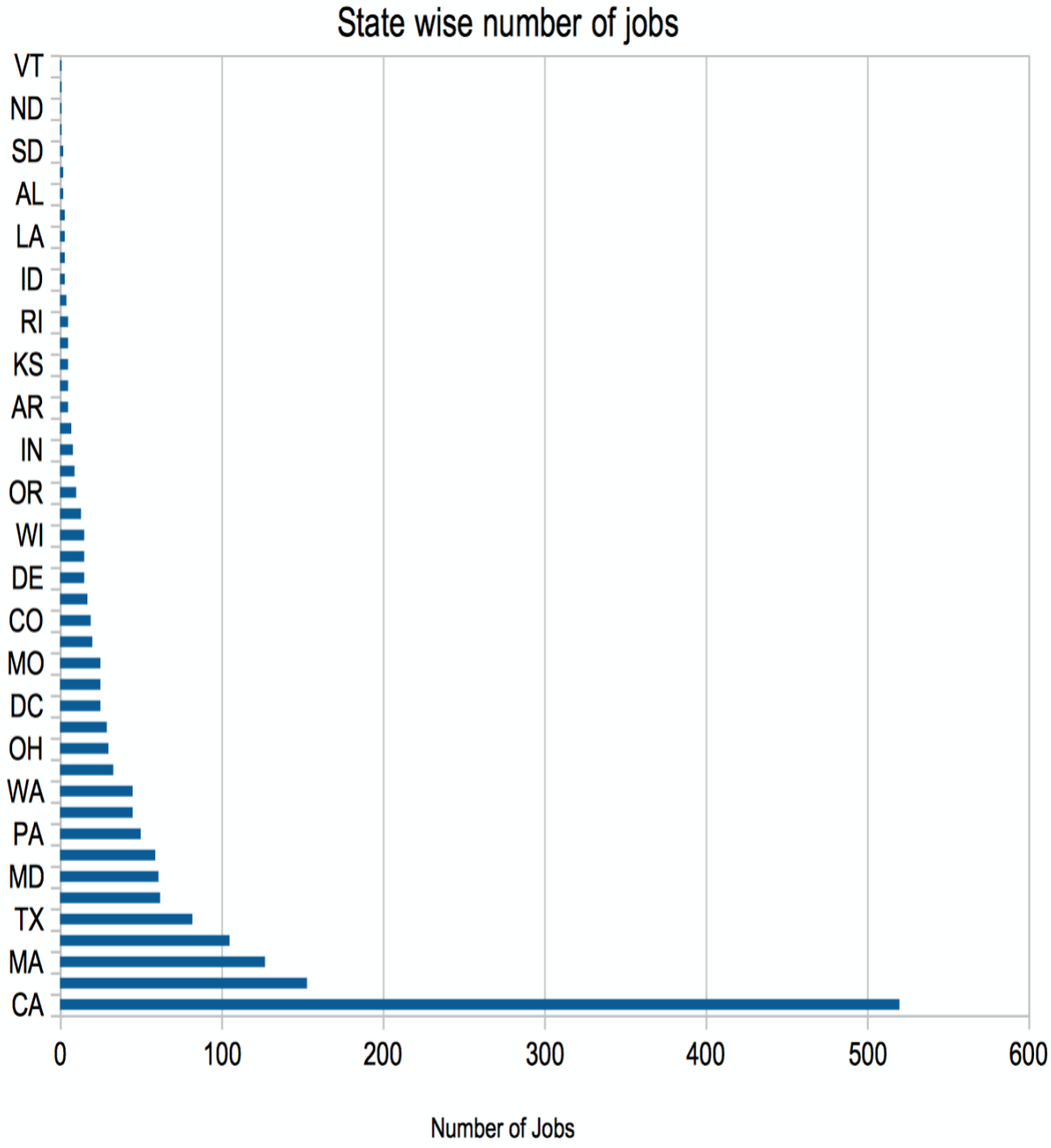
Assumptions:

Our key for removing duplicates is a concatenated string of state, company name, job title, job location. It could be possible that all job sites may not use same text while describing these feature of job e.g. If for the same job, the text for job title is not same on different sites, then duplicate jobs will appear in our results.

Summary and Conclusions:

I want to create a job aggregator from with as many jobs as possible. As of now I am successful in getting jobs from 5 web sites. Our analysis shows that California has more number of jobs for Data Scientist than any other states. Also, maximum number of duplicates were present on glassdoor.

Appendix:


State wise number of jobs
VT
ND
SD
AL
LA
ID
RI
KS
AR
IN
OR
WI
DE
CO
MO
DC
OH
WA
PA
MD
TX
MA
CA
0
100
200
300
400
500
600
Number of Jobs