Mining the US Technologist data

Abulitibu Tuguluke
DHI Group Inc.
6465 Greenwood Plaza Blvd
Greenwood Village, CO 80111
abulitibu.tuguluke@dhigroupinc.com

Abulitibu Tuguluke
Department of Computer Science, CU Boulder
430 UCB, 1111 Engineering Dr,
Boulder, CO 80309
abtu8803@colorado.edu

1 PROBLEM STATEMENT/MOTIVATION

There are many reports stating that the US is (or will) face a technology workers (Technologists) shortage. Our study will mine through the US H1B data set along with Dice.com's job and candidate profiles (along with skill sets for each job title), to understand which technologist job are in increasing/decreasing demands, which skill sets are more popular/unpupular, through time. Our goal is to better understand the US technology market and what we can do to help people who try to get into tech market, by providing them with skillset guidance.

The motivation fo this report came from a conversation with one product manager stating there are no analysis on technologist supply and demand, even on salary data: The product analysts stated they don't have the data, and the R&D team think it is a analyst job and should not 'waste' time on. I decided to try that as a personal project. As I dive in more, I realized the official data set provided from department of labour is vague and uninformative, most of the job titles are lumped together, and focusing only on salary. I decided to mine my own data, while adding relative skill set by mining seperate job description data set.

My philosophy to tackle this problem/project is to use a sompple algorihtm on multiple complicated sets, and not the other way: some complicated algorithms on a nice set. Personally, I had some 'fatigue' issues with words like 'Deep' and 'Neural' for now, both at work and school, so I've decided to do some old fashion minings by not using heavy GPU compyting with 'sexy' machine learning titles.[1].

2 LITERATURE SURVEY

SpaCy's Pipeline component for part-of-speech taggging[2] comes in really handy for the data preparation. What we did was to manually add a 'Tag' button for SMEs to spot the 'tech' words from our training set in HTML, like the following table.

	Sentence	Word	Tag	POS	Job description #
0	1	Softwares	Tech	NNPS	description: 1
1	2	SQL	Tech	NNP	description: 1
2	3	Java	Tech	NNPJ	description: 1
3	4	Web	Tech	NNS	description: 1
4	5	Azure	Tech	NNP	description: 1
⋮	⋮	⋮	⋮	⋮	⋮

Once we got enough samples, we apply the Viterbi algo to the future data set and mine the list of words for each job post, we then trim the list to get something like the following for sample job titles.

Job	Skills
Net Application Developer	Microsoft technologies;Software development;C#;HTML;Quality assurance;ASP.NET;Visual Basic .NET;.NET;Agile.
Android Developer	Software development;Java;Mobile development;Quality assurance;Android development.
⋮	⋮

3 PROPOSED WORK

Shockly, the department of labour data does not give any insite on specific job title but a very general one. The following 3 code has most of the tech jobs:

- Management Occupations 11-0000
- Business and Financial Operations Occupations 13-0000
- Computer and Mathematical Occupations 15-0000

Roughly 35 of them instead of what we proposed of roughly 1400+.

```
['Computer and Mathematical Occupations',
 'Computer Occupations, All Other*',
 'Mathematical Science Occupations, All Other',
 'Computer Occupations',
 'Miscellaneous Computer Occupations',
 'Computer Occupations, All Other',
 'Mathematical Science Occupations',
 'Miscellaneous Mathematical Science Occupations',
'Data Scientists and Mathematical Science Occupations, All Other'
 ...
 ]
```

Which mean we either has to add a new attribute by clustering the 1400 into 15, or again write a new scrapper it mine glassdoor's data()This should not end up being a scrapper project), which does

[1] I will leave that to CSCI 5922-Neural Networks and Deep Learning, this fall.
[2] https://spacy.io/api/tagger

CSCI 5502, Boulder, University of Colorado

DOI: 10.1145/nnnnnnn.nnnnnnn

not guarentee accuracy since all are self-reported, which could be more biased than H1b data. Somehow, we have to reconcile that.

- Data scraping: Screaping HTML data from the web and store them in seperate json file
  EMPLOYER : String, Nominal
  JOB TITLE : String, Nominal
  BASE SALARY: Float, Interval
  LOCATION: String, Nominal
  SUBMIT DATE: (date) String, Ordinal
  START DATE: (date) String, Ordinal
- Data cleaning: The most time consuming process, we need to be able to get rid of the outliers once we spotted them.
- Data preprocessing: Converting the saw data into meaningful use, for numeric data, we need to convert them into float64, date has to be standard date type.
- Data integration: Integrate internal data with mined H1b data by join.
- Data mining and analysis: Clustering and time series.

4 DATA SET

- H1B data scraping from from open data online.[3] The data set is mainly from the United States Department of Labor (DOL) on how many H1-B petitions were filed and approved, with detailed information such as Employer, the title of job, city/state locations, along with base salary, submission and acceptance dates.
- Skill set (Internal data)
- US Bureau of Labor Statistics [4]
- Data mined and stored on my machine (Prcessor: 2.6 GHz 6-Core Intel Core i7. Memory: 32 GB 2667 MHz DDR4. Graphics:AMD Radeon Pro 5500M 4 GB Intel UHD Graphics 630 1536 MB).

5 EVALUATION METHODS

There are many ways how we can evaluate the results. Clustering Performance Evaluation Metrics like Silhouette coefficient is a good start when applying clustering algorithm, for time series, all the error evaluations. Besides, comparing time-series analysis data within each job we can compute the increaing or decreasing sides of each job along with correlations. Finally ,we are looking forward to apply all the relavent techniques we are currently learning in the class.

As for salary data, percentile based on title, location, and or starting time is definitely the first thing to try, 99% and 1% are definitely outliers for that, and balancing what percentile is something I need to be careful about. I am not thinking of 'elimination' since all data counts for that column, I may if it is way imbalanced. Hence we need separate analyses for both individuals and employers. For skill sets, I am considering density distance, which is yet to explore.

6 TOOLS

- Beautifulsoup for mining.
- Python library for all the computing and analysis.
- JSON for data store.

7 MILESTONES

The Following table is my work plan for carrying out the project.

Date	Work Plan
Week 1 - 6: 16 May - 26 June	Research and data scrapping while class works.
Week 7: 27 June - 3 July	Submit Project Part 1: Project ProposalsAssignment.
Week 8: 4 July - 10 July	Data cleaning
Week 9: 11 July - 17 July	Submit Project Part 2: Proposal PaperAssignment
Week 10: 18 July - 24 July	EDA and Classification Algorithms.
Week 11: 25 July - 31 July	Submit Project Part 3: Progress ReportAssignment.
Week 12: 1 August - 7 August	Continue writing the project, and building interactive demo.
(Final)Week 13: 1 8 August - 10 August	Submit Peer Evaluation Form, Project Final Report(with codes), and Project PresentationAssignment(Video).

8 PART 1 PROPOSAL FEEDBACKS FROM CLASSMATES

Pratik Prasanna Raghavendra

Interesting topic, Abulitibu. Just curious, what would outlier data look like in your case? Also, how do you detect and eliminate them, since for example, really high income may simply be an outlier for individuals, but an important point for other employers?

Rohit Kharat

I find this a unique topic and interesting to analyze in today's scenario. I liked that you are using web scraping, which I enjoy; using the BeautifulSoup library for web mining is fun. I am not familiar with the SpaCy library, something you can shed a light on. I also liked the idea of using location information for your analysis. I would recommend geolocation API something which I used earlier in my projects. I would also recommend doing feature reduction or getting the important features in your proposed work for focusing on only the crucial attributes.

Abhinav Gupta

This looks like an unusual topic! Curious to see what your data cleaning and preprocessing would look like. As your job skill data is internal, I don't know how it looks like, but I have a couple of questions. First question is that technologies are evolving and higher demand will always be for the new tech. As that tech, perhaps, is not even in the market yet, how would you predict the demand? Second question is if you would be considering countries

[3] https://h1bdata.info/index.php.
[4] https://www.bls.gov/developers/api$_$python.htm

Table 1: Scrapped data as off July 15th

	Dice_job_title	EMPLOYER	JOB TITLE	BASE SALARY	LOCATION	SUBMIT DATE	START DATE	Job_Title
count	2495079	2495052	2467304	2467304	2467304	2467304	2467304	2495079

and degrees as visa approval ratio varies highly on these factors. Final question is that as per uscis.gov - “The H-1B temporary visa program has been exploited and abused by employers primarily seeking to fill entry-level positions and reduce overall business costs”, so wouldn’t your dataset be corrupted too?

Varun Manjunath

An interesting topic. But is often the case that H1b workers are paid less as H1b is a nonimmigrant visa and is temporary in nature. So the point where you perform analysis on the dataset to check whether H1b workers are paid less is not very useful. Rather a more interesting trend would be to check which location hires the most H1b candidates. Dataset description with a bit more depth would be great in your case. Also, are you planning an inner join query to combine the datasets? Would suggest you remove certain outliers in the combined dataset before performing K-Means clustering.

Danielle Aras

This is a very interesting and relevant topic. I agree that data cleaning is an important part of your project as you are merging different data sources with different features. I would be curious to know more details about the data sets like the size and date range. Technology is a fast moving field so the most current data will be the most useful for job seekers; looking at which jobs are trending up or down is a great idea.

I agree with other posters that the H1B visa data will introduce some challenges to your analysis. Given the trend of employers looking to take advantage of the system to under-pay workers, the popularity of H1B workers in a particular job may not actually correspond with a genuine demand that is not met by domestic workers. An interesting "spin-off" question would be creating an algorithm to detect companies abusing the system based on the availability of candidates in their area and H1B salaries.