LinkedIn, Glassdoor Skill and Salary Analysis

Long Yang
Courant Institute of NYU
New York, US
LY603@NYU.EDU

Jing Xia
Courant Institute of NYU
New York, US
JX379@NYU.EDU

Zhaohui Zhang
Courant Institute of NYU
New York, US
ZZ609@NYU.EDU

Abstract

Employment opportunity, good salary, manageable work-life balance are dreams of almost every people. This paper aims to help job seekers, especially college students, to enhance their prospects of getting better salary via learning valuable skills. In the first section, we analyzed the features of datasets from two professional websites, linkedin.com and glassdoor.com to produce (company, position, skills) and (company, position, salary) pairs. Then we implemented algorithms to join the data from the first step to generate (company, position, skill, salary) pairs to give job seekers valuable advices. Finally we evaluated the hot topics of job skills from twitter to predict the job market trend.

Keywords—LinkedIn, Glassdoor, Skills, Salary

I. INTRODUCTION

Along with the scare of 2007's Subprime mortgage crisis running away, more and more companies start to hire new employees, both new opportunities and challenges appear for students, we are starting this project aiming to provide more valuable advices for students on what kind of top paid skills you should learn for certain industry's certain position. We use the user profiles data from LinkedIn, from which we can extract company name, position, and skill sets for each LinkedIn user, to form the (company, position, skill sets) pairs. Inside each pair, we do a rank for each (company, position) pair to sort out top 5 skills. Then we also get (company, position, salary range) pairs from Glassdoor. Using these two data form, we join to form the (company, position, skill sets, salary range) pairs. Finally, we can compute the average salary range for a skill in certain company's position, and a final ranking list, which to be presented to students for advices, which skills worth more. To achieve all these goals, we use HDFS and HBase to store the raw data got from both LinkedIn and Glassdoor, and MapReduce to do the parsing and ranking job, Pig to do the join job, and D3js to do the final visual display. Hopefully, all these advices can be helpful in students' skills choosing process and give them a bright career.

II. MOTIVATION

The financial crisis in 2007 gives tremendous negative impact on us economy. The unemployment rate displayed in the US Bureau of Labor Statistics is 6.7% which is 2.4% higher than that was 4.4% before the crisis. It's well-known that unemployment cause enormous harm to workers, families and communities. In today's labor market, job skills relates with salary greatly. This paper tries to address these questions for job seekers: What's are the hottest job positions today? What kind of skill should I learn to get a secure job? How much can I get paid? Therefore people may benefit from these answers when hunting a job.

III. RELATED WORK

To start this analytic research, we read many related papers as the first step. These papers are:
Paper [1] presents a new cluster-computing framework called Spark, which supports applications with working sets while providing similar scalability and fault tolerance properties to MapReduce. Spark introduces an abstraction called resilient distributed datasets (RDDs). An RDD is a read-only collection of objects partitioned across a set of machines that can be rebuilt if a partition is lost. Spark can outperform Hadoop by 10x in iterative machine learning jobs, and can be used to interactively query a 39 GB dataset with sub-second response time.
Paper [2] describe Bigtable, a distributed storage system used at Google for a wide variety of applications. Bigtable presents a different abstraction of data than the other distributed storage systems we've looked at, allowing a fair amount of control over locality, memory residency, and version control over particular items in the table. Bigtable's achievements in scalability and application author control are commendable (though perhaps near duplications of work that we've read in recent weeks). The work on optimizing for "sparse" and multi-dimensional maps is interesting and should influence distributed storage system work in the future. The authors show that the system scales well and is suitably efficient for real-time data services.
Paper [3] was motivated by developing a mashup based on users' explicit and implicit requirements. They emphasized the interactions between services are important for a more accurate recommendation. In order to improve recommendations, the author proposed a novel social-aware

service recommendation approach, where multi-dimensional social relationships among potential users, topics, mashups, and services are described by a coupled matrix model. Also they designed a factorization algorithm to predict unobserved relationships. They conducted experiments on programmableweb.com and showed the Data Density and the Weight of Factor have great impact on the recommendation performance.

Paper [4], the author first illustrates the drawback of current SQL to MapReduce translator which cannot operate in a one-operation-to-one-job mode and do not consider query correlations cannot generate high-performance MapReduce programs for certain queries, due to the mismatch between complex SQL structures and simple MapReduce framework. Then they introduce their solution, a system called YSmart, a correlation aware SQL-to-MapReduce translator. YSmart applies a set of rules to use the minimal number of MapReduce jobs to execute multiple correlated operations in a complex query. YSmart can significantly reduce redundant computations, I/O operations and network transfers compared to existing translators.

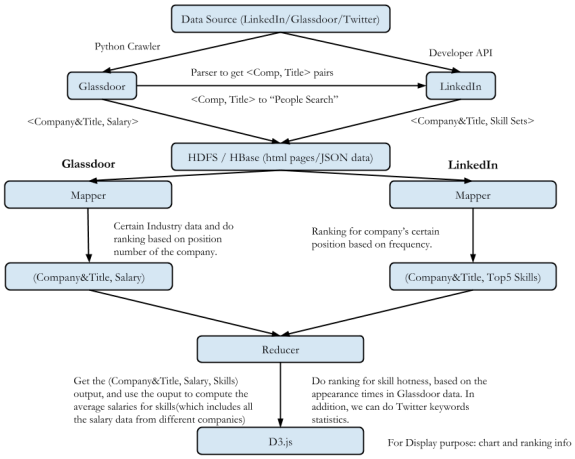
Paper [5] presents Shark, a new data analysis system that marries query processing with complex analytics on large clusters. The authors conducted experiments on four datasets and concludes that Shark significantly enhances a MapReduce-like runtime to efficiently run SQL, by using existing database techniques and a novel partial DAG execution technique that leverages fine-grained data statistics to dynamically reoptimize queries at run-time. This design enables Shark to approach the speedups reported for MPP databases over MapReduce, while providing support for machine learning algorithms, as well as mid-query fault tolerance across both SQL queries and machine learning computations. This research represents an important step towards a unified architecture for efficiently combining complex analytics and relational query processing.

Paper [6] introduces how hadoop and Hbase supports the new generation of applications arisen at Facebook that require very high write throughput and cheap and elastic storage, while simultaneously requiring low latency and disk efficient sequential and random read performance. In addition, the author presents specific improvements made to HDFS and HBase to enable them to scale to Facebook's workload and operational considerations and best practices around using these systems in production.

IV. Design

The following are the main project flow:

- Get JSON data from Glassdoor website including all companies information, and we extract company's name, industry type (which is used for more detailed analysis on different industries), titles and regarding salary ranges, to form the <Company & Title, Salary Range>
- Use the <Company, Title> pairs to make "People Search" through LinkedIn official development API to get people's profiles to stretch his skill sets, such as Java, C++, and HTML/JavaScript, to form the <Company & Title, skills> pairs. Inside each company's certain position, rank the top 5 skills based on the appearance frequency.
- Mapping the <Company & Title, Salary Range> and <Company & Title, Skills> to form the triple pairs <Company & Title, Skills, Salary Range>, and compute the average salary range for different skills and make the final ranks based on "salary/skill".
- Hotness ranks: we will also record the total appearance numbers for a certain skill in the whole industry.
- We will separate our analysis in industry base, because we think cross-industry makes not so many points for this project. So we will rank industries based on total company numbers of each industry from Glassdoor's data.



V. Results

(Future… In this section, you can describe: Your experimental setup/issues with data/performance/etc. Describe your experiments, describe what you learned. Did you prove or disprove your hypothesis? Were some results unexpected? Why?)

VI. Future Work

(Future… Given time, how would you expand your analytic? Could it be applied to other areas? Etc…)

VII. Conclusion

(Future… One or two paragraphs about the value/accuracy/goodness of your analytic.)

Acknowledgment

(This section is optional. It can be used to thank the people/companies/organizations who have made data

available to you, for example. You can list any HPC people who were particularly helpful, if you used the NYU HPC.)

REFERENCES

[1] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, Ion Stoica. Spark: Cluster Computing with Working Sets, 2010.

[2] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber. Bigtable: A Distributed Storage System for Structured Data, 2006.

[3] Jian Cao, Wenxing Xu, Liang Hu, Minglu Li. A Social-Aware Service Recommendation Approach for Mashup Creation, 2013.

[4] Rubao Lee, Tian Luo, Yin Huai, Fusheng Wang, Yongqiang He, Xiaodong Zhang. YSmart: Yet Another SQL-to-MapReduce Translator, 2011.

[5] Reynold S. Xin, Josh Rosen, Matei Zaharia, Michael J. Franklin, Scott Shenker, Ion Stoica. Shark: SQL and Rich Analytics at Scale, 2013.

[6] Dhruba Borthakur, Kannan Muthukkaruppan, Karthik Ranganathan, Samuel Rash, Joydeep, Sen Sarma, Nicolas Spiegelberg, Dmytro Molkov, Rodrigo Schmidt, Jonathan Gray, Hairong Kuang, Aravind Menon, Amitanand Aiyer. Apache Hadoop Goes Realtime at Facebook, 2011.