JOBVIZ – A VISUAL TOOL TO EXPLORE THE EMPLOYMENT DATA

By

JAY TAKLE

A thesis submitted to the

Graduate School – New Brunswick

Rutgers, the State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Deborah Silver

And approved by

_____

_____

_____

New Brunswick, New Jersey

May, 2015

ABSTRACT OF THE THESIS

JOBVIZ – A VISUAL TOOL TO EXPLORE THE EMPLOYMENT DATA

by JAY TAKLE

Thesis Director:

Deborah Silver

Job posting web sites are the most common way that students and professionals search for employment. All of the job searching tools operate in a similar way, users filter the postings through the use of keywords. Postings matching the keywords are displayed as a list and the user is required to read through this list one by one. However, there are many other ways to display text-based data. In this thesis, we explore the use of other types of information visualization techniques that would be appropriate for this type of data. We have developed a new visualization especially for student-based job searches and have applied our results to data from the Rutgers Career Knight Portal.

Acknowledgements

I would like to thank Dr. Silver for her expert guidance and support throughout my years in Vizlab. It was because of her expertise and creative mindset that I could work on such a diverse set of projects in Vizlab.

I would also like to thank Dr. Bemis and Dr. Tremaine at Vizlab; Katrin Heitmann, Tom Peterka and Salman Habib at Argonne National Laboratory; George Zagaris at Kitware; and Dr. Parashar, Dr. Jha, Dr. Marsic and Dr. Rona at Rutgers University. Collaborating with them on research as well as academic projects has helped me gain experience to work on different types of problems.

I would also like to thank my roommates, all my friends in the ECE department and the friends I have made over the years at Rutgers. It has been an experience to remember.

Lastly I would like to thank my parents Vishwanath Takle and Asha Takle, my sister Amruta Takle and all my relatives for their continuous support each and every day of my life.

Table of Contents

List of Figures

Introduction

To search for jobs in today's marketplace, many seekers utilize job search engines such as Indeed, Monster, Glassdoor, etc. These websites accumulate job listings from company career web-pages, job boards, etc. and display them as a search result. To initiate the search, the user has to enter a keyword related to their skill or a job title of interest. The search engine then gathers every relevant job posting and displays them as a list. Further filtering of jobs can be done using filtering criterion based on the individual's interest and skills. To know in detail about a job in the list the user has to open the job in a new window and read the whole description, requirements, qualifications, etc. If at all everything fits the user's qualifications and expectations, they can apply for the job. Otherwise they have to navigate back to the initial list of jobs and click on the next job to repeat the process.

Freshman and sophomore students who still have to decide their major do not have comprehensive resources at their disposal for making an informed decision. Many students also end up taking classes which are easy and popular instead of taking classes that add valuable skills to their resume. All students whether undergraduate or graduate can enrich their resume by knowing the exact skills required by employers. The most accurate way to find what the employers want is by scanning job listings and reading job descriptions, which again directs to list based job search engines.

The whole iterative procedure of searching job lists considerably slows down the decision making process for an individual. If we relate this to the 'focus and context' technique in the field of visualization, this way of visualizing job information only provides a low level focused view of the jobs to the user. When industry and labour department experts comment on the job scenario in today's world, they have to do an extensive study and crunch a lot of numbers to come to a

conclusion. The job seekers must also have this option of understanding the whole picture without the necessity of being a domain expert. They can then dive into individual employment opportunities through different levels and make their decision making process faster and easier. In order to attain this goal we have developed a system for providing a non-traditional view of jobs.

Through our system we first provide an overview of all the jobs available, thus putting the "job market" into context for the user. The user can then navigate intuitively and focus on jobs of interest depending on their major, class type or type of job. Our system can be used by graduating students who are searching for jobs as well as students who are pursuing college and want to know the skills they should learn for later job searching. In addition, course planners, faculty and department heads can utilize our system for structuring their courses and programs according to the skill demand of recruiting companies.

In the following chapters we will explain our system as follows: In Chapter 2, we will discuss related work and current methods used to visualize job postings. The job posting data visualized by our system was provided by Rutgers Career Services. In Chapter 3, we will explain in detail the data supplied to us and how we restructured the data for various information visualization functionalities in our system. One of the most important challenges in information visualization is utilizing real estate on the visual canvas effectively. In Chapter 4, we will talk about the different data layout approaches for effective use of the viewer's screen and for conveying information to the user quickly. The various interactive features of our system are elaborated in Chapter 5. In Chapter 6, we discuss the benefit of our system for different kinds of users in academia. We conclude with a summary of the system and future work in Chapter 7.

Related Work

In the field of employment and human resources, information presented in the most simple form is a job posting. It has a title, company name, hours of work, geographic location, job description, etc. with “keywords” or “tags” that are different for different jobs. When there are hundreds of such jobs, they are presented as a list. From a job seeker’s point of view this is the only way job data is generally shown. Domain professionals on the other hand have used various information visualization techniques for analyzing job data. But most of these techniques have been used for getting an overview of the job scenario. They use it for explaining reports, accompanying studies and for making decisions through such visualizations. In this section we discuss different job sites and the various visualizations related to the work in this thesis.

2.1 Job Searching Tools

2.1.1 Career Knight

Career Knight is a job portal maintained by Simplicity.com for the career services department at Rutgers University. It provides students an interface to search through jobs posted by various companies for Rutgers students. Mostly companies geographically located near Rutgers post their jobs in the system. Students can then search through the system depending on their major, class level etc. and apply to jobs which meet both the company’s and the student’s criteria. Students can also search for jobs by specifying keywords or tags such as skills, geographical location, etc.

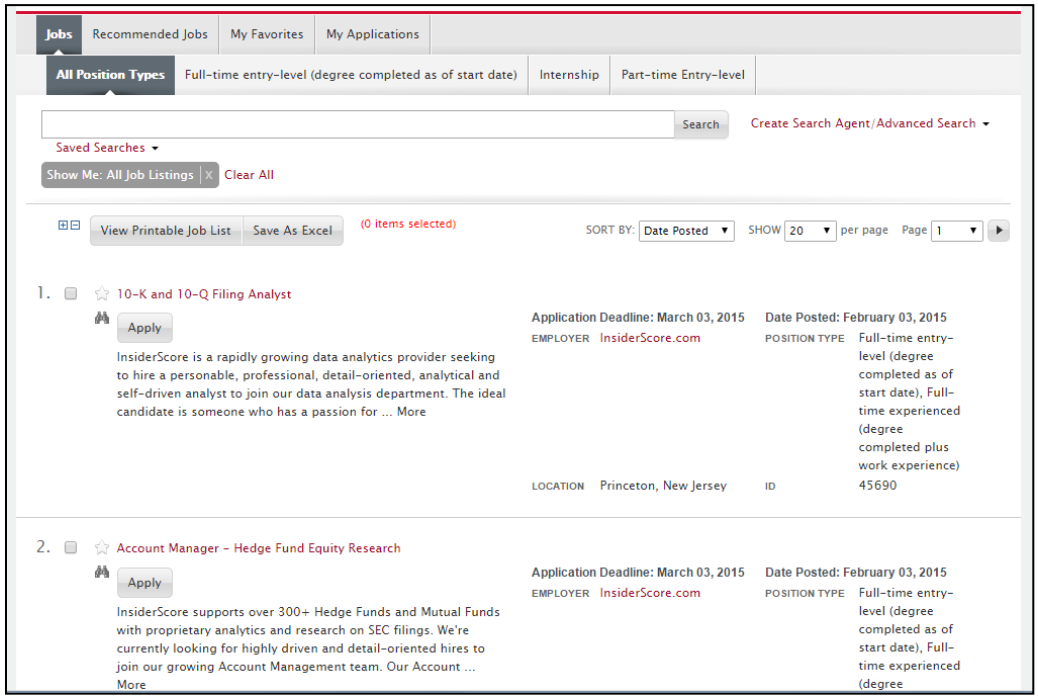

Figure 2.1.1: The job search page on Career Knight shows two jobs from a list. There are also tabs for recommended jobs (picked by Career Knight for the user), my favorites (saved by user) and my applications (applied by user) [1].

The Career Knight page for job search shows a list of all available jobs in the system. The users have an option to filter out unrelated jobs. Also present is a tab for recommended jobs. These are jobs picked by Career Knight which match your profile. Career Knight would compare the user's major, class level and other information with the jobs and show the ones that match. The number of jobs for a specific major could be very limited since these are jobs from locations near Rutgers and only from a small number of employers.

2.1.2 Major Internet Search Engines

A number of employment related meta-search engines have been deployed on the internet to help job seekers search job postings uploaded by employers. These websites accumulate job listings

from company career pages, job boards, etc. and display them as a search result. The major job sites are Indeed, Glassdoor, Monster, CareerBuilder, Simply Hired, snagajob, usajobs.gov etc. We will discuss a few of these and their features in this section. Figure 2.1.2 shows the main page of one such website Indeed.com.

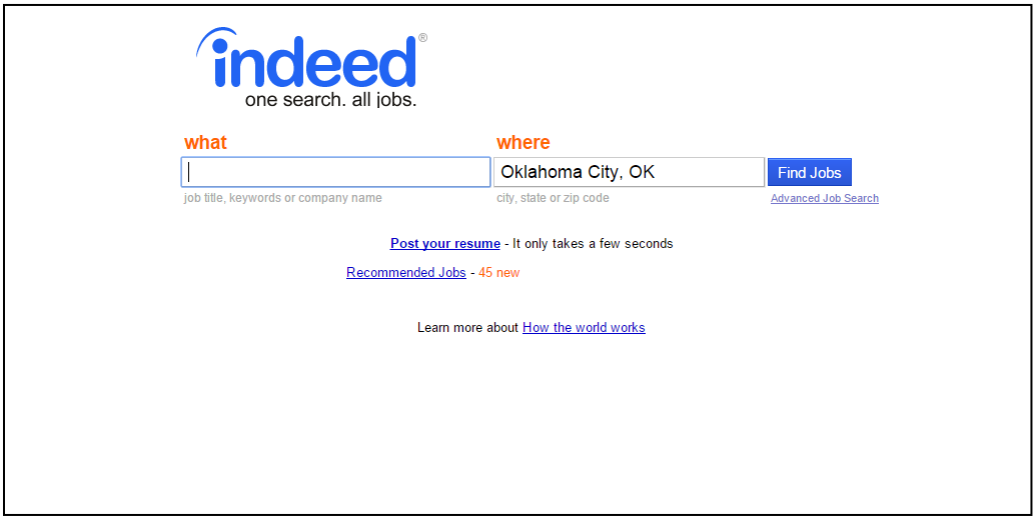

Figure 2.1.2: The home page on Indeed.com expects a user to enter 'what' they are searching for (job title, keyword, company) and 'where' (location) [2].

Users have to enter a keyword related to their skill or a job title of interest on the main page. The search engine then uses web crawlers to search for job postings over the internet which include the specified keyword. All relevant jobs are gathered by the search engine and displayed in the form of a list on the screen as shown in figure 2.1.3. Users can dig deeper by focusing on more specific jobs. Further filtering of jobs can be done by selecting qualifications and/or categories such as salary, region etc. After a satisfactory filtering of jobs, the user can read in detail about each job in the list. By clicking on the individual jobs, users are forwarded to the original webpage. This can either be the company's career website or a job posting website.

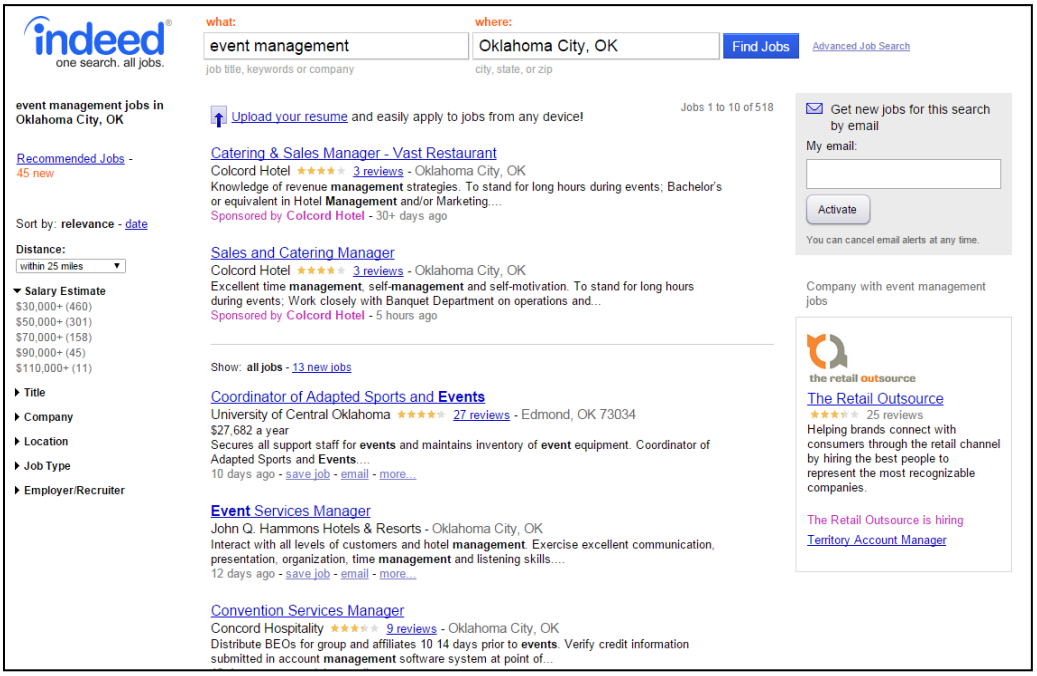

Figure 2.1.3: This shows the results page on Indeed.com after a user enters their field of interest. As shown, jobs are listed in a linear fashion one below the other [2].

This process has to be repeated for all jobs in the list. This technique of displaying jobs has been traditionally used by all employment websites. But due to the iterative 'to and fro' nature of navigation, a limit on number of jobs that can be viewed at a time, the need to read the whole description of each job posting this is a time consuming process.

Glassdoor is a job search engine as well as a review site, better known for the latter. Like Indeed and other search engines users can seek jobs from around the internet. But Glassdoor added an additional feature, company reviews. Here current and former employees anonymously review companies and its management. They also provide salaries for different job positions which can help new employees bargain their salary and also an idea about the salary trend in the industry.

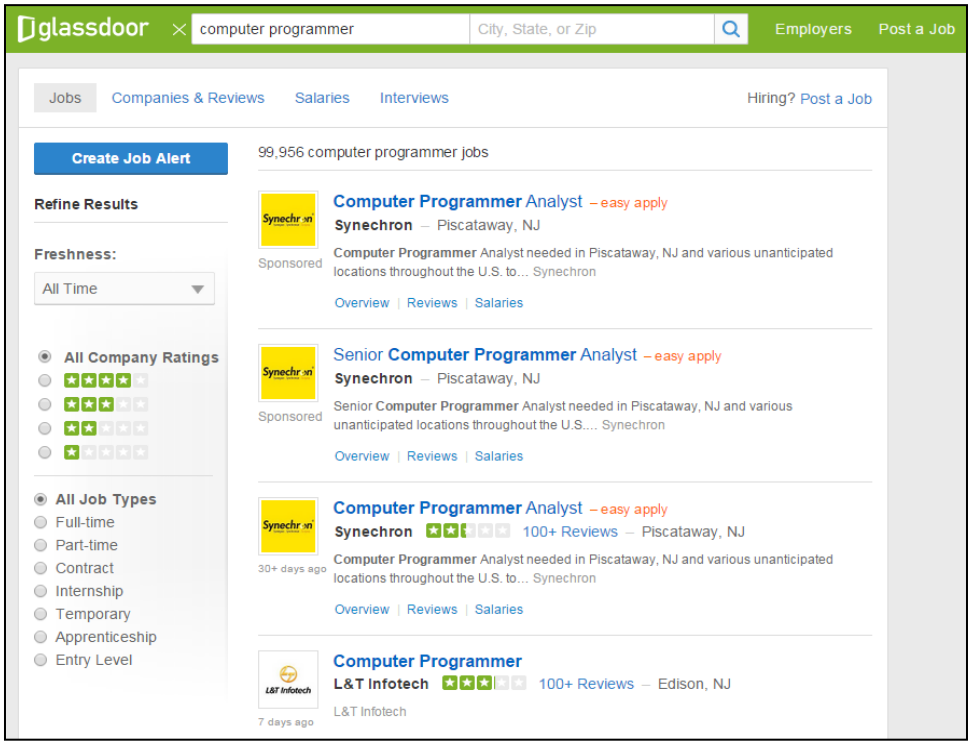

Figure 2.1.4: The interface of Glassdoor is similar to Indeed and other search engines. But we can see extra tabs such as ‘companies and review’, ‘salaries’ and ‘interviews’ [3].

Figure 2.1.4 shows that the Glassdoor interface is very similar to the other job search engines in a lot of aspects. There are ways to sort the jobs, filtering can be done based on type of job, geographical location, etc. But three more tabs on the top give more information about jobs. The ‘companies and review’ tab would give an insider’s outlook as the work ethics in the company. Job seekers always want to know if the company is ideal for them. Other than that the salary section gives a job seeker a good idea about the salary trend. This is important especially for experienced candidates who need to bargain their salary for the talent and experience they would bring to the company.

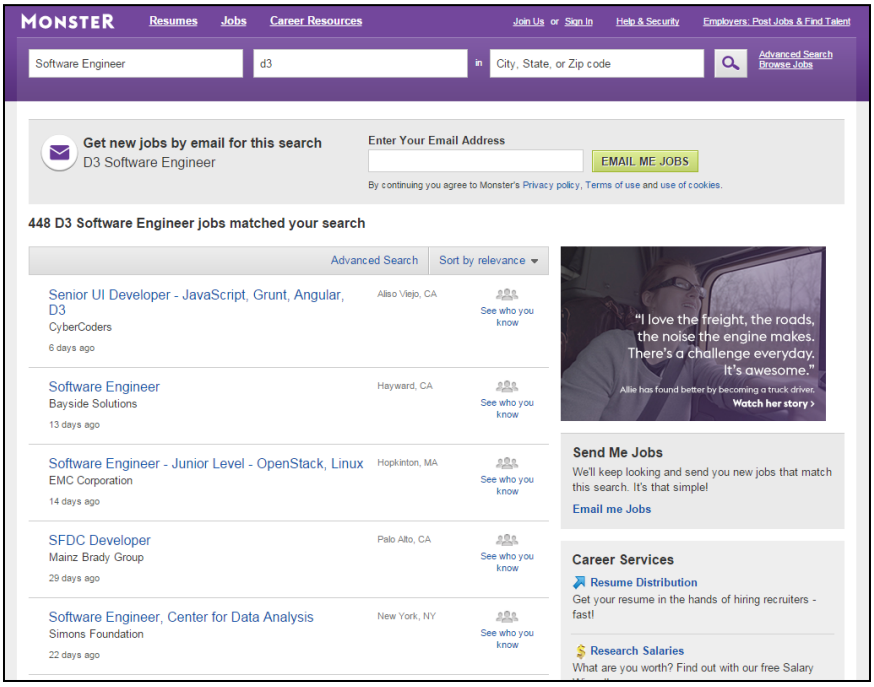

Figure 2.1.5: The user interface of Monster.com shows options to upload resumes and get email alerts [18].

The other well known job search engine is Monster. Along with the main functionality of displaying job postings according to user entered keywords, it also allows users to upload their resume. One other option they have is sending users alerts when a job with a specific keyword is posted on Monster.com. Figure 2.1.5 shows a snapshot of Monster's user interface. Some companies provide information about real-time job market compensation such as salary, benefits, etc. PayScale is one such company. The federal government also has a website called "usajobs.gov" for users to search for jobs in different government institutes.

2.2 Information Visualization

There are a number of information visualization techniques that are used by researchers in the employment sector to present their statistics and results. We will discuss some of these visualizations in the next few sections.

2.2.1 Tag Cloud

A tag cloud is a visual representation of keywords in a text, when the goal is to provide an overview of important words. Keywords or important words from a text are placed in different orientations, thus forming a cloud. The font size of each word indicated the importance of that word. Tag clouds are used prominently for visualizing the frequency of most used words on websites and blogs.

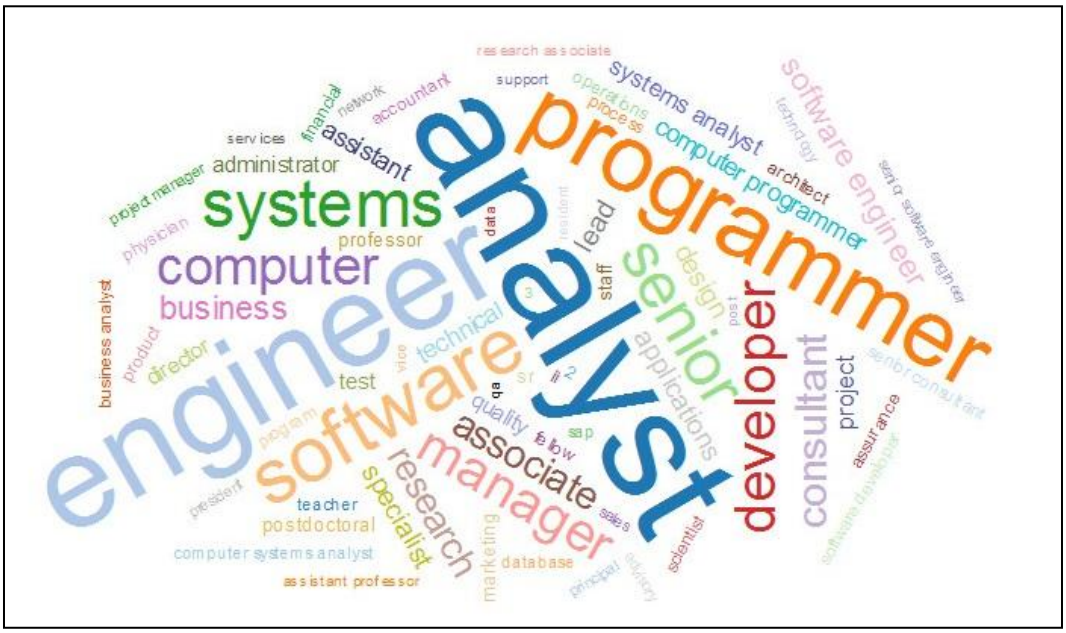

Figure 2.2.1: A tag cloud rendered to show the salaries for different job positions for H1-B visa holders. Font size of each word reflects the relative salary [4].

Some experimental visualizations have used tag clouds to display job titles as tags. Some use them to indicate the salary for different positions in different companies as shown in figure 2.2.1. The font size of each job title indicates the relative salary that job offers amongst all others. This can help job seekers get an overview of relative differences between positions. But most of the times tag clouds have been used as a visual aid to help explain an article related to employment as shown in figure 2.2.2.

Figure 2.2.2: The green words on the left half of the cloud give an idea of skills and industry sectors in California, while the red words on the right hand side show what professionals can expect in New York [5].

In the above figure two technology hubs were compared using a tag cloud. The green words on the left half of the cloud give an idea of prominent skills and industry sectors in California. While the red words on the right hand side show what professionals can expect in New York.

2.2.2 TreeMap

Treemaps are used to display hierarchical data in the form of nested rectangles. It is like looking at a tree structure from the top with several branches and sub-branches. Treemaps are used to reveal patterns which wouldn't be visible in other forms of visualization. Color coding and sizing of the rectangles are very important for studying and analyzing the patterns.

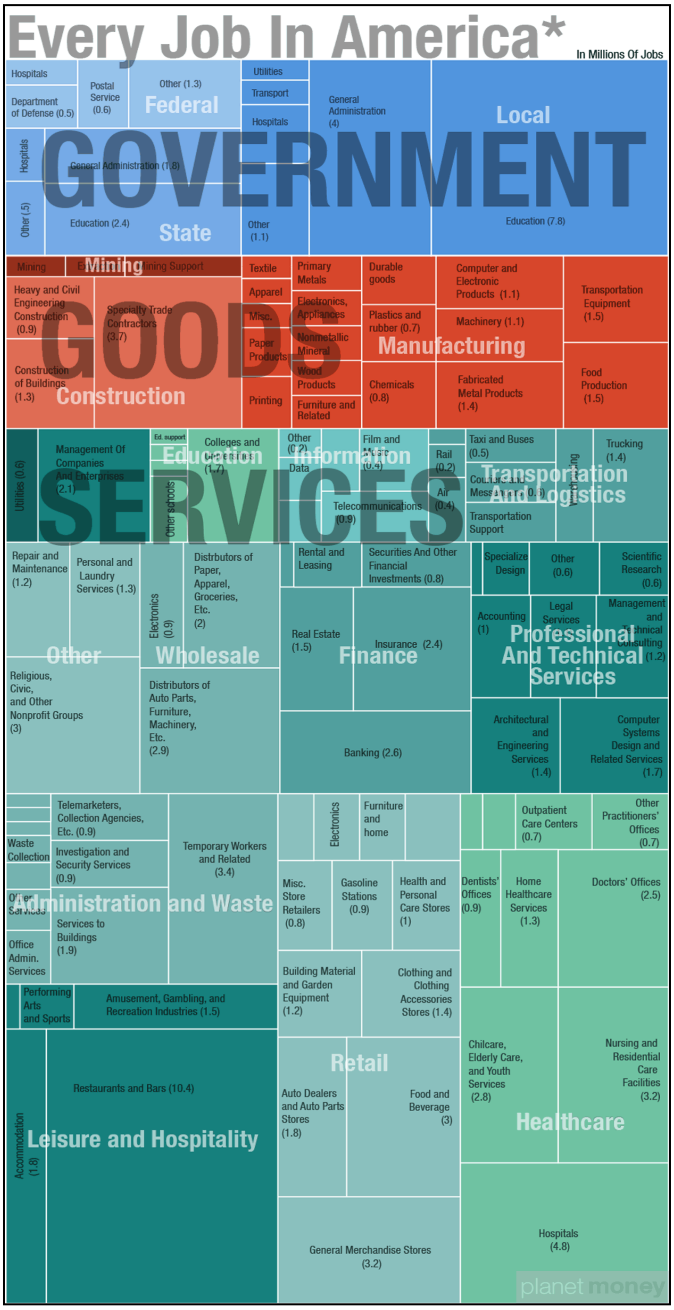

Figure 2.2.3: This treemap was plotted to give an overview of all the jobs in the United States for one month. Here three sectors have been branched and further divisions show one more level of detail in the industries. Numbers in parenthesis indicate jobs in millions [6].

The usage of treemaps has been limited to act as supplementary visualizations for news articles on jobs distribution in a country or state. An example as shown in figure 2.2.3 gives an overview

of the job market in the United States for one specific month. Here the job market is divided into three branches, viz. government, goods and services. Each branch is further divided to give one more level of detail. Numbers in parenthesis show the number of jobs in millions.

2.2.3 Bubble Chart

In general usage bubble charts are applied to a data with three parameters, viz. the X and Y co-ordinates plotted on the axes and the size of the disk for each data point. A bubble chart is essentially a scatter plot with an added dimension of radius of the data point. Additional dimensions can be added by carefully applying a color scale and labeling each disk as shown in figure 2.2.4. Here we see the relationship between the life expectancy and GDP per person of various countries. The size of each circle indicates the country's population while the color is coded by geographical regions.

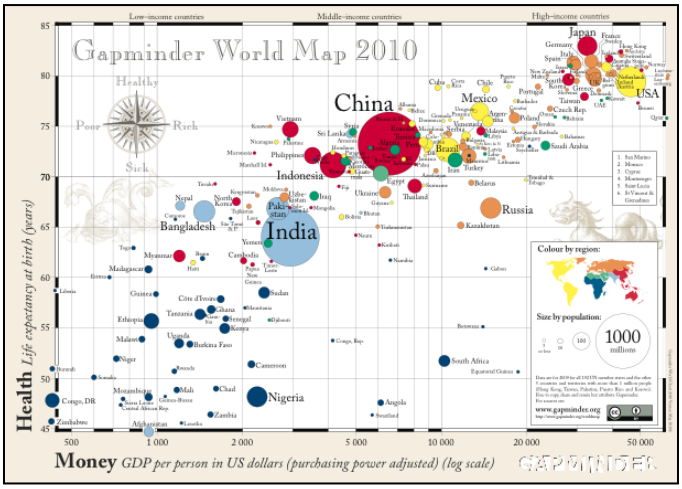

Figure 2.2.4: This bubble chart shows the relation between health (Y-axis) and wealth (X-axis) of different countries. The bubble size indicates population while the color indicates the region [7].

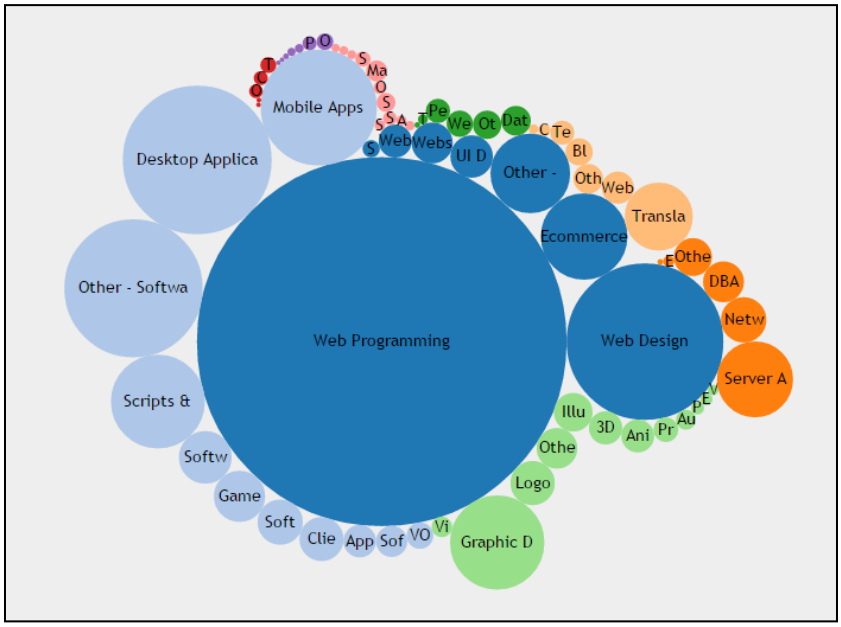

Figure 2.2.5: A bubble chart showing the distribution of man power in the different sectors of web development in Russia [8].

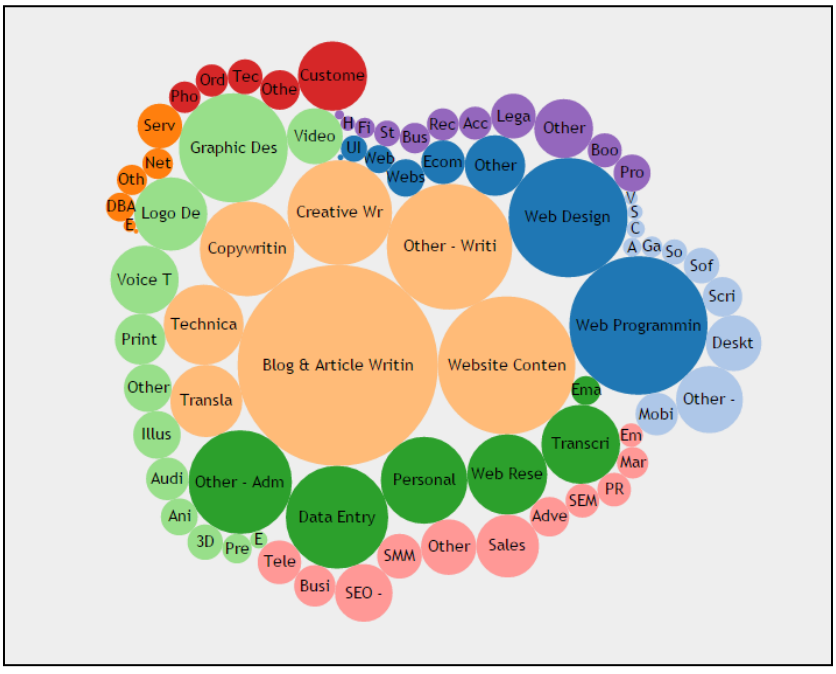

Figure 2.2.6: A bubble chart showing the distribution of man power in the different sectors of web development in the United States [8].

For the purpose of information visualization in the field of employment, bubble charts are also used in a similar way as treemaps as shown in figure 2.2.5 and figure 2.2.6. In such a configuration the aim is to pack the bubbles as closely as possible, thus the X and Y co-ordinates are rendered meaningless.

The above figures were used to compare the job distribution in the Russia (figure 2.2.5) and the United States (figure 2.2.6). Comparing the two charts blog authors provided some conclusions. Such as how the blue colored bubbles indicate that web development is more prominent in Russia with more man power in the web programming sector. In the United States on the other hand the online writing industry takes a larger share with a fairly even distribution in its sub-sectors. Again this information visualization technique was used to explain and comment on the present job scenario.

2.2.4 College Major and Occupation Contingency Wheel

Categorical data such as job listings can be analyzed in a number of ways. One of the widely used methods is to view the data in the form of a contingency table. A contingency table shows relationships between different categories. Alsallakh[9] introduced the contingency wheel, a technique for visual analysis of large contingency tables. The categories are plotted on a ring chart in the form of sectors. A relationship between different categories is shown with the help of a chord connecting those two sectors.

The United States Census Bureau used the contingency wheel to show the flow of college graduates from college to the industry [10]. The wheel is first divided into two equal segments. One segment shows the different majors offered in colleges. The other segment represents different types of occupations taken up by the college graduates after finishing college. Both

segments are further divided into individual college majors and occupations. The length of each segment represents the number of people. Relation or flow of students from a college major to an occupation is represented by a chord. The thickness of the chord indicates the share of people in that major-occupation relation.

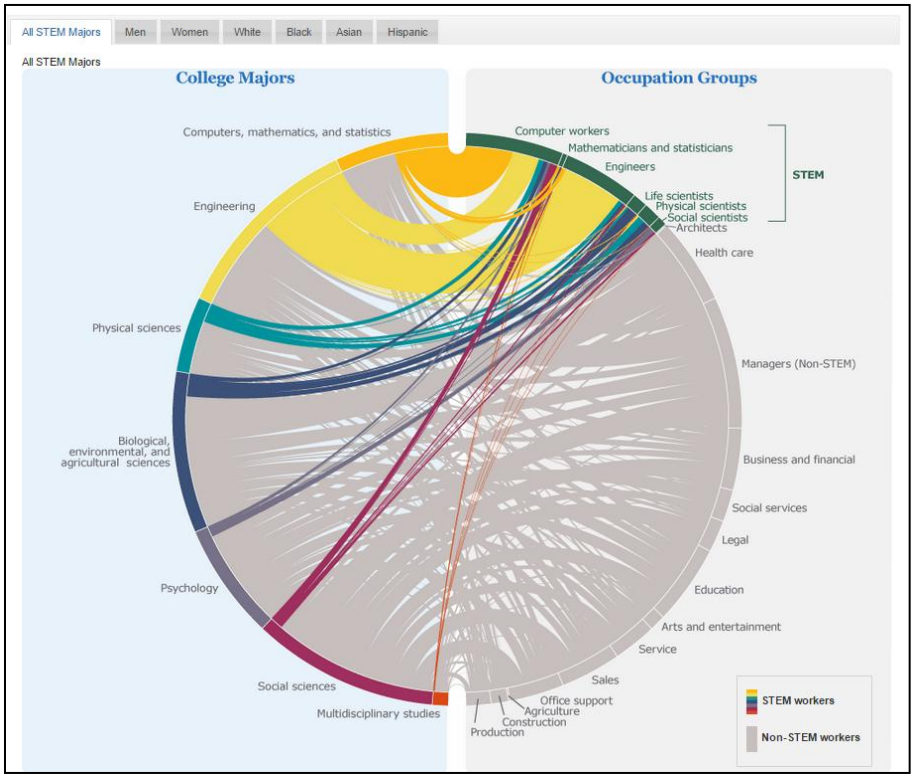

Figure 2.2.7: This contingency wheel shows the flow of college graduates to the industry. One half of the wheel is categorized into segments for college majors while the other half is divided into various occupations [10].

By hovering the mouse over a college major two types of chords are highlighted. Chords highlighted in color of the major's segment show the flow of students from a major to an occupation related to that major. While chords highlighted in grey color show the flow of students to an occupation not related to the major. For reverse analysis users can hover on the occupation of their choice. This leads to highlighting chords in color that show the inflow of students from various college majors. In order to study the trend for different ethnicities and sexes, these

contingency wheels have been plotted for each of the different groups, viz. Men, Women, White, Black, Hispanic and Asian. The US Census Bureau used the contingency wheel to provide an overview for studying the trends in the college major to occupation flow.

2.2.5 TextArc

TextArc[19] is a tool based on Java designed to help users discover patterns in text. It tries to show the writing style of a document by highlighting words as they would be placed in the text. As shown in the figure 2.2.8, all the text is shown in form of two concentric spirals. There are also words inside the spiral. Words inside the spiral are placed in a position that is nearer to the chapter where it was used.

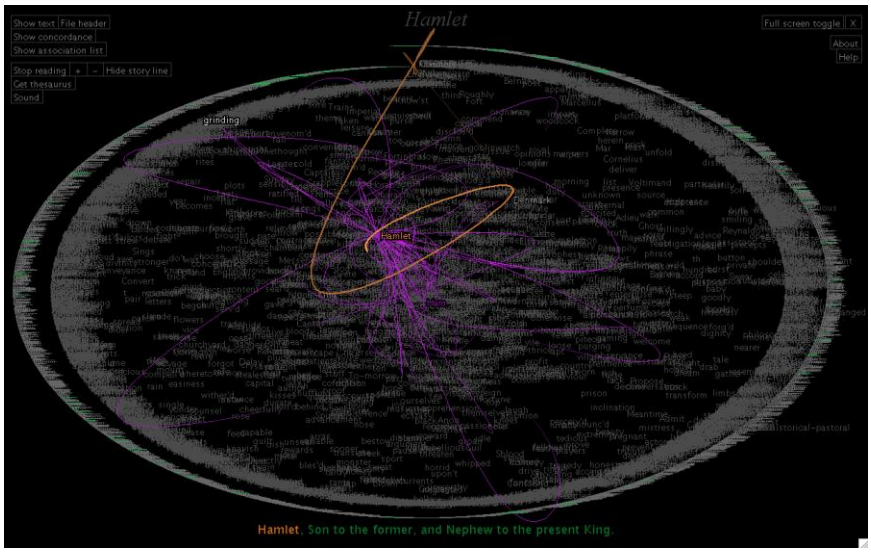

Figure 2.2.8: This textarc shows words from the book “Hamlet”. The sentence at the bottom shows where the word is used and the colored links/lines show with which other words the word ‘hamlet’ was used.

When a word is clicked or highlighted, textarc shows links to indicate with which other word the highlighted word was used.

2.2.6 Visualization Software

All the above mentioned visualization techniques, viz. Tag Cloud, Treemap, Bubble Chart and Contingency Wheel can be rendered using a wide variety of simple or complex software. The data used for plotting these would be extracted from the original job listings mentioned in section 2.1. But that kind of data crawling and acquisition can take time. Thus most of the examples we have shown in the above sections use data from sources like the US Census Bureau and the Labor Department. Their data is a higher level quantitative data as far as details are concerned, since the goal is to just provide an overview of the scenario.

There are a number of software such as Tableau, Spotfire, Circos, etc. used for plotting data in the form of various charts [11][12][13]. Most of them have to be bought from the company and are pretty expensive. Although they do not require any programming experience a basic knowhow of their working is necessary. These software can draw the abovementioned plots individually or as part of a dashboard but there is no flexibility in changing the arrangement of the data and the visualizations. The other point from a data perspective is that they are mainly developed for visualizing tabular numerical data.

2.2.7 Visualization Toolkits and Libraries

For programmers who want to develop custom made charts, there are open source toolkits and libraries. Developers can design and build simple charts to complicated dashboards using them. Toolkits like Piccolo, Prefuse, ivtk, can be used to build stand alone Java applications [14][15][20]. These applications are custom made for a specific type of data with functionalities as per client needs. Being open source these toolkits can have some glitches and may require extensive testing.

A widely used library for developing information visualization frameworks on the web is D3 [16]. D3 (Data Driven Documents) can be used with JavaScript to visualize different types of charts on the web through SVG (Scalable Vector Graphics). We have used D3 to develop our system in an endeavor to provide a ‘Focus and Context’ view of the job listings. We will talk in detail about D3 in chapter 4.

In the following chapter we will talk about our motivation for building the visualization system for employment data.

Motivation and Goal

All the job searching tools we mentioned in the earlier chapter layout information in the traditional way, a list of jobs with multiple pages to fit them all. This forces users to view information in a linear fashion making it an exhausting process. Users are made to directly focus on detailed information without the option to get an overview of the job market. Basically the information is not visualized effectively. The job statistics visualizations on the other hand are only used as supplementary visual aid for reports and articles. Some of them are interactive but only to the limit of masking and unmasking information. There is no system that combines a job search tool with information visualization techniques to provide a tool for exploring information. We plan to bridge this gap by designing a system that can allow users to explore information at different levels of detail, allow them to ask questions through interaction and most importantly present the information through effective and concise visualizations.

The aim of developing this system is to help students, faculty and human resources domain experts. Since the user profile covers a wide variety, we would like to describe how each type of user can take advantage of the system.

3.1 For Newly Enrolled Students

Freshmen, junior and even first year graduate students sometimes take classes which they think would be beneficial for finding jobs. It is only later during job applications or interviews that they realize they should have opted for some other class. We want these newly enrolled students to take informed decisions about class selection by using our system.

Job postings contain the exact information on skill sets expected from students coming out of colleges. At their stage in college students would not take interest to read through paragraphs and paragraphs of job descriptions. This is where our system can help.

Show skill keywords for each major: Students can use our system to know the skills most wanted by companies. They can then select classes most related to those skills.

Show job scenario according to major: One of the other big problems faced by new students is deciding a major. Through our system students will have a good overview of the job market according to majors and they can use it as a real world supplemental factor while deciding their major. In addition to above, current students can also use the system to search for internships and other part time jobs faster.

3.2 For Graduating Students

Graduating students start the job search process pretty early. With the aim of having a job ready when they graduate, students apply to as many jobs as they can. Some students only focus on campus recruitment, but most of them use the different online job searching engines. As we have mentioned earlier these search engines present a list of jobs from the internet which match the keywords entered by a user. The students would generally type a job title they might have heard from people in their major. And then they would go over a list of jobs laid out by the search engines. Students who are not very patient with long lists would try to filter as many jobs as they can by using the search engine's options. In either case they would have to read each job separately and come back to the list, making it an exhausting process.

The other problem here is that the students do not know what they are searching for. By entering words they have heard around, they hope that at least a few jobs pop up which might be remotely related to what they want to do. By typing specific keywords they unknowingly filter out jobs that

might be related to their major but do not contain the keywords. It is important to first get a view of what is available out there and then start focusing on individual jobs.

Show all jobs: Through our system we want to first show students all the options and then let them explore through different entry points. Our aim will be to put out all the information without making it overwhelming for the students. We want to help the students make a mental picture of the job market, so that their decision to look for specific jobs is more informed.

Show jobs according to major, class level or position type: The students can then filter jobs according to parameters like their class level, so that only jobs pertaining to their class would be highlighted. Or they could be looking for temporary or part-time jobs. So they could select that position type.

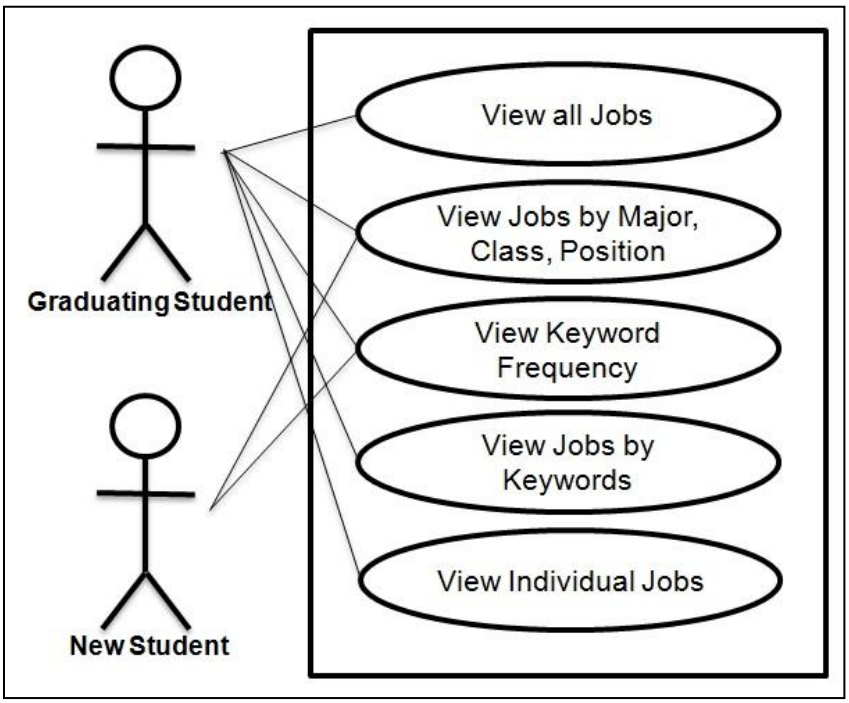

Figure 3.1: This use case diagram shows the various ways graduating students as well as newly enrolled students can use and benefit from the system.

Show keywords frequency for each major and jobs for each keyword: We also want to make the job searching process fast for the students, so that their hit rate is higher. And that is why we

would like to focus our system on processing the part of a job posting that takes the most time to read, the job description. Through our system we would like to show students the technical skills companies are looking for, without them having to read each and every description.

3.3 For Faculty and Administrators

Job descriptions are rich with information pertaining to skills sought by companies. Basically if we look at job descriptions with a cumulative description, they show us the skills demanded by the companies. Recruiting companies do not always get candidates possessing all the skills, thus they end up spending a few of the early months training the new recruits.

Show skill keywords for each major: The best way for an institute to increase their recruitment figures is by providing qualified candidates with more of the skills required by companies. The faculty and administrators can use our system to analyze the skills companies want a candidate from a specific major to possess. They can then restructure courses, upgrade current software taught, etc to meet the demand in the industry.

3.4 For Domain Experts

We see a lot of articles on employment and skilled labor in magazines and news papers. Experts in the field have to do a lot of number crunching in order to get decisive results to comment on. Just about everyone uses the same statistical methods and techniques for researching in this field. Domain experts can use our system to get a new direction of looking at labor data. It can help them uncover more facts to supplement their studies.

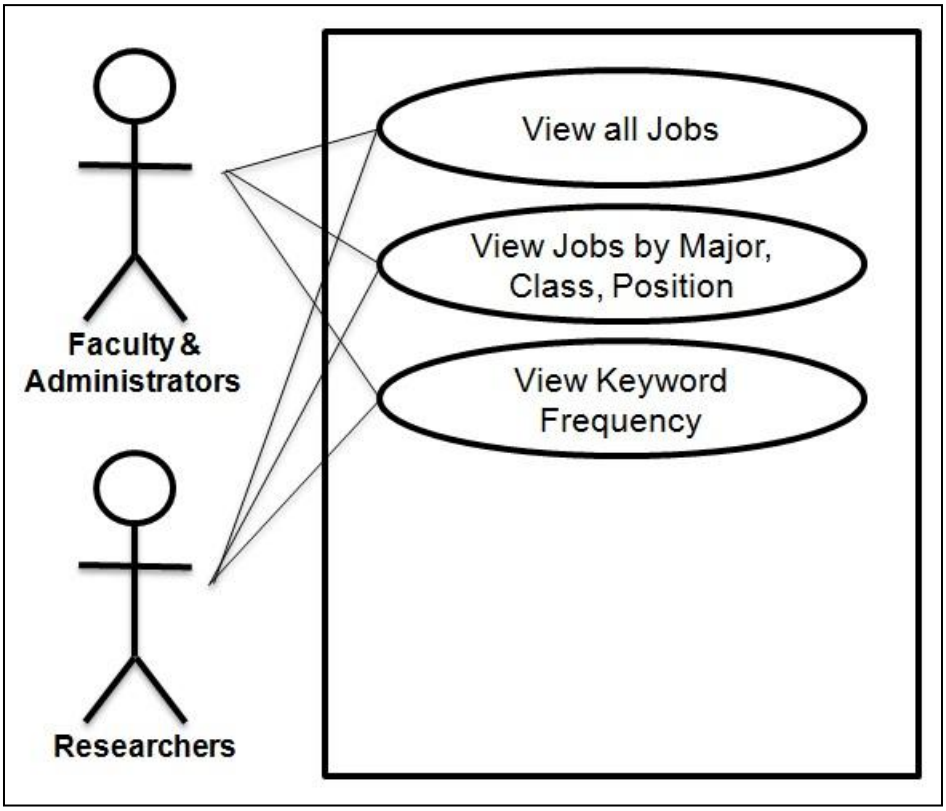

Figure 3.2: This use case diagram shows the different ways faculty, administrators and researchers in the employment field can use the system for learning more about the data first hand.

Show jobs according to major, class level and position type: They can study the job market and employment opportunities according to the academic majors, class level of candidates sought by companies and even by the different positions available.

Show keywords from job descriptions for each major: As we already mentioned the importance of information rich job descriptions, researchers can analyze them through our system. Through our system they can study a sizeable number of job descriptions and build relations in the data.

3.5 Design Goals

The user profile of our system can vary from graduating students who are searching for jobs to students who are pursuing college and want to know the skills they should learn for later job searching. In addition, course planners, faculty and department heads can utilize our system for structuring their courses and programs according to the skills demanded by recruiting companies. And labor department researchers can use our system for analyzing the job market. In order to cover such a wide range of users our design goals are as follows:

1. Display data through a concise visualization.
2. Communicate efficiently through minimum text.
3. Provide a ‘focus + context’ view.
4. Keep interaction with the visualization intuitive.
5. Show relationships between parameters.
6. Avoid any installation or setup.
7. Provide global availability.

In the next chapter we will talk about the technical platform we chose to design our system in order to attain our goals. We will also discuss the data, the visualization layout approach and the data processing.

Data Driven Documents (D3)

In our motivation and goals we elaborated how we want our system to be useful for every kind of user. Through our system we want to show the job posting data in a way that would help the users discover more facts and make them explore the data more. In order to design a system that can have concise visualizations with a potential for knowledge discovery, we decided to use D3.js (Data Driven Documents), a JavaScript library for data visualization. Our visualization system would be a web based application, so that maximum number of users can explore the features. In this chapter we will introduce D3, discuss some of its functionalities and explain how it works with data.

4.1 The role of D3 in web applications

D3.js is a JavaScript library for visualizing data using HTML, SVG and CSS. Before going deep into D3 let us understand how it fits into the world of web based applications. HyperText Markup Language (HTML) is a standard for structuring web browsers. It is the basis for creating web pages. A browser does not show the HTML code but uses it to display the content or data. HTML codes can be used to display text, images and interactive objects. A simple HTML code is shown in figure 4.1

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

Figure 4.1: A sample HTML code with basic elements. This code prints ‘Hello world!’ on a webpage [17].

HTML can also embed scripts which are used to automate execution of tasks. JavaScript is one such scripting language. It allows control of the browser to communicate asynchronously and change the content to be displayed. Our framework uses JavaScript extensively along with D3 for data processing and for interacting with the visualization. As shown in the figure 4.2, an HTML page calls a JavaScript file which runs a number of operations like any computer programming language.

```
    border: 0px;
    border-radius: 8px;
    pointer-events: none;
  }

</style>

<script src="http://d3js.org/d3.v3.min.js"></script>
<script src="http://coewww.rutgers.edu/www2/vizlab/careerSrv/wordCloud/d3.layout.cloud.js"></script>
<script src="http://coewww.rutgers.edu/www2/vizlab/careerSrv/wordCloud/fisheye.js"></script>
<body>
    <div id="body"></div>
    <script type="text/javascript" src="cloudWheel_1.js"></script>
  </body>
</html>
```

Figure 4.2: This snippet of HTML code shows JavaScript files being specified as source files and a JavaScript file called in the body.

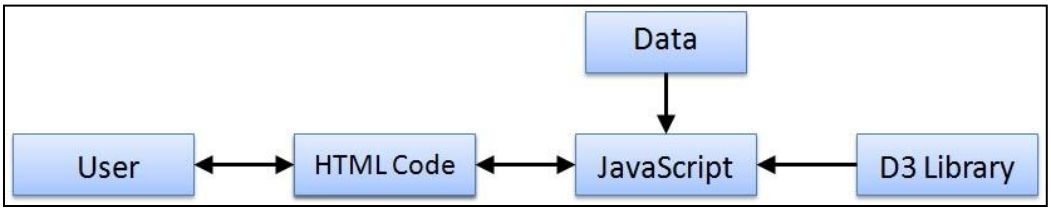

Figure 4.3: This schematic diagram shows the role of D3. The user loads up the HTML code. The HTML code in turn calls the JavaScript. The JavaScript loads the data and calls D3 functions to process data and send back visualization to the HTML code.

For visualizing data in the form of rich graphics, the JavaScript calls various D3 functions. D3 uses Scalable Vector Graphics (SVG) for visualizing the data. SVG is a vector image format for

two-dimensional graphics that also supports interactivity. Figure 4.3 shows the role of D3 in a web application.

D3 allows binding the data to a Document Object Model (DOM), which can then be manipulated by data-driven transformations. For e.g. a 2D array of numbers can be visualized as a table or can also be visualized as a bar chart. D3 has a collection of components that can be reused, modified and interlocked giving a lot of operational flexibility. In the next section we discuss the main components of a complete D3 based JavaScript code.

4.2 D3 API

Since our visualization framework is web based and highly interactive, we have extensively utilized JavaScript along with D3.js. D3 has a lot of functions that can be used for developing simple as well as complex visualizations. They can be used for reading the data, manipulation, visualization and interaction. In this section we will go over some of the important functionalities available in D3.

4.2.1 Selections

An array of elements or a single element can be pulled from a document as a selection. Selections can be done by class, tag, attribute or identifiers. D3 provides two methods, d3.select(selector) and d3.selectAll(selector). They can be used to accept either selector strings or nodes. Once elements are selected we can apply operators to them and manage them. We can use them to get and set attributes, set styles and even text content.

selection.**attr**(*name*[, *value*])

Multiple operators can be chained together for a compact code. The value of the attribute can be either specified as a constant or a function. Specifying the value as a function can be useful when

a different value has to be assigned to each element. The function is evaluated for each selected element and the function's return value is used to set the attribute.

New elements can also be appended to the current selection using:

selection.**append**(*name*)

Each new element inherits data of the current element. The name can be either a string or a function that returns the DOM (Document Object Model) element to append. Selections can also be joined to data for data driven transformations using:

selection.**data**([*values]*)

Again, the values can either be an array or a function that returns an array. The data is available on re-selection. Binding data allows for adding and removing nodes in correspondence with the data.

D3 has event listeners that can be applied to selections. Event type names such as "click", "mouseover", etc can be used.

selection.**on**(*type*[, *listener*[, *capture*]])

Multiple listeners can also be used for the same event name using namespace. Events taking place can also be stored using:

d3.event

For example, the amount of mouse translation can be stored for a function such as zooming. This can be attained in combination with the mouse operator:

d3.mouse(container)

This returns the X and Y co-ordinates of the current event, relative to the specified container which could be a SVG or HTML container.

4.2.2 Transitions

When we need to apply an operator smoothly instead of a sudden change, we use the transition operator on a selection. Transitions can also be made on a per element basis using a function. Transitions have a default delay of 250ms but can be specified too.

transition.delay([delay])

The delay can be set using a function so that a staggered transition over all the elements can be attained. D3 also has some interpolators for transitioning between arbitrary values. A few examples are:

d3.interpolate(a,b) – interpolate two values.

d3.interpolateRgb(a,b) – interpolate two RGB values.

d3.interpolateArray(a,b) – interpolate two arrays of values

Custom interpolators can also be built using the tween operators, such as attrTween and styleTween. When a certain function need to be invoked as a certain time custom timers can be set using:

d3.**timer**(*function*[, *delay*[, *time*]])

This invokes the function till it returns true. A calendar notification can also be coded in this operator by setting the time parameter.

4.2.3 Processing arrays

D3's accepted representation of data is in the form of arrays. Thus D3 has a lot of array manipulation functionalities on top of the already existing JavaScript array methods. Arrays can be ordered using the d3.ascending() and d3.descending() operators. Quite a lot of operators are available to get statistical information about an array. A few of them are:

d3.min() – find the minimum value in an array.

d3.max() – find the maximum value in an array.

d3.extent() – find the minimum and maximum value in an array.

d3.sum() – compute the sum of an array of numbers.

d3.median() – compute the median of an array of numbers.

d3.variance() – compute variance of an array of numbers.

JavaScript works with objects, which has a set of named properties. D3 provides operators to convert these objects to standard arrays.

d3.keys(object) – returns an array containing property names of the object.

d3.values(object) – returns an array containing property values of the object.

d3.entries(object) – returns an array containing property keys and values of the object.

Some of the array manipulation operators built into D3 are:

d3.merge() – merge multiple arrays into one array.

d3.range() – generate a range of numeric values.

d3.permute() – reorder an array of elements according to an array of indexes.

d3.zip() – transpose a variable number of arrays.

d3.transpose() – transpose an array of arrays.

d3.pairs() – returns an array of adjacent pairs of elements.

Adding these to the already available array operators in JavaScript makes D3 powerful for processing arrays.

4.2.4 Loading external files

Very small data can be directly hardcoded into JavaScript but for larger datasets D3 provides functions specific to the type of file from which it is being loaded. Following are some of the operators:

d3.text() – request a text file.

d3.json() – request a JSON blob.

d3.xml() – request an XML document fragment.

d3.html() – request an HTML document fragment.

d3.csv() – request a comma separated values (CSV) file.

d3.tsv() - request a tab-separated (TSV) file.

4.2.5 Scales

Scales help us apply a specific type of mapping from input to output. D3 provides functions for continuous domain quantitative scales, discrete domain ordinal scales and also time scales. In quantitative scales D3 has options for linear scales, identity scales, power scales, log scales, quantize scales, quantile scales and threshold scales. Identity scales are linear scales where the scale and its invert method are identity functions. Power scales as the name suggests has an exponential transform applied to the input. In a similar way for log scales a logarithmic transform is applied to the input values. The ordinal scale in D3 is mostly used for colors. Fd3.scale.category10() for example constructs a new ordinal scale with a range of ten categorical colors. The time scale in D3 uses JavaScript ‘Date’ objects as domain representation. Domain values are based on dates.

In the next chapter we will talk about the data from Rutgers Career Services and our approach towards visualizing the data for the different users we have mentioned in our motivation.

Visualization Layout Approach

Data is produced or recorded in various forms. But whether it is observational data or simulated data, some restructuring is necessary in order to gain knowledge from it. Transformations such as processing the data to build relations can reveal various patterns. These patterns can be detected by a user while glancing over the processed data in its numerical form, when presented as tables. But as data size increases so does the relational complexity. At this juncture applying information visualization techniques becomes imperative.

If the user is a domain expert they know the exact kind of visualization necessary for understanding the data. But a lot of times the user is in search of non-traditional ways to discover new facts. Experimenting with various techniques for visualizing the information can show a new way of analyzing the data. Through our work we have tried to do the same. We want to uncover the different relations that are generally unnoticed in job listing data. By popping out various relations through visualization we hope to reveal patterns for knowledge discovery in the employment sector.

5.1 The Job Posting Data

The job listing data we have used in our system was supplied to us by the Rutgers Career Services. The data was retrieved from a database where recruiting companies filled information about the kind of candidates they seek for a job position. The data is stored in the form of a CSV (Comma Separated Values) file. It is divided into various columns. Each column represents a category, viz. Employer, State, Job Title, Position Type, Job Description, Major, Class and Cluster as shown in figure 5.1.

Bluewater Marketing	Bluewater Marketing Gr	Part-time entry-level	Freshman,Senior,Sopl	Web/E-comme	New Jersey	Advertising/Ma	Marketing Group Seeking Writing/Editing Intern (Metuchen, NJ)
Description	Bluewater Marketing Gr	Part-time entry-level	Freshman,Senior,Sopl	Communicatio	New Jersey	Advertising/Ma	Marketing Group Seeking Data Entry/Processing
Description	Bluewater Marketing Gr	Part-time entry-level	Senior,Sophomore,Ju	Telecom,Comn	New Jersey	Advertising/Ma	Seeking/Sales Marketing Outbound Caller
Bluewater Marketing	Bluewater Marketing Gr	Part-time entry-level	Freshman,Senior,Sopl	Web/E-comme	New Jersey	Advertising/Ma	Marketing Group Seeking Paid Intern- Writing/Editing Metuchen, N
Brand Acumen is a	Brand Acumen	Full Time-Mid-Career	Alumni,Non-Matric,G	Marketing/Sale	New York	Advertising/Ma	Sales Executive
A unique opportunity	Campus Entertainment,	Part-time entry-level	Senior,Sophomore,Ju	Advertising	New Jersey	Advertising/Ma	Windows UCrew Representative
A unique opportunity	Campus Entertainment,	Temporary/Seasonal	Senior,Sophomore,Ju	Other,Commun	New Jersey	Advertising/Ma	Windows UCrew Representative
This is not a freelance	Captivate	Full Time-Mid-Career	Freshman,Alumni,Nor	Other	New Jersey	Advertising/Ma	Web Designer / Developer
The Altitude Group is	CBS Altitude Group	Internship (relevant e	Senior,Sophomore,Ju	Marketing/Sale	New York	Advertising/Ma	Strategic Marketing Intern
The Altitude Group is	CBS Altitude Group	Internship (relevant e	Senior,Sophomore,Ju	Communicatio	New York	Advertising/Ma	Strategic Marketing Intern
Collective is currently	Collective	Full-time entry-level (	Alumni,Senior,Gradua	Advertising	New York	Advertising/Ma	Cox Optimization Associate
ABOUT ZAPP360	Collective	Full-time entry-level (	Alumni,Senior,Gradua	Entrepreneursh	New York	Advertising/Ma	Sales Engineer
Looking for outgoing (ca	Costech Inc	Part-time entry-level	Sophomore	Marketing/Sale	New Jersey	Advertising/Ma	Marketing/Sales assistant
We are looking for an o	DiMassimo Goldstein	Internship (relevant e	Freshman,Senior,Sopl	Advertising	New York	Advertising/Ma	New Business Intern
DiMassimo Goldstein	DiMassimo Goldstein	Internship (relevant e	Senior,Junior	Advertising	New York	Advertising/Ma	Social Strategy Intern
Launched in 2004, DX is	DXagency	Internship (relevant e	Freshman,Senior,Grad	Web/E-comme	New Jersey	Advertising/Ma	Digital Marketing and Advertising Internship
Gain marketing	Eastern Union Funding	Temporary/Seasonal	Junior	Marketing/Sale	New Jersey	Advertising/Ma	Marketing assistant
EQ New Jersey is a	EQ New Jersey	Full-time entry-level (	Alumni,Graduate Stud	Sports,Retail,C	New Jersey	Advertising/Ma	Brand Ambassador for Verizon - Event Marketing and Sales
EQ New Jersey is a	EQ New Jersey	Full-time entry-level (	Alumni,Senior	Sports,Retail,Er	New York	Advertising/Ma	Marketing, Advertising, and Sales - Account Executive Position wit
EVINS is looking for	EVINS	Internship (relevant e	Freshman,Alumni,Sen	Public Relation	New York	Advertising/Ma	EVINS Internship Program
Focus Pointe Global, a	Focus Pointe Global	Internship (relevant e	Freshman,Alumni,Nor	Marketing/Sale	Pennsylvania	Advertising/Ma	Marketing/Social Media Internship
Position Description:	GfK	Part-time entry-level	Freshman,Alumni,Nor	Information Te	New Jersey	Advertising/Ma	Part-time Help Desk Support
Gigya prides itself on	Gigya	Full-time entry-level (	Senior	Marketing/Sale	Arizona	Advertising/Ma	Gigya G-Force - Business Development
Gigya prides itself on	Gigya	Full-time entry-level (	Senior	Communicatio	Arizona	Advertising/Ma	Gigya G-Force - Business Development

Figure 5.1: This snapshot of the CSV file shows all of the parameter. From left, description, employer, position type, class level, major, state, cluster and job title.

1. Description

The most information rich parameter, the job description enlists in detail the skills and qualities an employer expects from a candidate. This section is filled by the employers according to their protocol and thus differs from company to company. It is mainly made up of sentences which list the different technical skills, desired and expected qualifications, company policies, etc. In the future sections and chapters we will elaborate the importance of this information for our system and the users.

2. Employer

The ‘Employer’ column consists of the employer’s name advertising the job.

3. Position Type

‘Position Type’ specifies whether the job position is an Internship, Co-op, Fellowship, Temporary/Seasonal, Volunteer, Part-time entry level, Part-time experienced, Full-time entry level, Full-time mid-career or Full-time experienced. Employers can choose one or multiple

options from the above for a job. For e.g. some job positions advertise for both Full-time entry-level and Full-time experienced.

4. Class Level

The ‘Class level’ tells which types of candidates are eligible for the job, viz. freshman, sophomore, junior, senior, graduate, alumni or non-matric. An employer may also keep a job position open for different types of class levels or just one level, for e.g. ‘seniors and juniors’ or just ‘alumni’. There can be various combinations. Some companies want a more diverse pool of candidates to apply for a job position and after the interviews they can select the appropriate candidate.

5. Major

The ‘Major’ column enlists the various majors accepted by the employer for that job. For this parameter there is an interesting option called ‘All Majors’. If a employer feels the candidates for a specific job position need not be from any specific major, they can opt for ‘All Majors’ for the ‘Major’ category. A job position can also have multiple majors as specified by the employer.

6. State

The next column specifies the ‘State’ where the advertised job is located.

7. Cluster

The Rutgers Career Services have categorized jobs into clusters of specific academic concentrations. This information is stored in the second to last column.

8. Job Title

The last column in the CSV file is the ‘Job Title’. The ‘Job Title’ gives an idea about the job function of the position in a nutshell. Job seekers generally decide whether to read the job description depending on the first impression made by a job title. Some titles can be very specific to the duties while others can be vague or have very less relation to the duties.

All of this information will be used to build relations based on common factors. Many of the Majors, Classes, Clusters and Job Positions are common among several jobs. Viewing common relations visually in the form of an overview can answer many questions of a user. In the next section we will discuss the idea behind restructuring the raw data.

5.2 The need to rearrange data

The aim of developing this system was to help students, faculty and domain experts. We want graduating students to better understand the job market and search for jobs more efficiently using our system. Students who have just entered college or are currently pursuing a degree can use the system to know the skills and qualities expected by most of the recruiters. They can then plan to take courses accordingly. Reacting to the demand of companies by producing an eligible workforce can help educational institutions in a number of ways. Thus faculty and administrators can use the system as a tool to modify the syllabus according to industry trends and needs. Domain experts who need a new direction of looking at labor data can uncover more facts to supplement their studies. In order to help all these users gain more insight from the column data (CSV file) we need to present the information in the form of a structure. A structure that shows connections between jobs through common factors.

As mentioned in section 5.1, our data is tabular in nature with various categories. When we scan through any tabular information one row at a time, we are searching for patterns or repetitive commonalities that might popup. We are essentially trying to make a mental image of the commonalities while reading each job entry (row). Our goal was to design a system which can do this for the user through data processing and visualization. The natural and most simple choice in such a case would be visualizing the table as an association matrix as shown in figure 5.2. Unique ID's of the elements, in our case job titles would be the rows while the columns would be values in each category. Connections between a job title and any category would be shown by coloring a tile common to the specific row and column.

	Description	Employer	Position Type	Class Level	Major	State	Cluster	Job Title
Job 1								
Job 2								
Job 3								
Job 4								

Figure 5.2: Visualizing the CSV file data as an associative matrix for showing connections between jobs and different parameters.

But an association matrix still keeps the information in the form of a table and requires scrolling up and down in a linear fashion to learn about relations of other job titles. This motivated us to wrap the data spatially so that it can fit in one single view. With the goal of a concise visualization we plotted the job titles and all the categories in a circular layout as shown in figure 5.3. It is an effective way of presenting multi-category relational data when there are a large number of entries. The circular layout also lets us make better use of the visualization canvas.

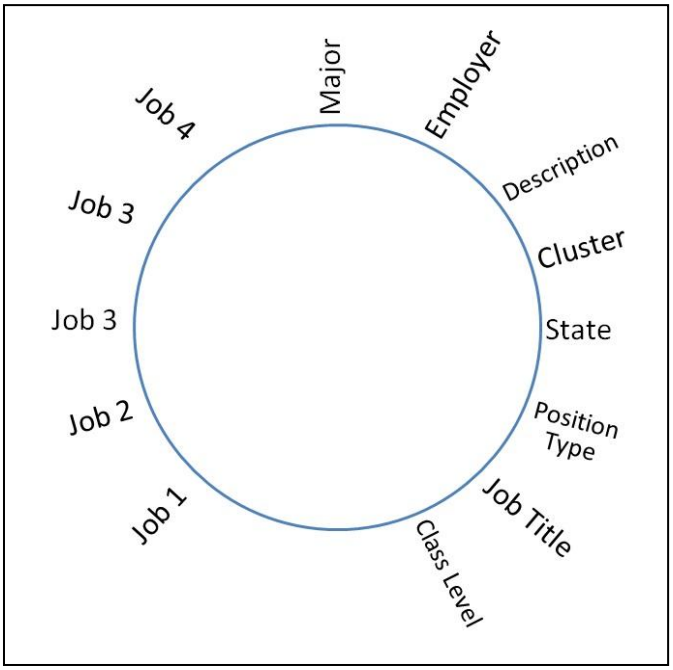

Figure 5.3: The parameters and jobs (rows and columns) are plotted in a circular layout as a rough idea of visualizing the CSV file data.

Each category or column from the CSV file is assigned a segment on the circle and so are the job titles. We are essentially converging the data into a smaller area. This makes the data concise for a quick overview by forgoing the need to scroll up and down a table. Our aim here is to let the user first have a look at all the data at once in the form of an overview. They can then dive into details by interacting with the visualization. In the next section we will talk in detail as to how we have modified the circular layout in order to make the visualization more engaging and to add more features.

5.3 A Modified Circular Layout

As mentioned in the earlier section we chose to layout the information in a circular fashion. All unique elements of each category are plotted as nodes on the circle with equal spacing. Such as in the ‘Class’ category we found there were seven unique elements, viz. freshman, sophomore,

junior, senior, graduate, alumni and non-matric. Each job title can have one or more of these assigned to them. Similarly a job title can have multiple ‘Major’ and ‘Position Type’ values assigned to it. Every unique element is assigned an individual node. If two nodes are related, the connection is represented by a link between the two as shown in figure 5.4. In the following sections we start building up the visualization.

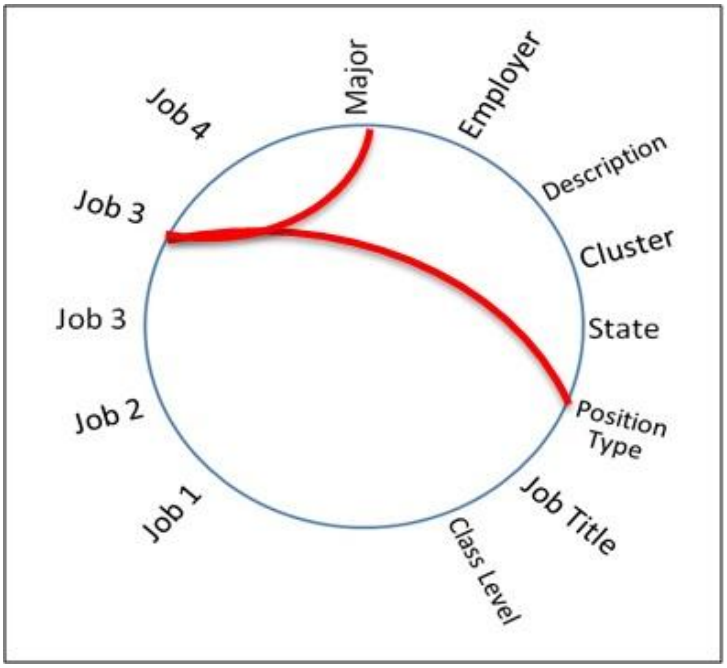

Figure 5.4: Relationship between different parameters is shown by links between parameter nodes. As more links start filling up, a link density distribution will be visible.

5.3.1 Varying Radii

For our visualization we divided all the nodes into segments, one each for the majors, the class, the position type and the job titles as shown in figure5.5. The segment for job titles is the longest with 500 nodes, one node for each job. The segment representing the different majors is the second largest with 194 nodes. The other two segments are much smaller with 8 nodes for the class category and 10 nodes for the position type category.

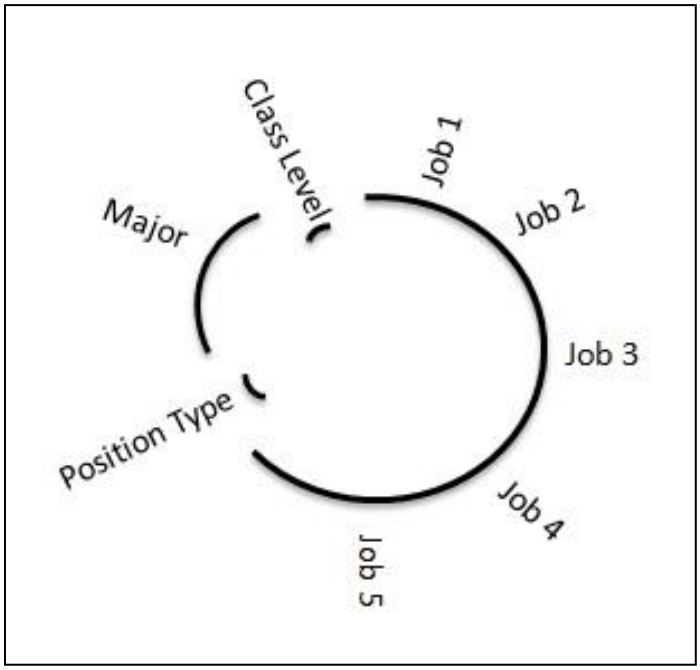

Figure 5.5: Segments are detached so that each segment can have a varying radius. This helps to distinguish different segments and also allows the Job segment to grow in length as more jobs are added.

The number of jobs to be visualized can increase to 1000, 5000 or even 10000. But the number of nodes in the class category and the position type category will remain the same. Nodes in the major category might increase but not with a rate proportional to the job titles, since there is limited number of majors offered by any university. Taking into consideration the scalability, we have detached the segments from each other. The segments have varying radii, with the class segment as the innermost, followed by the position type segment. The majors segment has a larger radius to accommodate the 194 majors. And finally the segment displaying each job title is visualized with the largest radii as shown in figure 5.5. The floatation gives us the flexibility to accommodate more nodes on the job title segment by just increasing the length of the segment.

Since the position type and class segments are very small compared to the other two, we could place them in the interior of the layout with significantly smaller radii. But we want to place them

on the screen so that the overall layout of all the segments looks smooth and can help a user trace all nodes in one sweep of their eyes.

5.3.2 Job Title Categorization

The job title segment is an important segment mainly because it tells about the job in a nutshell. A mere glimpse at the job title can tell a job seeker what kind of work is expected at that position. Thus it is necessary to place each job in a meaningful way. We have considered the fact that this framework of visualizing job listings should be scalable for an increasing number of jobs. And as the number of jobs increase it would be harder for users to read every job title. Thus we tried categorizing the jobs on the segment itself. This way the users can dive into the second level of hierarchy. We categorized the job title segment in two different ways: according to the company and according to the cluster name

Some users search for jobs offered by specific companies. One of the factors being they know the area of expertise of that company, while others have heard about good work conditions and similar factors in that company. We thus categorized the job titles according to the companies which posted them. A different color is assigned to each company and the companies are placed alphabetically. The above way of categorizing is helpful when the job seeker knows which companies they need to target in relation to their education. But for the freshman and sophomore students the companies are just names they might or might not have heard. To help such students, the Rutgers Career Services categorized each job title into clusters.

5.3.3 Tag Cloud

A tag cloud is a visual representation of important keywords in a text. The words vary in length and can be placed in different orientations, thus forming a cloud. The font size of each word

indicates the number of time that word is encountered in the text. Thus a tag cloud can be used to tell the user about words most important or more frequently used in the text bypassing the need to read each sentence.

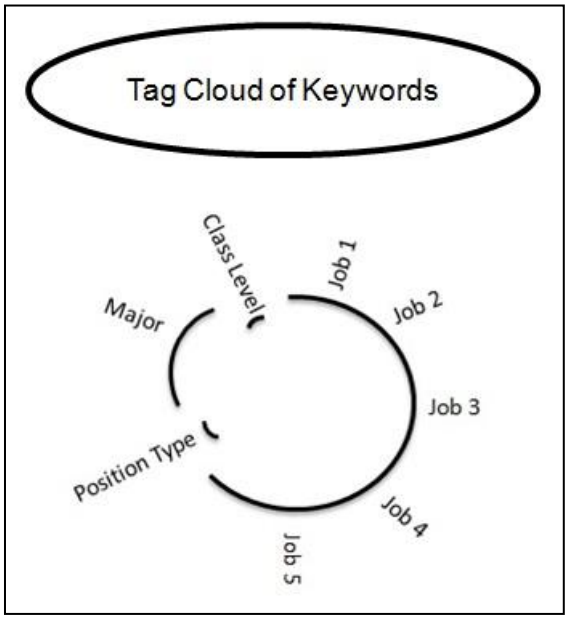

Figure 5.6: The keywords from job descriptions are represented by a tag cloud. Initially we placed the tag cloud on top of the circular layout.

We are going to implement a tag cloud in our visualization for the same purpose as stated above. With a goal to save a user's time reading all the text in each job description, we will only display keywords encountered in the description. As mentioned in section 5.1, the description enlists in detail the skills and qualities expected by the employer. Being sentences, the description is the biggest chunk of the data for each job. Thus the tag cloud will efficiently display only the required words. We will use the tag cloud for displaying keywords for each major. Every major will have a list of jobs and their descriptions will be used as raw data for the tag cloud. We will explain in detail the data processing in chapter 6. Initially we placed the word cloud on top of the circular layout as shown in figure 5.6.

Placing the tag cloud on top of the circular layout increases the visualization canvas we used. In addition this adds to user interaction. Users have to scroll up and down in order to view the nodes on the circular layout and the tag cloud. Thus we placed the tag cloud inside the circular layout as shown in figure 5.7. This way the empty space inside the circle is used and all the information is displayed to the user in one view.

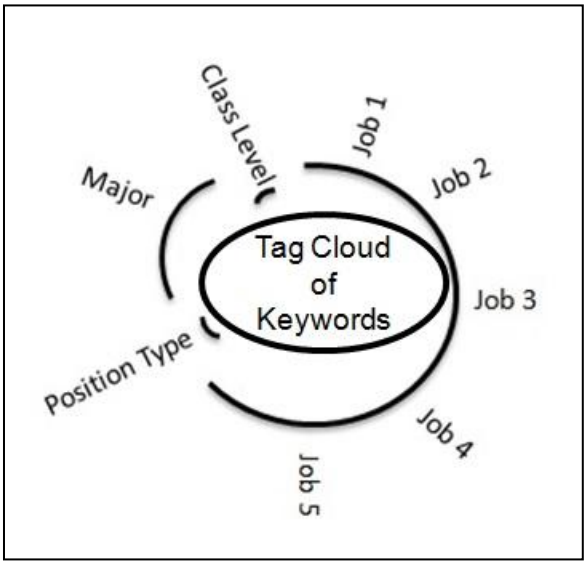

Figure 5.7: Placing the tag cloud inside the layout maintains context and reduces a user's eye movement.

Placing the tag cloud inside the circle will also help the users to interact with the visualization in an intuitive way. We will talk more about the interactions in section 7. In this section we wanted to put forward how the layout started to take shape and the factors contributing to the decision. In the next chapter we will talk about the more technical back end data processing stage.

Data Processing

In the earlier chapter we discussed step by step how the different parts of the interface were placed to form a single compact visualization. In this chapter we will discuss in detail the technicalities of restructuring the data in the form required by D3. The data in the CSV file we obtained from Rutgers Career Services had to be read and then processed according to our implementation needs.

Through our data processing we perform validation to ensure that the data is clean and correct, restructure the data to reveal relations, summarize some parts of the data to main points and finally convert the data storage file type. A large part of the processing is done for the job description of each job since it will be used to visualize the tag cloud. The job description for each job is written by each company's human resources department. This results in a lack of uniformity across jobs in the career services job database. All the other parameters like position type, class, major etc. are to be selected from a predefined set of values, thus they require less processing. Figure 6.1 shows a flowchart explaining the whole process. In the following sections we will talk in detail about each procedure.

6.1 Cleaning and summarizing data

The job description for every job is made of sentences which elaborate qualities, skills and requirements for an eligible candidate. A job seeker has to go over a hundred jobs to search for the one's matching their profile. Thus we want to save the job seeker's time by only visualizing the words that appear on most resumes, mainly the technical skills and requirements.

Step 1: In order to do this we first store a list of pronouns, adjectives, verbs and adverbs to be filtered out from the job descriptions. This list is saved as a separate CSV file so that additions

and deletions are easy. The file is read by the D3 function ‘d3.csv.parseRows’ and the words are stored filtering purposes later. We will call them as Filter-words.

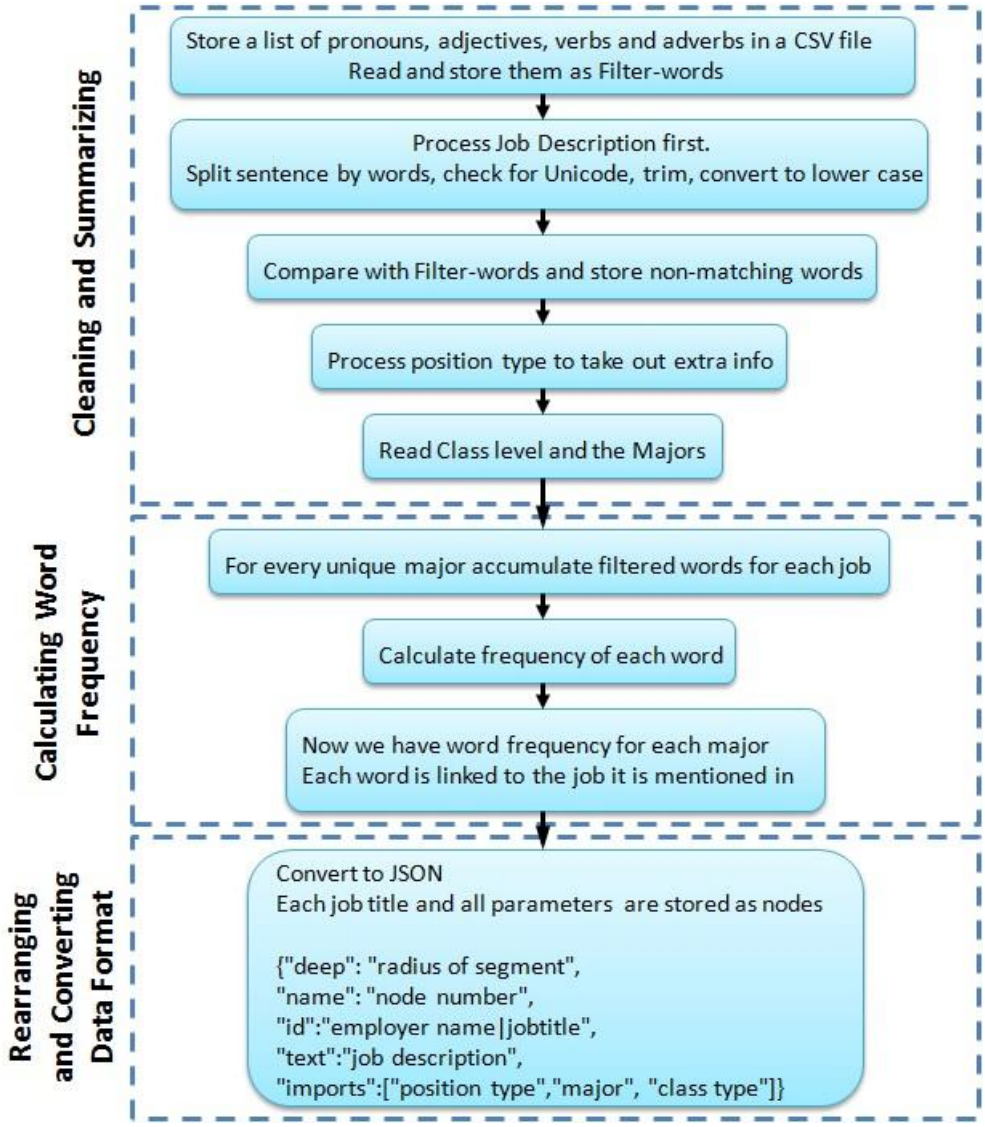

Figure 6.1: Flowchart of the CSV file data processing. The whole process is divided into three parts, cleaning and summarizing, calculating word frequency and rearranging and converting the data format.

Step 2: Next we read the CSV file from Career Services. In this file, each column represents parameters such as job title, job description etc. while each row is a unique job entry. All the entries are read using the same ‘d3.csv.parseRows’ function and stored.

As we mentioned earlier the human resources department of each company has freedom to enter into the Rutgers Career Services system job description and job title deemed appropriate by them. This means the information is not at all uniform throughout the Rutgers career services database. The description also contains special characters and Unicode characters. We therefore prioritize processing the job description first.

We split the sentences by blank spaces, giving us some proper words, some with special characters, some with extra blank spaces at their ends and some with Unicode characters too. All the words are first checked for any Unicode characters. Then we check for presence of any special characters. If found they are removed from the word. If any extra blank spaces are present at either end of the words, they are removed too. Finally all letters in each word are converted to lower case. This helps while comparing and filtering out unwanted words using the Filter-words.

Step 3: Now we check the processed words with the Filter-words to filter out the pronouns, adjectives, verbs, adverbs and other unwanted words. What we are left with are words that will catch a job seekers attention when visualized as a tag cloud. But they have to be further processed before being visualized as we will explain in the next section.

Step 4: For each job entry there can be one or more position types. And each position type has extra information which is not necessary for our visualization. For e.g. Part-time entry-level (some college education, less than 35 hrs/wk), Internship (relevant experience, furthers career goals, less than 25% clerical). Here all the information in the parenthesis is not required for our visualization. We only need the position types. Therefore when we read the position types for each job entry, we slice off the information in the parenthesis.

Step 5: Similar to the position type, a job entry can have multiple majors and class level. But they do not have any extra information and thus are read as they occur in the CSV file.

6.2 Calculating Word Frequency

Now we start structuring information for the tag cloud. From the processing in section 6.1 we have all filtered words for each job. And as explained above we also have a list of majors for each job. Our visualization goal is to show a tag cloud of filtered words of all jobs for any specific major.

Step 6: For the above reason we first make a list of all the majors occurring in the CSV file. Remember that some jobs may also mention ‘All Majors’ along with some specific majors. We consider this as a separate entity and thus it is one of the majors in the list.

Step 7: For every major in the list, we know the jobs where they are mentioned. And thus we can also connect them to the filtered words in each job. We first accumulate all filtered words for each major and then calculate the frequency of each word. We now have a word frequency for each major along with a link between the word and the job where it is mentioned. The last step is to structure the data in a format that can be quickly accessed by D3.

6.3 Rearranging and converting data format

For the D3 functions to quickly access the information we processed in the earlier two sections, we need to structure the information in a specific way. We will do so in JavaScript Object Notation or JSON. It is a format written in human-readable text. JSON is the syntax for storing data on a server and transmitting it to a web application. Data is stored as objects and every object

can have any number of ‘attribute-value’ pairs. To accessing the information we simply have to refer to the name of the attribute.

Step 8: As mentioned in section 5.3, we have planned a circular layout to visualize our data. Every unique job title, position type, class and major will be placed around a modified circle as individual nodes. When storing this in the JSON format, each job title, position type, class and major will be represented as an object. Following is the object format for all job title nodes.

{"deep": "radius of segment",
"name": "node number",
"id":"employer name|jobtitle",
"text":"job description",
"imports":["position type","major", "class type"]}

There are five attribute-value pairs. The attribute ‘deep’ specifies the radius of the segment on which the node will be placed. As we know we have different radii for different segments in our varying radii circular layout. Thus the value of this attribute will be different for ‘major’, ‘class’ and ‘position type’ nodes. The job title segment has the largest radius. The ‘name’ attribute specifies node number. The attribute ‘id’ specifies the employer name and the job title as a combination. The next attribute ‘text’ is empty for the job title nodes, but will be used for the major nodes. The last attribute ‘imports’ is very important since here we will be specifying every position type, major and class that was mentioned in the CSV file for that job. We will thus link this job title node to its respective position type, major and class type nodes, utilizing it for visualizing relations.

Once all the job title nodes are written in JSON format we write the position type and class nodes. These are the smaller segments and consist of around 10 nodes each. The objects are written in the same way as the job titles. The difference is that attribute ‘id’ has a value as a combination of the string “pos” and name of the position type for the position type nodes. And that for class is a combination of the string “class” and name of the class. The ‘text’ attribute is empty. When we specified the ‘imports’ attribute for all job title nodes we already created a link to and from the class, position type and majors. So the ‘imports’ attribute will be empty here.

The last set of nodes is the majors. These nodes will be written with the most amount of information, mainly in the ‘text’ attribute. The ‘deep’ attribute will reflect the second largest radius. The ‘name’ attribute will continue with the node numbering from the last set of nodes. A combination of the string “major” and name of the major serves as the value for the ‘id’ attribute. The most important attribute here is ‘text’. At the end of section 6.2 we had the word frequency for each major along with a link between the word and the job where it was mentioned. This information will be entered in the ‘text’ attribute. This can be easily pulled by the D3 code to visualize the tag cloud. The ‘imports’ attribute is empty for all the major nodes. This completes our JSON file.

In this chapter we started by reading column data from the CSV file and processed it so that we could write it in the JSON format. Now that we have our JSON file the data can be read by D3 for visualization and interaction. In the next chapter we will discuss how the JSON data was visualized using D3 according to our layout.

Implementation

Implementing ideas into a working system is one of the major steps in visualization. In the last two chapters we discussed in detail the groundwork necessary for building our system. And in chapter 4 we introduced the technical aspects of using D3 along with JavaScript and HTML for implementing an interactive system. Considering the CSV file as raw information, we would divide it into two parts from a visualization point of view. The information rich job description is one part and the other part comprises of all the other columns in the CSV file, viz. job title, major, class level and position type. Through chapter 5 we put forward our visualization layout design, a modified circular arrangement of parameter nodes with a tag cloud that fits inside. In chapter 6 we discussed in detail the data processing section for converting the CSV file to JSON objects, along with cleaning, summarizing, building relations and calculating word frequency. In this chapter we will show the results of applying the visualization ideas to the data using web based information visualization.

7.1 Interacting with the system

Through the motivation and related work section we explained how there is a need for a visual tool for exploring employment data. Our visualization layout approach is heavily inspired by the idea of ‘focus + context’. The system is flexible in the sense that it lets the user focus on detail along with a simultaneous overview of the context. Figure 7.1 shows an overview of the implementation and the flow of control. The system comprises of two important JavaScripts. One does the data pre-processing while the other does the visualization calculations. In the following sections we will show how our system can be used.

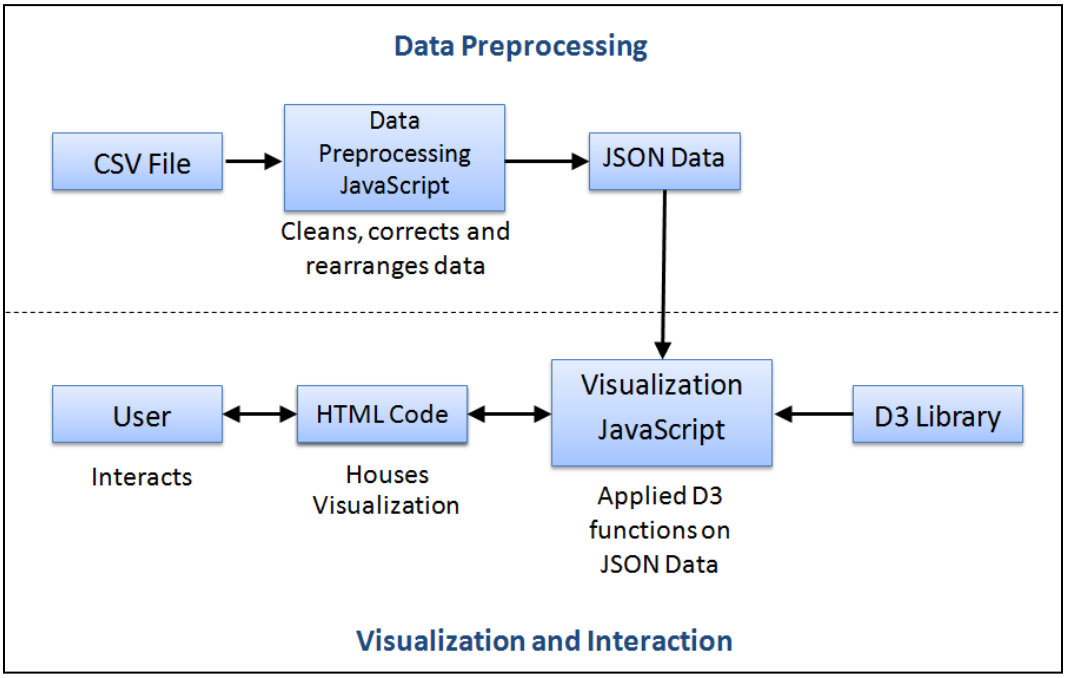

Figure 7.1: The diagram shows an overview of the system implementation along with the flow of control. The system is divided into two parts, Data Pre-processing and Visualization and Interaction. Data Pre-processing is done by a JavaScript that takes in the CSV file as input and produces JSON data. The visualization parameters on the other hand are calculated by a different JavaScript that applies D3 functions on the processed JSON data. The user interacts only with the HTML page to initiate and interact with the visualization.

7.1.1 Viewing the Whole Dataset

When the visualization loads we want to put in front of the user an overview of the data they can explore, so that they can get a sense of what is available. When the user loads the page, the HTML code in turn runs a JavaScript. The script reads the JSON data we processed in chapter 6 and calls various D3 functions to visualize the data. Figure 7.2 shows a flow of the instructions in the JavaScript to load the visualization.

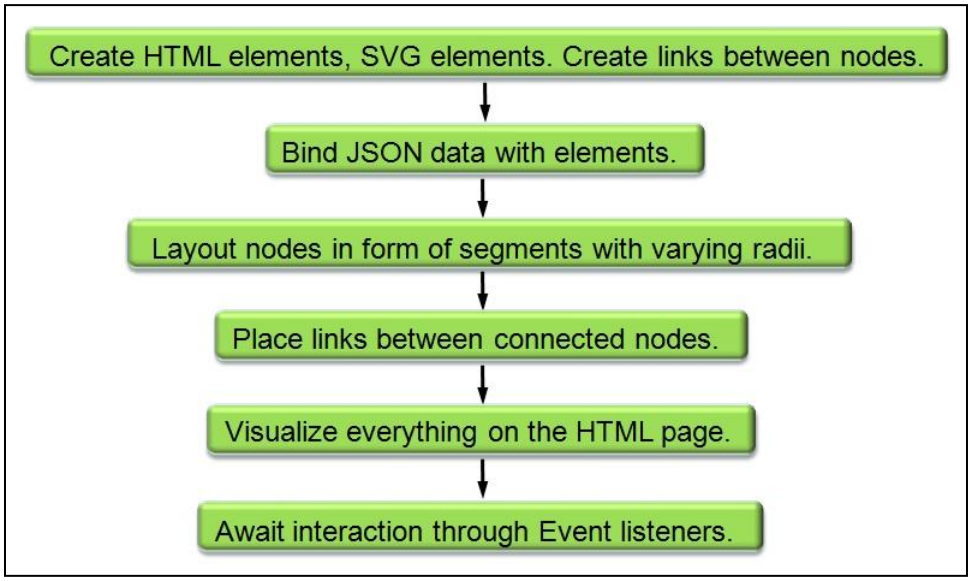

Figure 7.2: The flowchart shows the process of visualizing the whole dataset. Starting from creating HTML and SVG elements, bind them with JSON data, laying out the nodes, placing links and rendering everything on the HTML page.

Once the HTML elements, SVG elements and links are created, they are bound to the JSON data. Nodes are laid out in the circular form and links are placed between connected nodes. All these elements are then visualized on the HTML page awaiting interaction from the user. Figure 7.4 shows the view presented to the user when the visualization loads. Here we see four segments laid out with different radii. The longest segment represents the job titles, in essence each of the jobs in the dataset. The smaller segment comprises of college majors specified by the employers. And the other two small segments are for class level and position type.

These nodes are clickable and have event listeners incorporated for interaction. The links from one segment to the other depict relationship between nodes. Density of the links on the major segment shows the distribution of jobs among different college majors. The same distribution can be seen for the class level and position type segments too. Figure 7.3 shows job titles categorized according to employer. This categorization can also be changed to sort job titles according to the

cluster they belong to or the state where the job is located. This tells the user the distribution of jobs either in different states, different clusters or different employers.

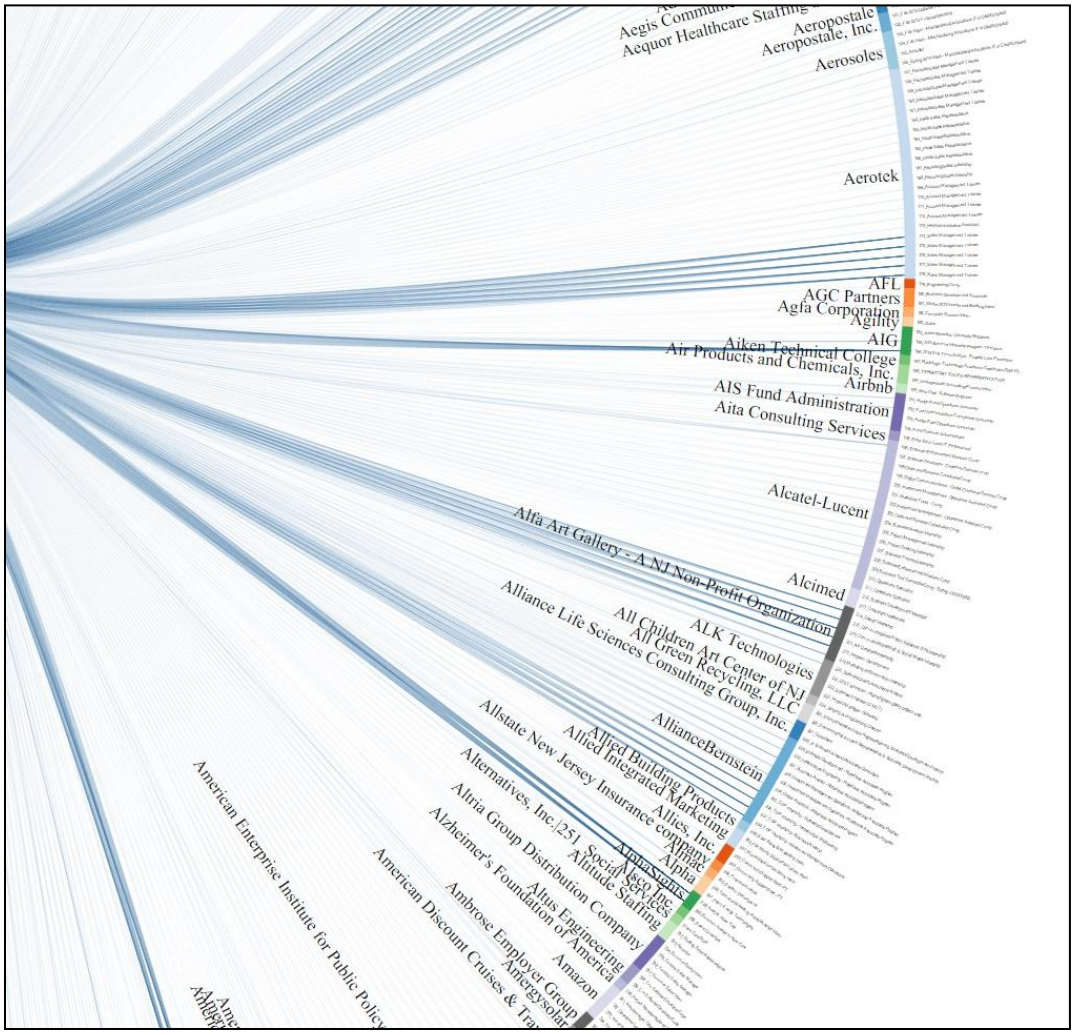

Figure 7.3: Job titles are categorized according to employer on the job title segment. Parts of the segment colored with different colors represent different companies.

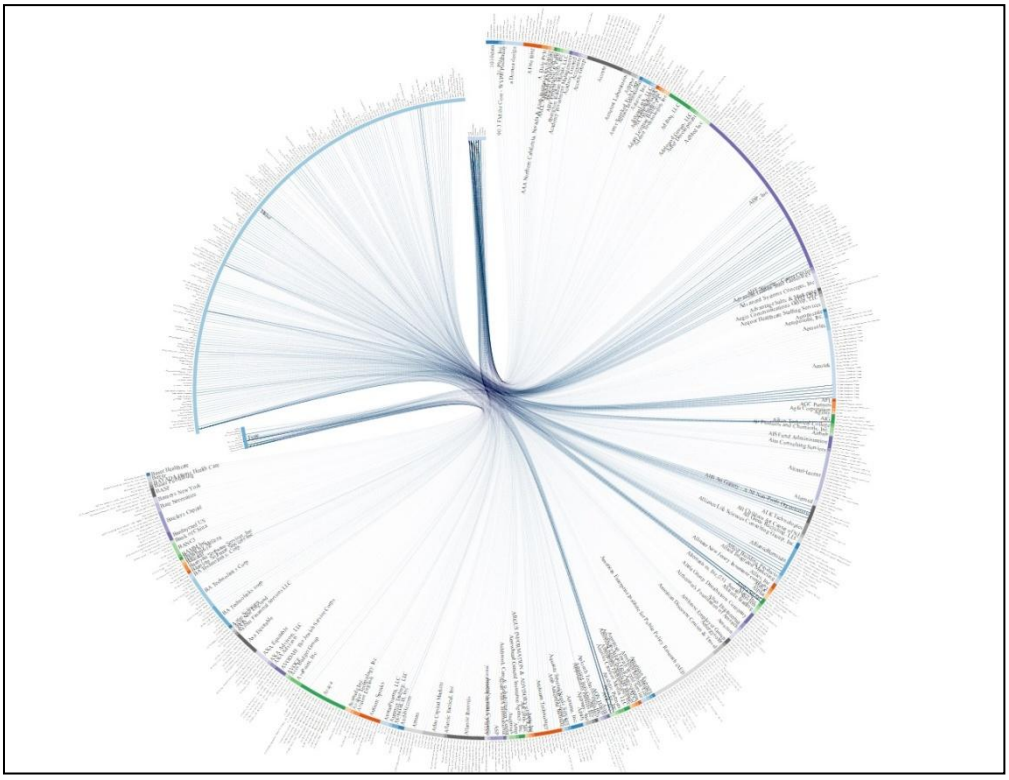

Figure 7.4: The visualization on loading up the HTML page shows an overview of the whole dataset. With rows and columns represented as nodes and connected nodes joined by links.

7.1.2 Viewing the Tag Cloud for a Major

A user can interact with the visualization by clicking on any node. By doing so, links between that node and all nodes connected to it are highlighted. Nodes on the major segment have an added feature. By clicking on a major node, the user can view a tag cloud of keywords relevant to that major, along with highlighted links from the major to all jobs accepting that major. Figure 7.5 shows a flowchart explaining the process. Once the major node is clicked, the event listener returns the node ID and highlights links connecting the major to all related job titles. It then lays out the tag cloud of the keywords with font size representing the word frequency. A collision detection algorithm built in D3 places the words without overlap.

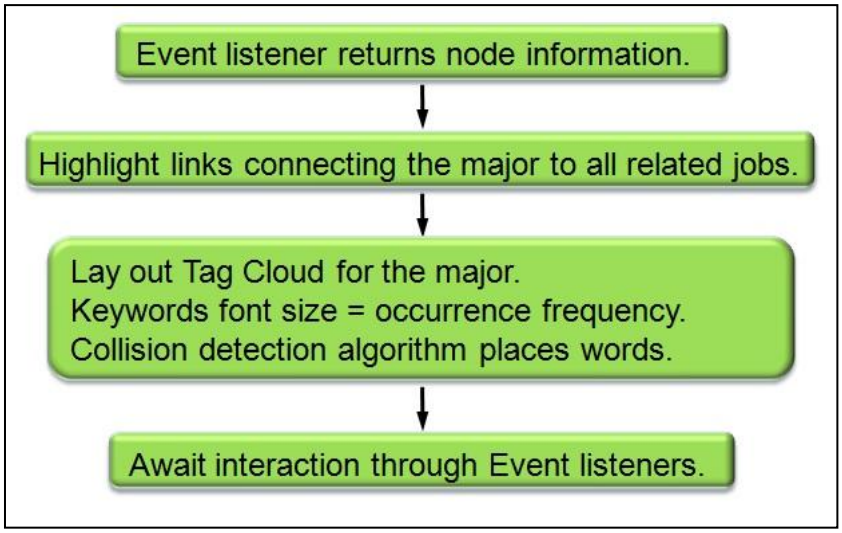

Figure 7.5: Flowchart shows the process of laying out the tag cloud and links when a major node is clicked.

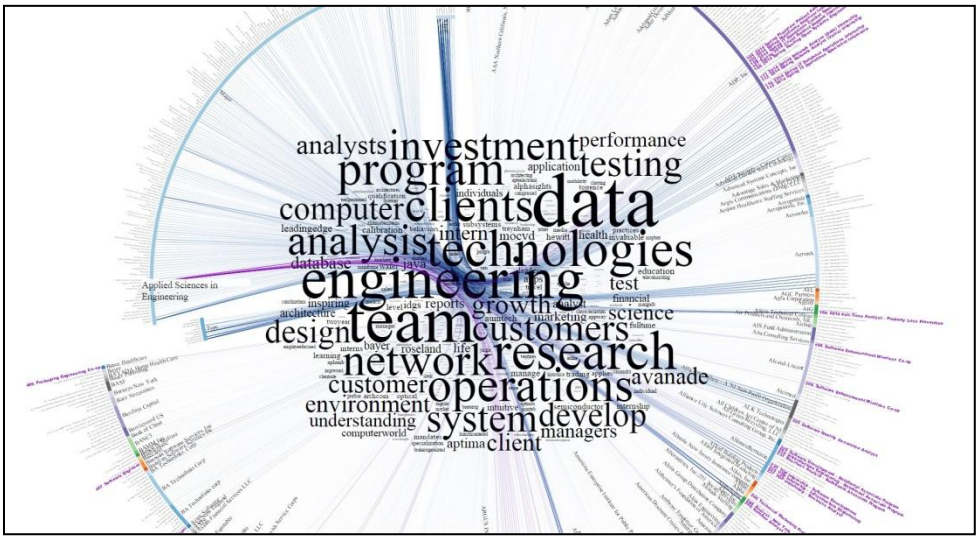

Figure 7.6: This figure shows the changes in the visualization when the user clicks on 'Applied Sciences in Engineering'. Links connecting the major to job titles are highlighted in purple along with a tag cloud in center.

In figure 7.6 we can see the tag cloud for major 'Applied Sciences in Engineering' along with the purple links connecting the major to various job titles on the job title segment. The highlighted links give an overview of all jobs accepting the specific major, by highlighting them. The job

titles at the end of each highlighted link are also magnified in order to be readable without the need for any zooming in. The tag cloud on the other hand gives an overview of the keywords most used by employers seeking candidates from that major. It shows how many times the word has been used for describing jobs linked to that major.

7.1.3 Viewing the Jobs to Keyword link

The tag cloud shows a distribution of words for the major, giving an idea about the most important skills for employers. To find out the exact job that mentioned a specific word present in the tag cloud (word to job link), the user can click on that word. When a word is clicked, the system indicates the related job by small ellipses near the job titles. Figure 7.7 shows a flowchart of pinpointing exact jobs related to a word.

So when a user clicks on a word in the tag cloud, the system refers to a list of all the jobs where the word was used. Next, the location of each job on the job segment is retrieved to place green ellipses as indicators. We decided on using remote indicators instead of curves for the links, so as to avoid occlusion and clutter. Figure 7.8 shows the green indicators placed on the inner side of the rim when a user clicks on the word ‘testing’ from the tag cloud.

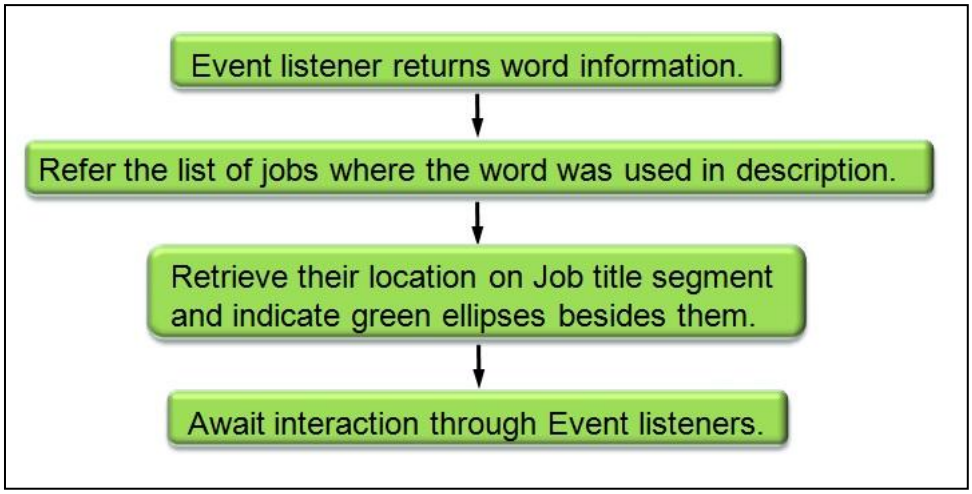

Figure 7.7: The flowchart shows instructions for initiating the word to job link. The system refers to a list of jobs where the word is used and retrieves their location on the job title segment. Green ellipses are indicated besides relevant jobs.

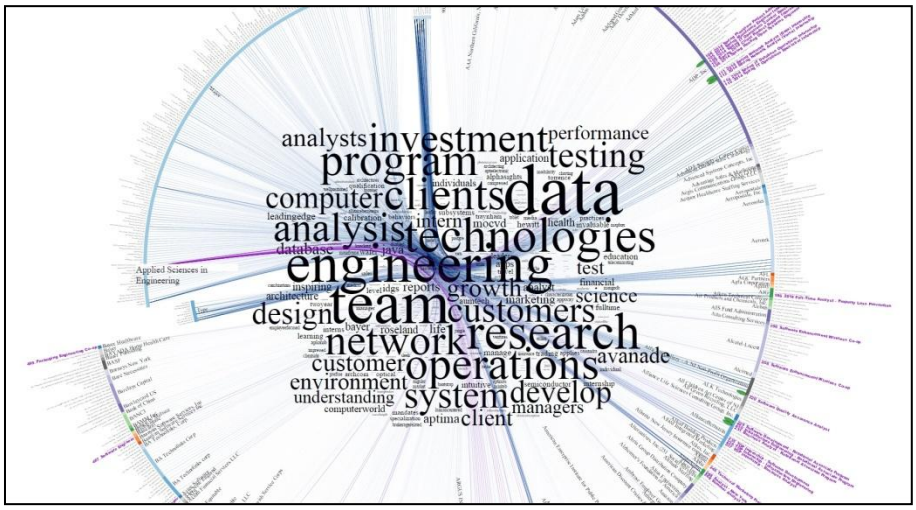

Figure 7.8: The figure shows green ellipses on the inner side of the rim when a user clicks on the work ‘testing’ from the tag cloud.

The whole visualization at this point shows different levels of detail in one view:

1. The system displays all the jobs available (the whole dataset).

2. Next for ‘Applied Sciences in Engineering’ it highlights the jobs keeping all other jobs still in the view.

3. For ‘Applied Sciences in Engineering’ it shows the keywords from all the highlighted jobs, essentially giving a gist of their job descriptions and also conveying importance of each word.

4. For the word ‘testing’ it pinpoints the jobs where employers used the word in the job description.

7.1.4 Viewing the Jobs for a Class or Position

If a user wants to focus on a certain group of jobs that are advertised for a specific class or a specific position, they can do so by clicking nodes on either the class level or position type segment. By doing so, the system will highlight links connecting job titles to only those nodes.

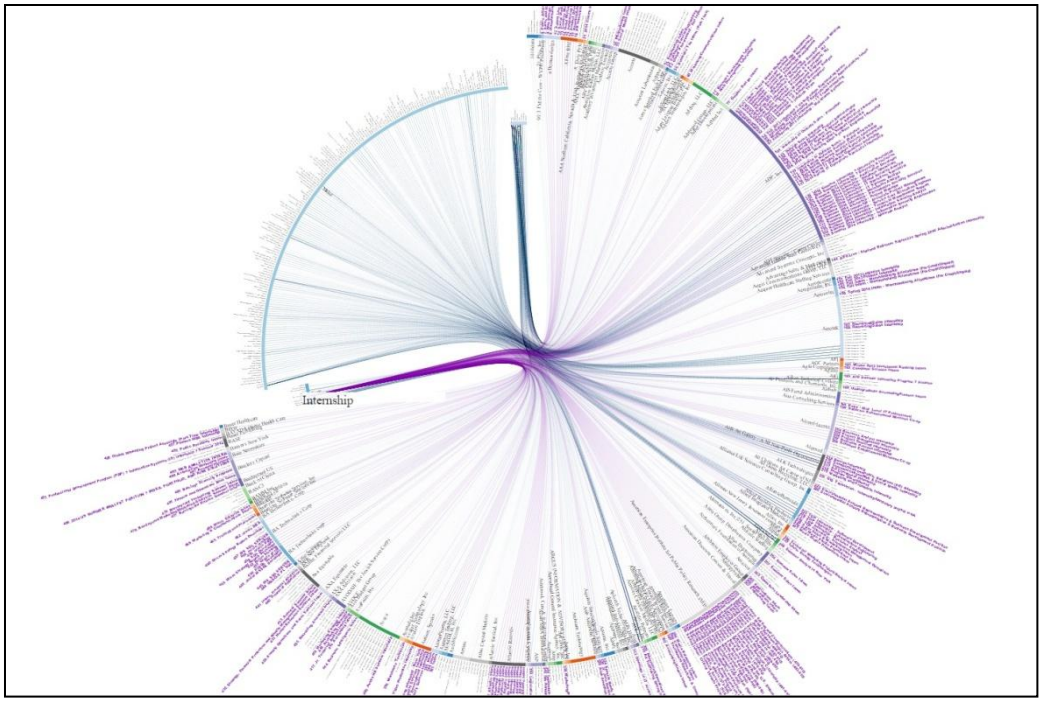

Figure 7.9: The visualization shows job titles highlighted after a user clicks on the ‘internship’ node on the position type segment.

Figure 7.9 shows links and job titles highlighted after clicking on the ‘internship’ node on the position type segment. This shows all the job titles which specified internship as the position type in the job postings. A user can similarly highlight jobs for any specific node on position type segments.

Visualizing job titles by specific class level or position type helps a user filter jobs according to their profile. Thus as compared to the traditional list based search process, users can view all jobs relevant to an applied filter in one view. It also gives an overview of the distribution of jobs which can be used for comparing statistics of different class levels, position types and college majors.

7.1.5 Viewing individual Jobs

Users also have the option to scan the individual job titles by going through the job title segment. By hovering the mouse over nodes on the job title segment the node’s job title is shown in the form of small pop-ups. Clicking on a specific job title then highlights links to the majors, position types and class levels requested by the employer as shown in figure 7.10. This provides the most detailed view of the data, but some users prefer to search jobs according to job titles.

Sometimes job seekers want to search jobs according to the employer, mainly because they know some well established employers doing business in their field of study. For this reason we have employer names on the inner side of the job title segment. The user can refer to the employer names and scan jobs relevant to the employer of their interest.

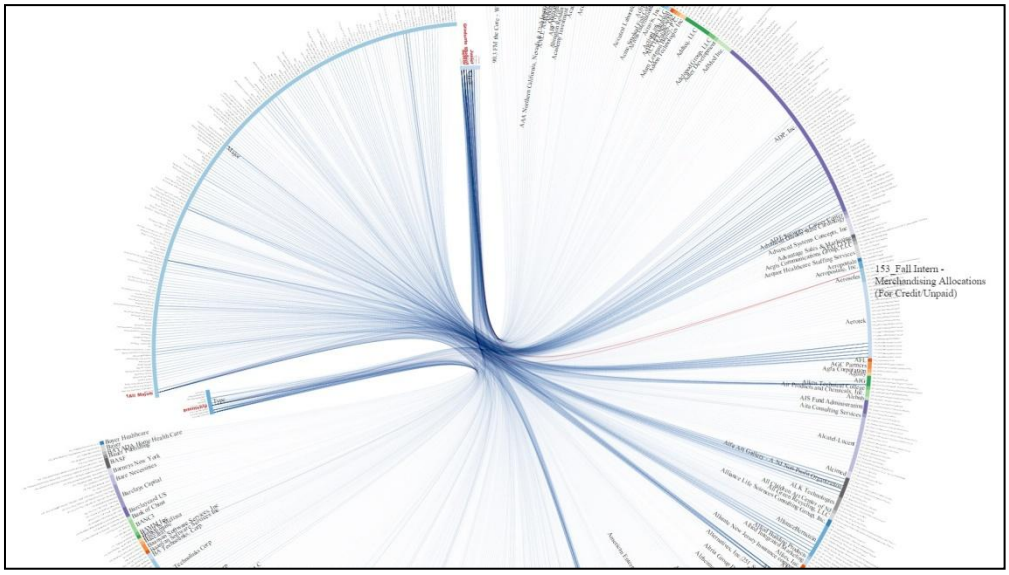

Figure 7.10: Individual jobs can be explored by clicking on job titles on the job title segment. Here we see links highlighting class levels, position type and majors for the job titled ‘Fall Intern – Merchandising Allocation’.

Discussion

The user profiles for our system vary from students to domain experts. Irrespective of their role in the employment sector everyone likes to use a system that is easily accessible. Designing the system as a web application was thus important. Because the webpage and script files are hosted on a server, there is no installation or setup required from the user's end. Any user can access the system anywhere. The other great advantage of the system is that once it is loaded even if there is no internet connection, the system can still be used. Interactions with the visualization will be carried out by the system irrespective of the internet connection.

One of the elements in our visualization which made the system slow to interact with was the SVG curve. D3 plots Bezier curves to represent links from one node to the other. In the dataset visualizing 500 jobs, the system plots more than 1500 curves. Because of the curves, zooming in and out becomes slow. One solution to this is to use curves that are not very smooth, in turn leaving the links dispersed and not bundled at the end node. But then the density of lines representing job distribution is not so apparent. As a solution to this we plot all the links when the visualization loads up the first time. This way the user can have the overview first. Subsequently when the user clicks on a specific node to highlight its links, all other links are removed and only the ones connected to the node are highlighted and rendered. This made the zooming and panning easier and faster.

One other disadvantage of D3 is that the system's interaction capability scales badly as the number of nodes are increased. But we were successful in reducing interaction latency by displaying node text on hovering and clicking only. We wanted to keep a balance between the minimum number of elements always viewable and the number of elements for a user to

intuitively click, zoom and pan the visualization. This design principle was also applied to the usage of text in the visualization.

Coming back to the issue of scaling with number of nodes, we tried a different layout to accommodate more nodes. Figure 8.1 shows 5000 jobs placed in an elliptical layout. The modified circular layout is good for square visualization spaces. But since most of the desktop and laptop screens are rectangular an elliptical layout uses most of the available space. This also lets us visualize more jobs and majors.

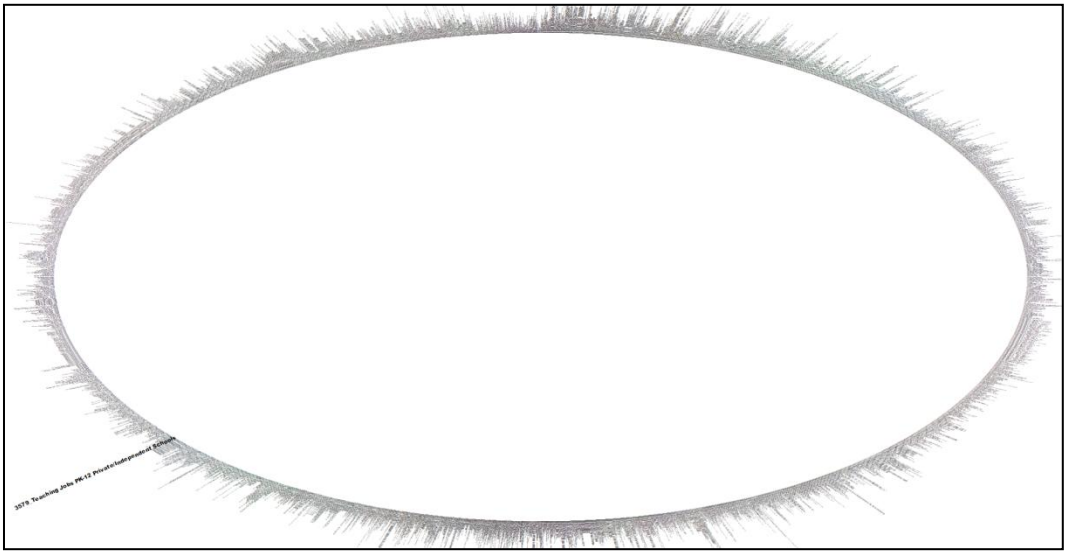

Figure 8.1: The figure shows 5000 jobs placed in an elliptical layout. This way we can use most of the available visualization space and accommodate more jobs and majors.

When the number of job postings and subsequently the number of nodes to be placed on the screen touch thousands, it will be hard for the user to read each node individually even if the node is enlarged upon hovering. Also as seen in figure 8.1 the overview is cluttered and is not very welcoming for a user to explore the data. A solution to this would be using a hierarchical layout. Instead of visualizing all the nodes at once we could categorize them according to industry or cluster and only show the main group names on the initial visualization. When the user is interested in a specific group, they can click on that group so that the system visualizes all nodes for that group. This way the user can scan all the nodes group by group without being overwhelmed.

Conclusion

Job posting web sites are the most common way that students and professionals search for employment. All the job searching tools operate in a similar way, where postings matching the keywords are displayed as a list and the user is required to read through this list one by one. In this thesis, we explored the use of other types of information visualization techniques that would be appropriate for this type of data. We have developed a new visualization especially for student-based job searches and have applied our results to data from the Rutgers Career Knight Portal. By putting stress on 'Focus + Context', we have tried to give the user a tool that can help them discover more. Displaying the whole dataset in one view and then letting the user access different levels of details is faster as compared to going through lists of jobs and reading each posting.

For future work we would like to make the interaction faster. Nothing discourages a user more from knowledge discovery than a slow system. As part of the next phase, we plan to deploy the system for user testing on Rutgers Career Knight Portal. Students can log into the system and explore the data in the career services portal. By connecting the system to a database we can record how students use the system. This can help domain experts in the labor sector understand how students from a major search for jobs when all the data is presented to them. Simultaneously we would like to try different layout options for visualizing bigger datasets. In the latest prototype we were able to visualize 5000 jobs, but we would like to experiment with other hierarchical layouts.

References

1. Career Knight, Career Knight, 2013.

2. Indeed, Indeed, 2004.

3. Glassdoor, Glassdoor, 2007.

4. bigml,Using Text Analysis To Predict H-1B Wages, October 2013, http://blog.bigml.com/2013/10/01/using-text-analysis-to-predict-h1-b-wages/

5. Satyam Gupta, Geographical trends in skills using LinkedIn's Endorsement feature, August 2013, http://engineering.linkedin.com/endorsements/geographic-trends-skills-using-linkedins-endorsement-feature

6. NPR, Every Job in America, in 1 Graph, January 2014, http://www.npr.org/blogs/money/2014/01/09/261053608/every-job-in-america-in-1-graph

7. Gapminder, The Gapminder World Map, August 2012, http://www.gapminder.org/downloads/world-pdf/

8. Panos Ipeirotis and John Horton, Visualization of the oDesk "oConomy", July 2012, http://www.behind-the-enemy-lines.com/2012/07/visualizations-of-odesk-oconomy.html

9. Bilal Alsallakh, Wolfgang Aigner, Silvia Miksch and M. Eduard Groller, Reinventing the Contingency Wheel: Scalable Visual Analytics of Large Categorical Data, IEEE TVCG, December 2012.

10. US Census Bureau, Where do college graduates work?, July 2014.

11. Tableau, Tableau, 2003.

12. Spotfire, Spotfire, 1996.

13. Krzywinski, M. et al, Circos: An Information Aesthetic for Comparative Genomics, Genome Res, 2009.

14. Bederson, B. B., Grosjean, J., and Meyer, J., Toolkit Design for Interactive Structured Graphics, IEEE Transactions on Software Engineering, 2004.

15. Prefuse, Prefuse, 2004.

16. Data Driven Documents, http://d3js.org/, 2011.

17. Wikipedia, http://en.wikipedia.org/wiki/HTML, 2015.

18. Monster, Monster, 1999.

19. TextArc, www.textarc.org, 2002

20. Jean-Daniel Fekete, The InfoVis Toolkit, IEEE Symposium on Information Visualization, 2004.