University of Glasgow | School of Computing Science

Are matrix-based or node-link graphs more readable when representing causal relationships for social and health data?

Kristina Lazarova

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March, 2017

Abstract

People who work in the social and health policy make decisions based on causality. Causality is usually visualised by node-link graphs, but matrix-based representations have the potential to outperform them because of their concise and clear visual representation. However, matrix-based graphs have not been extensively studied in causal relationship visualisation. Therefore, an experimental software was created to test human performance on reading causal relationships in node-link and matrix-based graphs. To make a thorough investigation of the factors influencing reading time different sizes, layouts and type of questions were created. Response time and correctness of the answer were measured. The results showed that matrix-based graphs outperforms node-link representations in direct questions. When path questions are concerned, however, node-link graphs were found to be more readable than matrix-based graphs. (DISCUSSION)

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____   Signature: _____

Contents

Chapter 1

Literature Review

1.1 Data Visualisation

Data visualisation has been described as a technique that makes use of computer-supported, interactive illustrations to deepen human's understanding of a dataset [5]. Information visualisation (InfoVis) is necessary when dealing with increasingly large and complex data. InfoVis systems are most helpful when a set of data is being explored [12]. Usually this occurs not when someone is looking for a specific answer to a question, but when a deeper understanding of that data set is sought. It was suggested that the value of information visualisation is not in understanding a specific question, it is about developing and deepening one's insights of a set of data [12]. Data visualisation is able to facilitate this process because there are a number of cognitive benefits associated with it. A large visual cue that illustrates data becomes a single point of reference for human cognitive processes. Visuals become external cognition helpers in facilitating human memory by providing a bigger working set for analysing data.

Density of the InfoVis and task difficulty are known to influence human performance [27]. For instance, following a single path on a graph with one trajectory would require less cognitive effort than performing the same task on a highly dense and complex network. However, the exploration of a representation can convey a better insight into the meaning of the data [2]. Al and colleagues [2] argues that InfoVIs has three main roles: communication field, knowledge management mean, and decision support instrument. Furthermore, visual architecture and design applied at company and department levels have been reported to be successful due to the low cognitive burden for visualization reading [24].

Data visualisation has been been used in a number of fields. For example, it is suggested to be successfully used in Bank of America Chicago Marathon, which is one of the biggest marathons in the world [20]. Large amounts of live data are gathered to establish some of the main principles for event management information visualization. The main benefits from these practices are keeping track of the progress of the participants, communicating information between the agencies organizing the marathon efficiently, and improving medical preparedness.

Moreover, astronomical researchers have been suggested to benefit from data visualization techniques [15]. One such technique is linked views in which a user is able to select, highlight and include or exclude points from display and analysis from various data visualisation dimensions. For example, an important visualisation for astrologists can be interactive exploration of relationships between data points in statistical graphs and locations in live 3D space. Another instance of beneficial data visualisation is "Geographic Information System" tools which are used in demographics and geography. For example, Engage3D provides the functionality of exploring layers of maps in a linked-view systems.

Inventions in this field have been continuously made in order improve the visualisations and benefit the target public. For example, a heatmap represents a graphical illustration that evaluates conical distribution values around data points based on a respective data value that has associated respective data point [6]. Various implementations of a heatmap were developed and discussed since the original heatmap was created, in order to enable easy future work with heatmaps suited for a particularly needed area.

There have been data visualisation issue related to Big Data [16]. First, visual noise is the phenomenon of having so many data points on the screen that the user cannot see them as separate points which leads to visibility loss. Second, large images are dependent on the device's resolution abilities. However, even if many devices are added together for partial data visualisation to represent a more detailed illustration or a larger amount of data, human perception will eventually reach its limit. Therefore, after a certain point people will not be able understand the representations or their analysis. A third problem is information loss which is caused by data filtration and aggregation. These techniques are likely to hide information and mislead people analysing the data. Another problem is High Performance Requirements especially in dynamic visualization. In addition, the visualisation might be accompanied by high rate of image change and lead to people being unable to react to the number of data changes.

A Big Data visualization method is suggested to be TreeMap [16] which is a hierarchical way of representing data by rectangles. Some advantages of this method are that the hierarchical matching clearly shows data relations and extreme outliers are visible. The disadvantages related to this method are that the data must be hierarchical which makes it unsuitable for examining time patterns, for example.

A well-known issue if InfoVis is that sometimes it is challenging for people to understand it. A study aimed to examine familiarity of museum visitors with different visualisation techniques [4]. They included charts, maps, graphs, and networks to reveal how familiar people are with them. It was found that even though most participants were interested in science and art they had difficulties naming and reading the visualisations. It was concluded that people are interested in visualisation techniques, but have significant difficulties in naming and understanding them.

In order to solve this issue, research has also investigated the way InfoVis novices approach reading data and the methoths that can facilitate their learning. A study used sales data to find the barriers for novices when reading iterative visualisation construction and the way they think about visualisation specifications [17]. They found that the biggest barriers were interpreting questions into data factors, visual mappings, and understanding the visual representations. It was found that there is a need for instruments that suggest possible visualizations, facilitate help with learning, and are integrated with tool support for the whole analytic process. Furthermore, recent research acknowledged that individual differences between people will have influence on the way they interpret graphical representations [31]. They suggested that visualization performance can be improved by personalising visuals according to one's needs, abilities and preferences.

Very little is known about the way users understand and read data visualisations and how they interact with different layouts [11]. It is interesting to explore if there are layouts which provoke better task performance and better response time. Etemadpour and colleagues [11] conducted a research to find out more about performance in similarity based layouts that are generated by multidimensional projections. The results suggested that projection performance is task-dependent and depends on the nature of the data. Therefore, they concluded that the same data layout can have different performance on different tasks and that performance will also be influenced by the characteristics of the data.

1.2 Causal Relationships

Causal relationships are of great interest for scientist who examine influence of different factors on each other. Particularly, research regarding health, social and behavioural sciences aim to investigate questions about causal rather than associative relationships. With the help of statistical analysis associations among different factors can

be inferred. Associations are relationships that can be observed in joint distributions of factors such as regressions and correlations. Causal analysis, in addition, is the practice that aims to infer probabilities under factors that are changing [29]. These could be, for example, changes influenced by factors such as drinking, childhood issues or applied treatments.

A study looking to identify factors influencing blog design used the Decision Making Trial and Evaluation Laboratory method (DALMATEL) which is used to illustrate the relationships between factors and allows causal relationships to be shown [21]. Some of the causal relationships they found were that color arrangement directly impacts simplicity of layout, colour arrangement directly impacts font arrangement, and color arrangement impacts itself.

ReView is a tool for finding causal relationships in anomalies in network traffic. It has been suggested to facilitate better understanding of network representations [37]. One of its features is minimizing the detailed information while showing the causal relationship. ReView can also quickly navigate the user through networks with a large number of requests and levels of abstractions.

Causal relationships are thought to be perceived easier if they are accompanied by animation. Ware and colleagues [35] followed Michotte's perception of causality principle which illustrates a causal event with a billiard ball hitting another ball and causing its motion. They introduced a visual causal vector method that shows a causal relationship between representations using animation. They found that the perception of causality depends on the simultaneous occurrence the visual causal vector and the change in the graphical representations.

1.3   Node-Link Graphs

Node-link graphs consist of nodes which are connected by edges. Large amount of work has been dedicated to visualizing those structures in the most readable way. However, the larger the network that is being illustrated, the higher the possibility that the graph becomes dense and unreadable. This is especially true when the direction of the edges is important [10]. When representing causal relationships with node-link graphs if node n is causing node m, then the edge between those two nodes will point towards node m. In order to avoid the complexity of the large network some researchers have tried to add interaction with the graph [13], while others' intention was to create visuals that do not change the structure of the network and does not require interaction [10]. An innovative interaction technique was designed by Abello et al [1] where a user is able to navigate through a hierarchically clustered base of the graph. In the noninteractive technique researchers decreased the complexity of large directed graphs by replacing single edges with edges which connect to groups of nodes [10].

Data driven journalism is also concerned with difficulties in showing directed relationships in large datasets [28]. Nodes have also been used to represent "visual programming language" or workflow and processing functions [33]. The system translates the user input into "a sequence of data language", which will then be transformed into "service commands" that will be executed. The dependencies between the nodes and data flows are illustrated as interconnections between the vertices on the graphical user interface.

A collider "C" is well represented by an "inverted fork", $A->C<-B$, in which the arrow shows a direct link from the tail factors to the head factor [18]. Stratifying on the collider is likely to induce bias and influence the the association between A and B. In addition, a similar pattern found in directed acyclic graphs, called a confounder, can be illustrated by a "causal fork" - $A<-C->B$. Greenland [18] suggested that any change in the A-B link upon C-stratification will induce bias. Statistical adjustment for collider bias is suggested to lead to as much bias as not making adjustments for a confounder [22].

Directed acyclic graphs (DAGs) are increasingly used in representing causal relationships in modern epidemiology [32]. The factors in DAGs are connected by arrows which can never create a closed loop [19]. A path is defined by a sequence of arrows between the factors of interest. The DAG method is valuable as it is found to show underlying relations explicitly and easily identifies sources of confounding. A confounder is known to be

a variable that is related with the exposure and the outcome but does not exists in the causal path between them. For example, chronic kidney disease and mortality are often caused by age. Confounding occurs when reading the causal relationships between chronic kidney disease and the outcome mortality.

Directed node-link graphs are often used to represent causal data. Dense node-link diagrams make rading graphs harder and colliders and confounders are known for inducing reading biases. However node-link graphs are among the visualisations that do not require extensive learning and are easy and intuitive to read, as they fit the structure of the mental represetation they visualise[27].

1.4   Matrix-based Graphs

An Adjacency matrix is frequently used to represent a network [26]. Matrices can be used to represent both directed and undirected graphs. If a network consists of n nodes, the matrix will consists n x n grid of cells. A new technique called Compressed Adjacency Matrices was introduced in 2012 for visualising gene regulatory networks [8]. As those directed networks have specific structural traits, standard representations such as adjacency matrices, and node-link diagrams are unable to depict all traits. Compressed Adjacency Matrices cut and rearrange adjacency matrix so that no space is wasted in case of sparse network. There are specific structures which represent sub networks. This is how scientists came up with a new data structure in order to fit the characteristics of the data they analyse.

Furthermore, PathwayMatrix is another visualisation tool that represents specific relations between proteins in a pathway [7]. The implementation of the tool consists of adjacent matrices that interact with each other. Additional features were added to facilitate the data analysis. This visualisation software received positive feedback in the specific area of representing relations in proteins pathways. Consequently, there is no one best representation technique. Depending on the dataset specifications, the complexity and size of the data there might be many or only few sufficient ways to visualise it.

A performance comparison between square and triangular matrices has been conducted to measure performance speed and accuracy [25]. It was found that performance is influenced by the matrix juxtaposition type which lead to the creation of a new matrix visualisation called TileMatrix. It represents a large amount of matrices and is effective at analysing networks that are multi-faceted and time-varying. With TileMatrix it is easy to see differences in matrices across time and facets. However, triangular matrices can work only in non-directed networks.

On the other hand, some of the disadvantages of matrix-based graphs are that the area increases quadratically and as large networks are sparse there will be mainly empty space on the matrix. Improvements such as compressed adjacency matrix try to account for some of these disadvantages and in general matrix-based representations are considered to be an unambiguous and concise way of representing data. The main advantages of a matrix-based graph are that it has no overlapping and can be ordered.

Chapter 2

Introduction

Alper and colleagues [3] compared augmented adjacency matrix with node-link visualization in brain connectivity by conducting a controlled experiment. Brain connectivity visualisations are in the form of weighted graphs which are node-link graphs in which each edge is given a numerical weight. They found that matrix-based graphs outperform node-linked graphs. It was concluded that for weight graphs, node-link representations are less readable and more error-prone when compared to matrices. On the other side, node-link graphs are adjustable to a specific spacial representation which might also be insightful when reading the graph. Overall, matrix-based graphs had higher accuracy.

Furthermore, it was suggested that adjacency matrices are superior to node-link graphs in representing dense graphs because they are more compact and easier to look at [30]. However, they also found that node-link graphs are better for path finding as a path can easily be followed if the arrows are not too tangled. Consequently, as mentioned earlier a unique best InfoVis does not exists. Depending on the data size, data characteristics and question of interest different representations can fit the best performing model.

Adjacency matrices and node-link diagrams have been compared in another study [23] to examine which is more suitable graphical representation for the general public. This research was provoked by the fact that matrices are mainly use in engineering area, while node-link graphs are generally a more popular way of visualisation. The research questions being examined were related to the attributes of the connectivity model influencing readability and which representation is more suitable for particular tasks. In addition, they filled the gap in previous research that did not take into consideration participants' familiarity with the data sets. It was found that error rates and response time are highly influenced by size and density. They confirmed Ghoniem's [14] findings that matrix-based graphs outperform node-link graphs for dense and large graphs, and node-link graphs are more readable for small and sparse graphs. The only exception found is finding the path between two nodes in a graph. Furthermore, experience with the data set showed to has effect on performance.

Ghoniem and colleagues [14] made a performance comparison between node-link and matrix-based graphs. They predicted that counting nodes should be equally difficult on types of representations, but counting links and finding the most connected node should be easier in matrix-based representations. Node-link graphs were hypothesised to be better when building a path between two nodes. The reason for this is that matrix based graphs have the nodes represented twice which introduces extra complexity. They also hypothesised that node-link graphs will be easier to work with when dealing with graphs with a small amount of nodes. In their experiment they had three sizes of graphs - 20, 50, and 100 nodes. They tested 36 participants and measured their performance on various tasks such as finding paths, neighbours, nodes and links between nodes. They found that performance in node-link graphs decreases as the size of the graph increases. This pattern was confirmed for all of the tasks except for finding path, where node-link graphs regardless of the size and density showed better results.

2.1   Aims and Objectives

The present research question aims to investigate readability of node-link and matrix-based graphs when representing causal relationships. As discussed above the two types of representations have already been compared for non directed networks, but there is a gap in the literature that discusses their performance in causal relationships. Therefore, it is important to investigate whether their readability in directed graphs will be influenced by the same factors as in non-directed visualisations. What is more, deeper knowledge of the difference in their performance can benefit those working in the social and health policy sectors. They need an accurate and efficient way of representing data in order to make a well-informed and quick decisions. The importance of their tasks is caused by the fact that they can directly influence health policies. In addition, the health and safety sector invest large amounts of money based on their interpretation of the causal relationships graphs they posses.

Therefore, knowing which type of causal visualisation is more efficient and accurate will help the health sector make correct and quick decisions. As discussed earlier there can be a best performing visualisation each type of task. To broaden our understanding of the factors that influence readability different factors can be manipulated during the experiment: type of task, question asked, graph layout and size. In order to find the answer to that research question an experimental software needs to be implemented. It needs to allow the researcher to add the questions of interest with corresponding graphs and the participants to answer them. To measure performance, the software must measure time to answer the question, and whether the answer is correct. The results of the experiment will show whether node-link or matrix-based graph is better when representing causal relationships. Also it will become clear if any significant performance difference is modulated by factor such as layout and size of the graphs.

Chapter 3

Implementation details

3.1 Requirements

In order to test the proposed research question a web application was to be created. This web application was going to be used by participants to give answers to the data visualisation questions and researchers to see the experimental data. Participants were going to see graphs and questions related to them. The participants' response time had to be recorded as well as whether their answer is correct. At the same time, the experimenter need to be able to check the participants' answers to the questions. After considering all requirements, a user story for each feature was created. Some of the most important user stories are:

1. As a researcher, I want to be able to see participant's answers to questions, so that I can analyse the data.

2. As a participant, I want to be able to see a graph and a questions at a time, so that I can complete the experiment.

3. As a researcher, I want to keep participant's scores anonymous, so that the experiment comply with ethics requirements.

4. As a participant, I want to be unable to go the next question before completing the present question, so that I am certain that I haven't missed a question.

5. As a researcher, I want to be able to see participants' demographic information, so that I can make generalisations about my experiment.

3.2 Prototype

After the requirements gathering analysis, development of wire-frames followed. Balsamiq Mockups 3 was the software used for the creation of wire-frames. An example of the experimental question page can be found in figure 3.1.
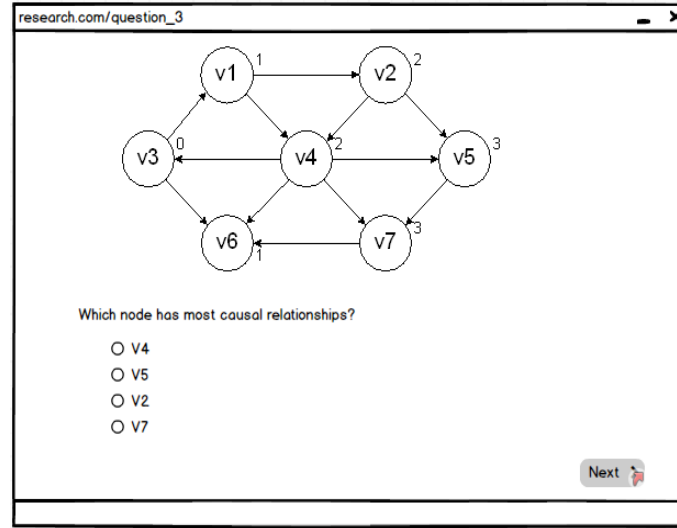
Figure 3.1: Experimental Question wire-frame

Participants were expected to be unaware of how to read the graphs and, therefore, a training page had to be created. It was vital to make sure that the participants are aware of how to read each graph before the actual experiment. This is how the requirements specification became an iterative process during which a better understanding of the product evolved.

3.3 Software tools and technologies

3.3.1 The Spring Framework

To choose the appropriate technologies for this project a research on possible software tools was made. In the beginning the Java framework Spring was decided to be used for the creation of the web service as it is among the most widely used frameworks in industry [36]. This decision was supported by extensive previous experience with Java from developer's point of view and the applicability of the skills to be acquired.

Challenges

However, one of the reasons why Spring is used in industry is because of the large and complex systems that exist there. The Spring framework works on a very high level of abstraction where you can easily write configuration files to add dependencies from different project. Therefore, it is considered rather unfriendly for small independent projects like this one and developers with limited Spring experience. The reasoning behind this conclusion was provoked after issues in setting the relative paths to different CSS and JavaScript files. The issue was found to be in the web application configuration file, but also the realisation that Spring is unnecessary abstract for this project appeared.

3.3.2 Django

As a web application framework that is being thought in university, Django was also considered to be used for the implementation of the web service. Django is a Python-based framework that encourages rapid development. Furthermore, previous experience with the framework might have benefited the development process. However, one of the objectives for this project to learn new technologies and as Django is already a familiar one, it was disregarded as an option.

8

3.3.3 Node.js

Node.js was selected on the grounds of being event-driven, non-blocking I/O model which contributes to a very efficient and lightweight software. A Node.js JavaScript engine is used in the Google Chrome browser and has a very good performance [34]. JavaScript servers have incredible performance due to their asynchronous I/O. Node.js appears to be single-threaded from a developer's point of view, as there is no thread management involved in the development process. However, behind the scenes Node.js handles threading, file system events, implements the event loop, feature thread pooling etc. Coming from a Java background, the Maven equivalent in Node.js is NPM. By using NPM commands the developer is able to install variety of different modules to help the implementation process. NPM executes the function of a package manager. Express is the standard server framework for Node.js. It is usually described as a minimal and flexible Node.js web application framework. Many popular frameworks such as KeystoneJS, Kraken and Sails, are built on Express. Consequently, the decision to implemented the back-end of the application with Node.js and JavaScript combined with the Express framework was taken.

3.3.4 AngularJS

AngularJS 1 was chosen for management of frontend functionality. Even though there is a newer version of the product, the lack of documentation and support online, was a sufficient reason for using the older AngularJS. It uses HTML as a template and enables the developers to extend it to express the application's components clearly. AngularJS supports features such as data binding and dependency injection which decreases the amount of code that a developer would usually write to implement them.

Challenges

An interesting issue appeared when combining AngularJS with Node.js. Usually in AngularJS one uses curly braces to reference data structure from AngularJS controllers. However, Node.js also uses curly brackets to reference information from the back-end in the front-end. This caused a conflict between Anglular and Node.js. An Angular error message appeared, but it was not informative enough to convey the reason for the issue. Later, it was found that Node.js overrides the use of curly braces and the application is not displaying Angular data as it expects references from the Node.js back-end not Angular controllers. Instead of using curly brackets one can also use "ng-bind" and achieve the same result. This approached solved the issue until "ng-bind" information was needed in "ng-src" clause to display the appropriate graph image. It is not possible to use "ng-bind" inside "ng-src" so the found solution at the time was no longer solving the problem. However, as the root of the issue was known, an answer to the question was found and implemented successfully. The Angular configurations had be altered to use a different symbol for data binding not the curly brackets. Implementing this solved the problem entirely.

3.3.5 PostgreSQL

As the size of this project is considerably small a relational database was chosen to be used. PostgreSQL was selected on the basis of being open source and having a reputation of a reliable database system. The database schema looked like this:
questions (question_id, question, answer_one, answer_two, answer_three, answer_four, correct, image, type, layout, domain_question, question_type)
answers(question_id, participant_id, time, correct)
participant_info(participant_id, name, email, degree)
latin_square(participant_id, sequence)

The first table questions had a primary key (PK) question ID, a text field with the actual question, the four

multiple choice answers, the correct answer, the image name, type of graph (node-link or matrix-based), the layout, the theme of the graph, and whether it is a direct question or a path question. The answers table consisted of two foreign keys - question ID and participant ID, the time taken to answer the question and whether it the answer is correct or not. The participant information table was populated by the demographic questionnaire with participant ID being PK, and name, email and degree the rest of the fields in the table. Latin square was used to randomise the sequence of questions for each participant **??**.

3.4 System Design

Figure 3.2 shows an abstract view of the system design. There is an Actor who will either be a participant in the study or a researcher **??**. They will interact with the front-end which will be in the form of a web application in a browser. The front end will communicate with the back-end which will be implemented in Node.js. The back-end will make requests to the database to retrieve and send information.
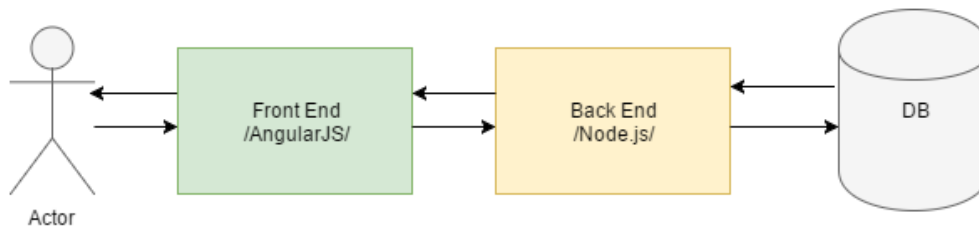

Figure 3.2: An abstract representation of the system design

Figure 3.3 displays a more detailed version of the system design. This design has been implemented to separate the different concerns in this specific system. When the Actor interacts with the application, the front-end will send information to the Controllers. There will be as many controllers as pages with front-end functionality in the web application. The Controller component decides what should the next action be according to the user input. It has control over the front-end logic and sending requests to the Service if information from the database or the server is requested. For example, the answers to all questions are kept in the front-end until a "Submit" button is clicked on. This action triggers a request to the Service. The Service component works with the back-end logic. It can send and retrieve data from the database and keep the information in the Repository. The Service also deals with the requests for the different web pages. It also ensures that the project dependencies are loaded.
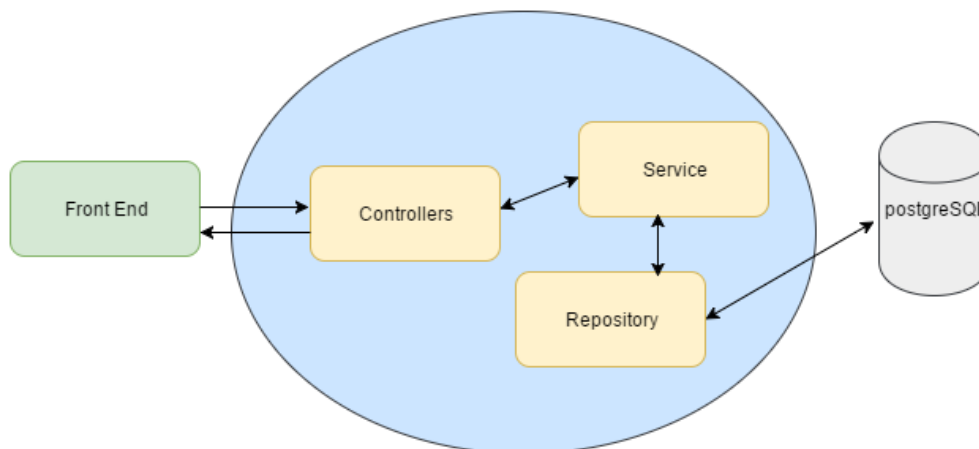

Figure 3.3: A more specific representation of the system design

3.5   Implementation

The development process was split into front-end and back-end. Following the prototype the front-end pages were created. Bootstrap was added to the HTML to improve the UI design and make it look more appealing for the participant. Furthermore, a hierarchical page set up was implemented by having all HTML pages extending one layout file. This also helped to avoid repeated code as the required libraries were imported only once and all other files inherited them.

Setting up a server in Node.js was very well-documented as Node.js is growing in popularity and there are a number of useful tutorials available on-line. After there was a working server, the Express framework was added to the project and the HTML pages were mapped to a Express handlebars or hbs files. Before implementing the front-end functionality the database system was created and connected to the web application. Afterwords, requests for fetching information from the database and sending information to the database were written. Postman was used to check whether the POST and GET requests are working correctly.

Once it was established that the skeleton of the application is working, Angular was used to do the data binding in the front-end. Angular controllers were set to gather the data for each participant. Then they were sending the information to the back-end in the form of JSON object and from there the JSON data was send to the database in the form of database query. For each new participant, there was a participant id created, which then was related to their answers to the experimental questions, and the demographic questionnaire. Cookies were used to keep the user ID throughout the experimental session. Latin square was used to randomize the question sequence for each participant. Node.js has a Latin square module which was used to generate the different sequences. Then the matrix sequences were kept in the database and linked to a participant id ensuring that each user gets a different sequence of questions. Angular was used for populating the website data - graph images and questions. This was done by using one page template for all questions and Angular given the required sequence of questions was displaying the appropriate information.

Challenges

Following the implementation, it was found that the front end structure needs to be changed. Sometimes the size of the large graphs was requiring a scrollbar. This was inconvenient for reading the graphs. Therefore, this was going to be a confounding variable in the experiment. The size of the graph had to be kept the same as the rest of the graphs so that the response time is not influenced by the scaled-down image. The bootstrap elements were splitting the page into separate well-defined components - the experimental heading, remaining time bar, an image section and a question section. After reviewing which is the necessary information on the page only the graph image and the question were left. This also meant that this particular sequence of pages had to extend a different layout page, which does not contain the navigation bar and has a different Bootstrap grid. Luckily, only one HTML page was used for the research questions and Angular was used to populate the information for the next question after the present one was answered. Therefore, very little alteration had to be made to implement these changes. Thus the issue of fitting the necessary information on one page was resolved.

Chapter 4

Experiment

4.1 Design

This is a within subject design experiment with two conditions. In the first condition participants answered questions related to causal relationships in node-link graphs, while in the second one they were asked to answer questions on matrix-based representation. Each condition had four levels - size, layout, question type and question domain. The dependent variables were time taken to answer each question and correctness of the answer. The participants were also given a questionnaire which required subjective answers about their graph preferences and how they enjoy solving logical problems.

4.2 Stimuli

To prepare the experimental stimuli, a variety of graph factors had to be addressed. First, the information complexity in all graphs had to be equal to prevent if from influencing performance time. Consequently, node-link graphs were created in yEd 3.16.1 and then converted into matrix-based representations using an XGML translation software created in Glasgow University. This is how equivalently complex node-link and matrix-based representations were created. As long as the graphs were from the same domain question and size, they represented the same information. However, each question was carefully structured to require different information to be read from the graphs to avoid learning effect.

Furthermore, in order to create a large number of graphs, three different question themes were introduced - drinking issues, exams, and healthy/unhealthy gym behaviour. To gain a deeper understanding of which are the most influential characteristics of a graph different layouts and sizes were also implemented. The three different sizes were small, medium, and large. A small node-link graph consists of 10 vertices and 10 or 9 edges 4.1. The equivalent matrix-based representation has 10 rows, 10 columns and 10 or 9 darker squares showing the causal relationships between factors 4.2. The medium size representations had 20 factors or 20 vertices (node-link graphs) and 20 row and columns (matrix-based graph). The large graphs had 30 variables or 30 vertices in node-link graphs and 30 row and columns in matrix representations.

For each type of representation different layouts were implemented. The matrix-based layouts were alphabetical, indegree descending and outdegree descending 4.3. The alphabetical layout had the row and column factors in alphabetical order. The indegree descending layout finds the number of edges that are coming into each vertex and sorts them in descending order. The outdegree descending layouts calculates the amount of edges going out of each vertex and applies descending order. The node-link layouts were hierarchical, orthogonal, and series parallel 4.4. yEd applies an algorithm for the creation of each layout. The hierarchical layout represents precedence relation in directed graphs. Series parallel graphs have one sink and source which are created recursively to gen-
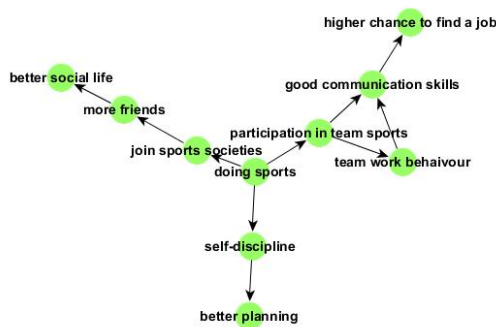
Figure 4.1: Small Node-link Graph

Figure 4.2: Small Matrix-based Graph

erate the series parallel graph. The organic layout has well-balanced spread out distribution of nodes. Figure 4.5 shows the pattern which was followed to create 36 graphs with equal amount of combinations of sizes, layouts and question domains.
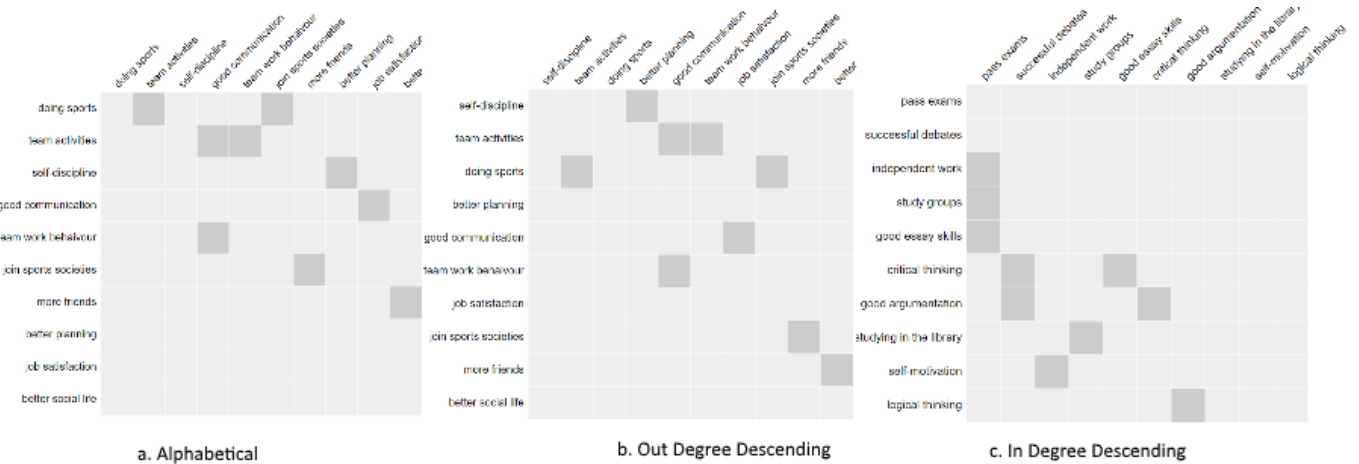

a. Alphabetical

b. Out Degree Descending

c. In Degree Descending

Figure 4.3: Matrix Layouts

There was a direct relationship question and an indirect causal relationship question asked for each of the created graphs. The direct question was requiring the participant to answer a question of the type "Factor X is causing which of the following factors" followed by four multiple choice answers only one of which was correct. The path question was asking which of the following causal relationships is correct, followed by four multiple choice answers of the type "factor A − > factor B − > factor C". The first six questions of the experiment were training questions which did not count towards the final results. The characteristics of the graphs in these questions were chosen to be different from the graphs in the experimental questions. They were added to the beginning of the experiment which generated a total of 42 questions.
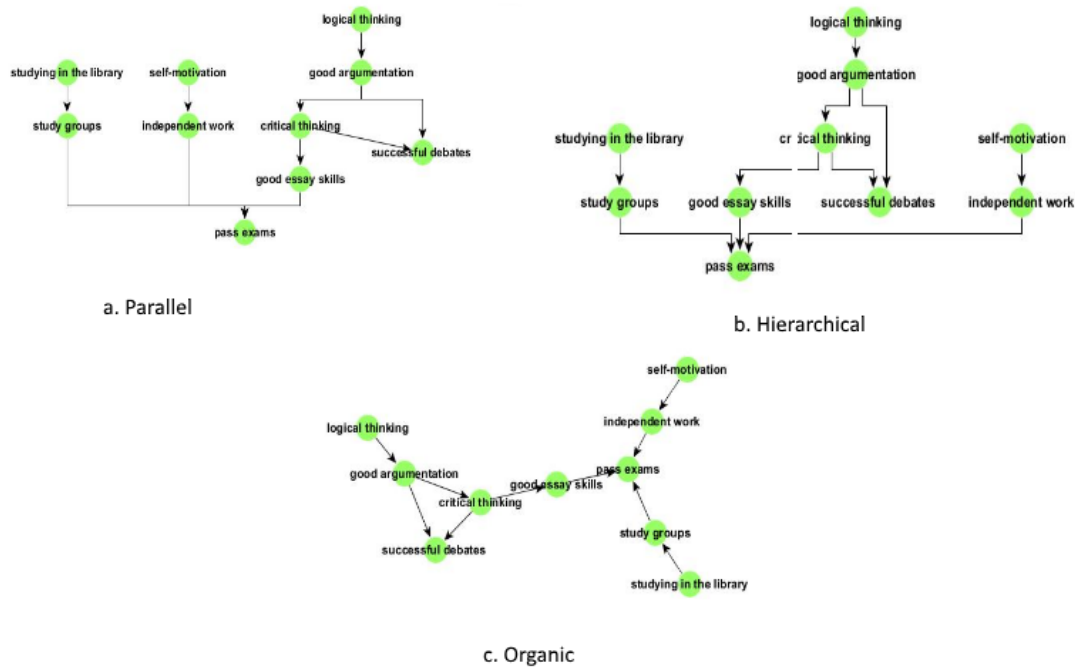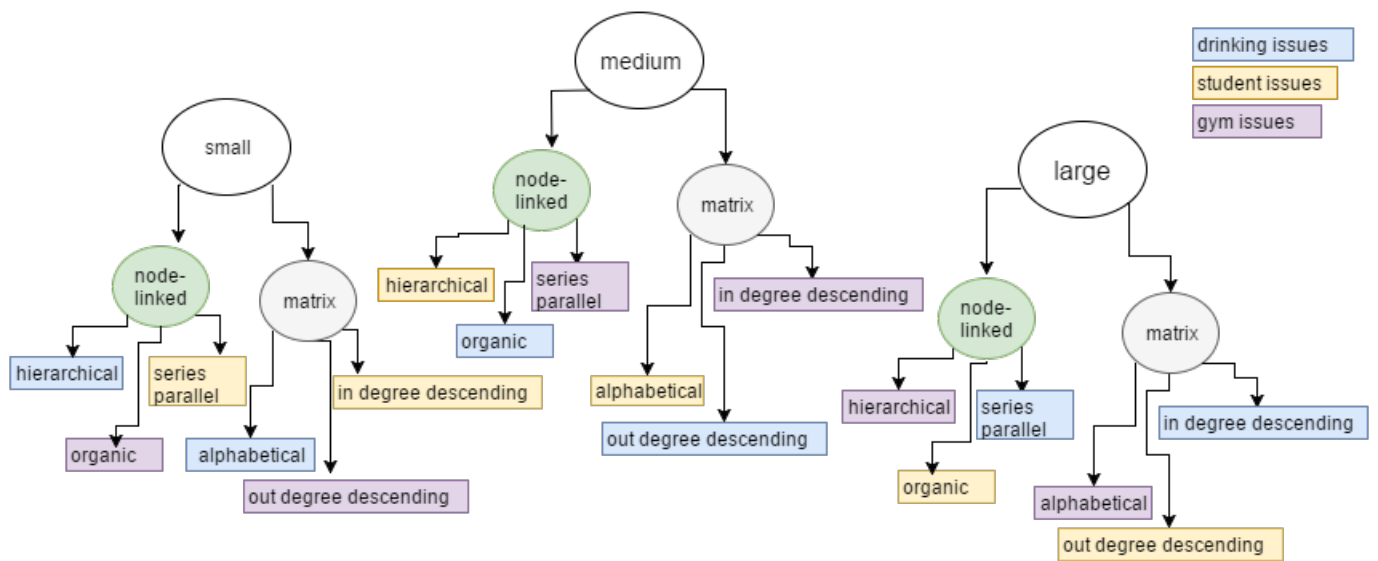
13

Figure 4.4: Node-link Layouts

Figure 4.5: Experimental Design

The specified sequence of questions with matched layouts, question themes, and different sizes was likely to influence the results of the experiment because of the exposure effect of that particular sequence. Therefore, to ensure that all participants receive the questions in a different order, Latin square /citewiner1962latin randomization was used. The sequence of questions was different for each participant and the possibility of the questions sequence affecting the results was eliminated.

All representations were displayed in their real size on the left side of the screen and the corresponding question was on the right. A special attention was paid to the letter's font size and retaining the original size of the image on one screen. Participants were not able to go to the next question before they answered the current one. In addition, to measure answer time more accurately, there was no "Next" button. Once a participant

clicked on an answer it was automatically submitted and the next question was loaded on the page. Figure **??** in Appendix C shows an example of a trial question.

Finally, a Google Form questionnaire was used to ask participants to reflect back on their experience. Two weeks after the testing has started the participants were sent an email with the subjective questionnaire about their graph preference.

4.3 Pilot

Conducting a pilot was an essential part of this study. It brought light into how people who have never been exposed to the graphs and the software before interact with them. It was important to learn whether the tasks are clear and the software is easy to use. Three volunteers took part in the pilot. They were asked to sit on a chair in a quiet environment and complete the experiment starting from the information sheet and finishing with the debrief form. At the end the participants were asked questions regarding the training, the clearness and enjoyment of the tasks. All participants said that the training in the beginning of the experiment is sufficient at explaining how to read the graphs. One of the participants, however, required confirmation that the correct direction of reading the graphs is from left to top. This is why, it was decided to make the first three questions trials in which the participants can ask as many questions as they need to understand how to read the graphs. The second three questions (4,5,6) were also going to be practice questions but they were not going to be explicitly informed about it. Consequently, during the first six tasks the participants were expected to fully learn how to interpret the graphs and their answers were not recorded in the results.

An interesting problem regarding the nature of the trial questions was also considered. Initially, it was suggested that the trial instances are always the first questions of each sequence. However, as those questions would not be accounted for in the results and the Latin Square randomizes each sequence, this was going to lead to uneven number of answers for each type of graph. If this was to happen, the results of the experiment were going to be negatively influenced. In order to avoid this, 6 new questions were created using a combination of layouts, sizes and domains that has not been used for the original 36 questions. A mixture of different sizes and types was included to ensure that the participants have exposure to the main challenges of the experiment during the practice trials. These 6 questions were going to be added at the beginning of each sequence so that all participants are exposed to the same trial questions.

All participants in the pilot said that the tasks are clear, the graphs' font size is readable, and different layouts and sizes are appropriately displaying the data. One participant mentioned that in the node-link organic layout, the label is on the arrow which makes it hard to read. The reason for this is that both labels and arrows are in black colour. Their note was taken into consideration but if changes are made to align the arrow and the label differently this would alter the organic yEd layout. Consequently, the results will provide information about a manually created layout similar to the organic yEd layout. Another solution of this issue was to change the label names to individual letters like "A" instead of actual factor labels such as "depression", and "pass exams". However, this was going to influence the complexity of the experiment and the participant's level of interest. Thus, no changes were made to the organic layout.

One volunteer in the pilot mentioned that sometimes it is hard to find the label you are looking for and they guessed the answer to the question by logical reasoning. To avoid this problem in the experiment, a note was made to inform participants that there is no logical relationship between the causal relationships and they should not attempt to guess the correct answer. This is supposed to encourage them to look for the labels rather than guess the correct answer.

The pilot was also extremely helpful for spotting technical issues. If it was not conducted and some of these problems were not accounted for, the recorded results would have been incorrect. First, it was found that in one question the correct answer was being evaluated to incorrect because of a typo in the csv file that was used to populate the database. In addition, the answers were not written in a consistent way, which meant that

the participants might had been influenced to choose the answer that is written differently from the rest of the answers. In order to fix these problems, the database table with the questions had to be altered.

Another technical issue found was concerned with the time recorder. It was found that the timing was starting and stopping when required, but the record of the time taken for a particular question was wrong. The issue was found to be caused by a wrong startTime variable during the timeTaken calculation. This was a scope issue, which was fixed for the real experiment. Moreover, in order to submit their answers the participants had to click a "Next" button. The timer used to stop once this button is clicked, record the time taken for the current answer, and start the timer for the next question. However, it was found that this extra click influences the recorded time and it would be more appropriate to have the time start and stop when the participants choose an answer. This led to the decision to completely remove the "Next" button and submit the answers when one of the radio buttons has been selected. On that click the next graph and question would be loaded. It was not expected from the participants to expect a submission of their answer on their first click so thorough instructions about this feature were added in the information sheet. In addition they were to have six trial questions to get used to that functionality.

In general, the pilot helped in identifying small issues regarding the graphs layout and the manner in which questions were asked. Other bigger issues, that were going to influence the results, were also found such as typos leading to wrong evaluation of correctness and inaccurate time recorder. The pilot also inspired a new way in which the trial questions should be created and accounted for. The present experiment would not have been as accurate and precise if the pilot was not conducted.

4.4 Participants

There were 30 participants who took part in the experiment, aged between 20 and 29 years. The excluding criteria restricted people specialized in the subjects of Maths, Engineering and Computing Science from participating. The reason for this is that these fields are likely to include preparation in reading graphs. In addition, findings are expected to be representative for Health and Social sector employees, who are not likely to have background in those degrees.

4.5 Procedure

In order to have a better controlled experiment, the study was conducted only on the experimenter's laptop in their presence. The participants were asked to sit on a chair and complete the experiment in a calm environment. They were recruited either in Glasgow University library or by posts in social media websites. The experiment was conducted in GU library in quiet group study areas. The participants had to read through an information sheet, explaining what is to follow in the next 30 minutes, a consent from, asking them to give their written informed consent, and a training showing them examples of how to solve the upcoming 42 research questions. At the end of the experiment participants were shown a short demographic questionnaire and a debrief form. Two weeks after the testing has started they were sent another questionnaire asking about their graph preferences.

4.6 Results

4.6.1 Data Analysis Methodology

Correct data analysis is vital for the successful completion of a research based project. This is why a pilot of the data analysis methodology with fake data was completed before all participants were tested. In order to achieve this the results of 20 participants were faked. As this is a repeated measures experiment the data layout had to be in wide format with each participant's data represented on each row. However, the way the database was being populated was in long format. Therefore, data wrangling had to applied before the data was in a format

able to be statistically analysed. The data analysis was done with R Studio which provides a package for data wrangling. However, as there were a number of different factors defining each answer, and thousands of lines of raw data, it was decided that writing SQL queries would be a more appropriate way to deal with the data layout. For each type of comparison, there was a query written to aggregate the needed information in a format ready to be inserted in R Studio.

Testing the date for outliers was the first part of the analysis. Then, Shapiro-Wilks Normality test was conducted to check whether the data is normally distributed. The test showed a p-value of less than 0.05 for the fake data which rejected the Null hypothesis that the data is normally distributed. Histograms were plotted using the hist() function. This was found to be helpful to visualise the distribution of the values. If the data values were normally distributed then a repeated measures t-test was going to be run to see whether there is a significant difference between the time taken to read the two types of graphs. However, as the data was found to not be normally distributed, a non-parametric test was applied. The non-parametric equivalent of repeated measures t-test is Wilcoxon.

The type of representation may be the main aspect being investigated, but there are other characteristics of the graph that might influence performance. In order to investigate whether layout and size have any effect, a different statistical test needs to be used. For non-parametric data this test is Friedman followed by post-hoc comparison in case of significant difference. The data was faked so that a significant result was found and there was a need to investigate which are the pairs of comparisons significantly different from each other. A function called friedmanmc(), was found to compute this comparison, but it returns True or False, not the actual p-value. Therefore, a new function was found but inconsistency between the two functions was found. After a deeper investigation it was found that the second function is not calculating the adjusted p-value which was leading to a difference in results. In order to account for this Bonferroni adjustment was added as a parameter to the calculation. This is how the process for calculating significance between 3 non-parametric factors was finalised.

The fake data analysis took longer than expected due to the unforeseen circumstances of data layout difficulties, inexperience with R Studio, and statistical analysis of large amounts of data. However, because of this process, a clear layout of the analysis for the real data was prepared. All steps of the process were clear beginning with SQL queries to aggregate data, to inserting data in R Studio, and running all statistical tests.

4.6.2 Experimental Results

The data analysis was first done for direct questions and then for path questions. The analysis was split into these groups, as they were found to be two separate tasks requiring two separate types of computations and showing different performance time. Considering that for the direct question one needs to find only one causal relationship, while for the path question - three, it is only logical to split the data in two independent parts.

Direct Questions

The research nature of the experiment allowed for thorough data analysis of different factors in the graphs. The results showed that regardless of the graph layout and size the matrix-based graphs (M = 14.15) have better performance than node-link graphs (M = 16.85) for direct questions (Fig4.6) with p-value = 0.0128 (Table4.1). When different sizes of each graph type were compared significant difference was found only for the large graphs comparison with matrix-based large graphs (M = 16.03) being easier to read than node-link graphs(M = 24.76). Figure4.6 shows the mean values of each each representations. Wilcoxon test showed a p-value of 5.633e-07.
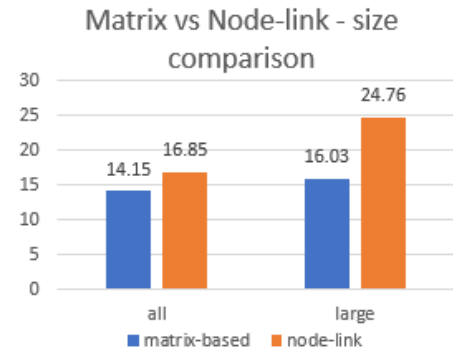

Figure 4.6: Matrix vs Node-link

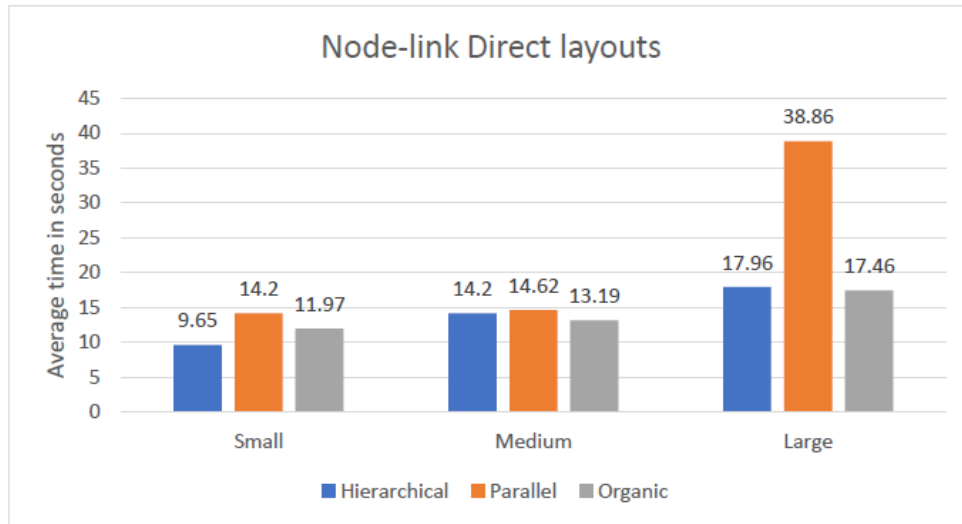Node-link layouts were compared in each size group. Figure4.7 illustrates the difference in mean values in

Figure 4.7: Direct Question: Node-link layout comparison in all size groups

all node-link layouts and sizes. In small size the parallel layout was found the have the worst performance (M = 14.20) and hierarchical layout was found to have the best performance (M = 9.65). Friedman test calculated a p-value of 0.0013* with significant post-hoc tests between all pairs (Table4.1). In the medium size measures, the parallel layout had the worst performance time of 15.62 on average, but when Friedman test was conducted no statistical difference between the three layouts was found with p-value = 0.6575. The node-link large size performance time for hierarchical (M=17.96), parallel (M=38.86), and organic (M=17.46) layouts 4.7 was significantly different and has p-value = 3.261e-06*.

The matrix-based based graphs were not found to have consistent performance differences between the various layouts when direct question was asked. Significant difference was found in the medium size with out degree descending layout (M=12.70) outperforming in degree descending (M=14.64) and alphabetical layouts (M=14.86). As far as the small and the large sizes are concerned p-value was shown to be insignificant.

Table 4.1: Direct Question comparisons

Matrix			Node-link			P-value
14.15			16.85			0.0128*
small 10.78			small 11.92			0.1055
medium 14.06			medium 14.01			0.8776
large 16.03			large 24.76			5.633e-07*
			small			
			H 9.65	P 14.20	O 11.97	0.0013*
			H	P		<2e-16*
				P	O	1.8e-09*
			H		O	3.5e-05*
			medium			
			H 14.20	P 14.62	O 13.19	0.6575
			large			
			H 17.96	P 38.86	O 17.46	3.261e-06*
				P	O	<2e-16*
			H	P		<2e-16*
			H		O	0.001*
small						
inDD 10.35	outDD 9.25	alpha 12.75				0.07939
medium						
inDD 14.64	outDD 12.70	alpha 14.86				0.03943*
inDD	outDD					2.4e-10*
	outDD	alpha				7.5e-07*
inDD		alpha				0.12
large						
inDD 15.34	outDD 14.76	alpha 18.00				0.355

H - hierarchical P - parallel O - organic inDD - indegree descending outDD - outdegree descending alpha - alphabetical

Path Questions

The path question results showed different pattern. The Wilcoxon comparison between the matrix-based (M=35.62) and node-link (M=31.81) representations of any size and layout showed a better performance for node-link graphs 4.8 with p-value=0.02479*. In the small group size of this question type, matrix-based representations (M=23.09) were outperformed by node-link graphs (M=27.83). Wilcoxon test reported a p-value of 0.03997* (Table4.2). No difference was found in the medium size graphs, but in the large size group matrix-based graphs (M=47.38) showed worse performance when compared to node-link graphs (M=39.51). Wilcoxon test found a p-value of 0.02847*.

Figure 4.8: Matrix vs Node-link

The node-link layout comparison showed that for small sizes the hierarchical layout (M=27.31) has worst performance, followed by the parallel layout (M=22.43) and the organic layout (M=19.62). Friedman test found a significant difference between the layouts with p-value=5.943e-06*. However, in the medium size group the layouts showed different results (Figure4.9). The organic layout (M=48.55) performed significantly worse than the parallel (M=26.44) and
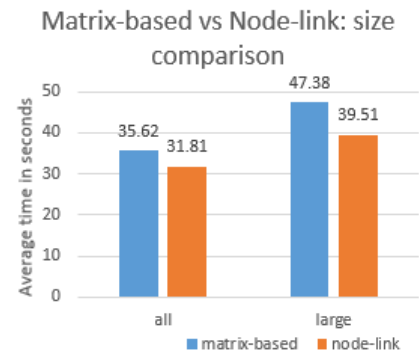
the hierarchical (M=23.54) 4.2. These results were consistent with the large group size layouts where the organic layout (M=47.88) showed worse performance than parallel (M=39.20) and hierarchical (M=31.44) layouts.
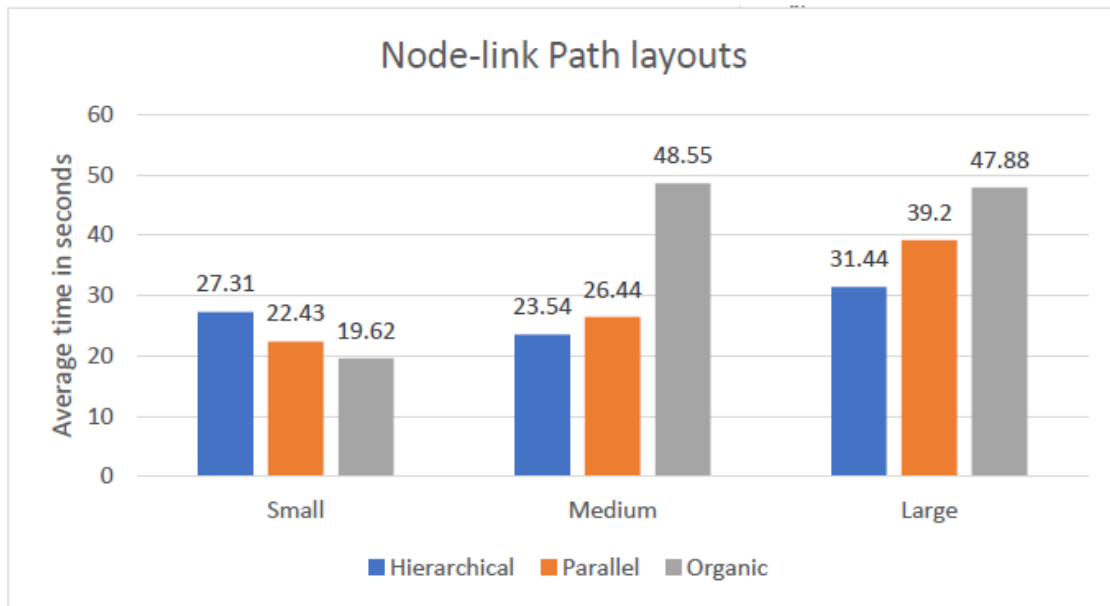

Figure 4.9: Path Question: Node-link layout comparison in all size groups

Among the matrix layouts significant difference was found in the small and large size groups. When the size is small, alphabetical layout (M=23.79) outperformed out degree descending (M=29.50) and in degree descending (30.20) layouts 4.10. A Friedman test found a significant difference between the three layouts with p-value=0.0003245*. In the medium size group, the alphabetical layout was confirmed to have best performance (M=27.29) when compared to in degree descending layout (M=29.07), but not statistical difference was found when compared to out degree descending layout (M=28.68) with p-value= 0.26. No significant difference was found among the layouts in large size. All p-values for path question can be found in Figure 4.2
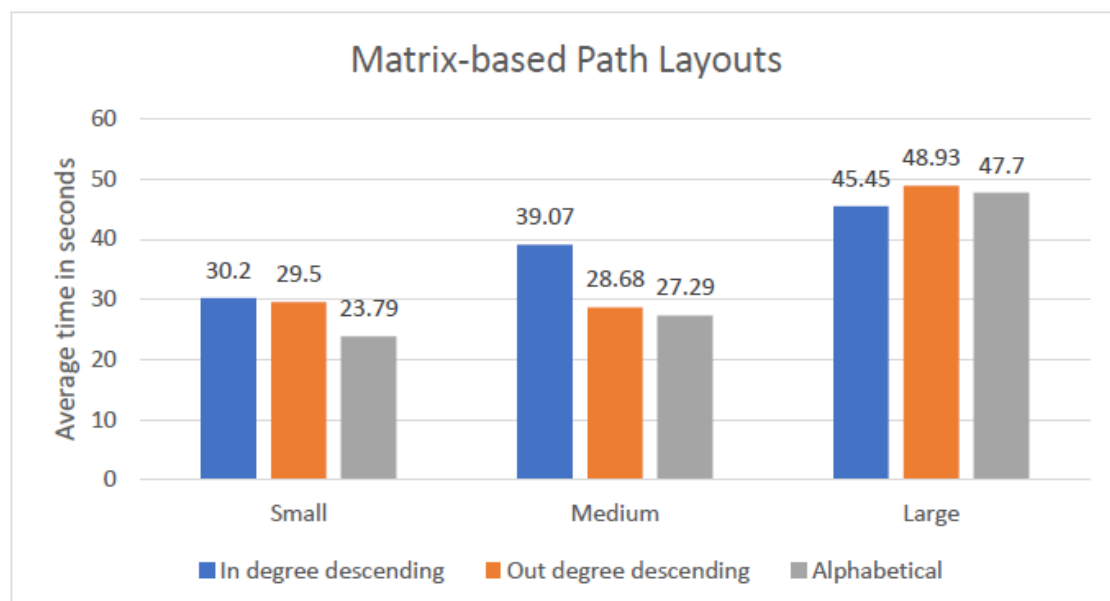

Figure 4.10: Path Question: Matrix-based layout comparison in all size groups

Table 4.2: Path Question comparisons

Matrix			Node-link			P-value
35.62			31.81			0.02479*
small 27.83			small 23.09			0.03997*
medium 31.67			medium 32.85			0.2665
large 47.38			large 39.51			0.02847*
			small			
			H 27.31	P 22.43	O 19.62	5.943e-06*
			H	P		4.5e-12*
				P	O	4.5e-12 *
			H		O	<2e-16*
			medium			
			H 23.54	P 26.44	O 48.55	3.08e-09*
			H	P		2.40e-13*
			H		O	<2e-16*
				P	O	<2e-16*
			large			
			H 31.44	P 39.20	O 47.88	0.0273*
				P	O	1.1e-09*
			H	P		1
			H		O	1.1e-09*
small						
inDD 30.20	outDD 29.50	alpha 23.79				0.0003245*
inDD	outDD					1.0e-09*
	outDD	alpha				<2e-16*
inDD		alpha				1.7e-07*
medium						
inDD 39.07	outDD 28.68	alpha 27.29				0.001454*
inDD	outDD					4.5e-13*
inDD		alpha				7.7e-16*
	outDD	alpha				0.26
large						
inDD 45.45	outDD 48.93	alpha 47.7				0.356

H - hierarchical P - parallel O - organic inDD - indegree descending outDD - outdegree descending alpha - alphabetical

4.7 Preference Results

The participants' answers to the questionnaire brought more light into their graph preference and experience during the experiment. It was found that the majority of people (80%) would prefer to work with node-link graphs. On the question asking how easy they found working with node-link graphs the participants reported an average of 2 (1="Very easy", 5="Very hard") while for the matrix-based representation an average rating of 2.9 was found. These findings suggest that on average people found working with node-link graphs easier which is consistent with the results for graph preference. On the question whether they enjoy solving logical problems the participants scored an average of 3.8 (1="Not at all" and 5="Very much"). Their liking for Maths on the same scale was averaged to 3.3 meaning that the participants had positive attitude towards problem solving and mathematics. On the last question of whether there was something stopping them from performing at their best most people said "No". However, some participants said that they had issues finding the label they were looking

for, some mentioned that they might have experienced difficulties paying attention, while one person just said that they did not find the task particularly interesting. One participant mentioned that the location of the experiment was an issue as there was noise in the environment that was preventing them concentrating properly. Even though the experiment was conducted in the library's quiet area, there were people unaware that different levels of the library require different behaviour, and caused difficulties for this one participant.

4.8 Qualitative Results

The experiment was done in the presence of the researcher to ensure high environmental validity and also to allow for observations while the participants were doing the tasks. For example, while one of the participants was working on the tasks their telephone rang and they got distracted from the tasks. However, a timer was started to measure how long it took the participant to come back to the tasks and then this time was subtracted from the total recorded time for this question. Additional issue which arises from this event is that it is impossible to measure whether the participant had to read the question again when they came back from their distraction or they were able to continue from where they stopped. Fortunately, this happened only once, and the time before the distraction was added on to the time after the distraction.

In general, there was positive feedback about the questions asked and the nature of the topics. Most people found them very familiar and were interested to find the causal relationships. A small amount of people (N=3) mentioned that the font-size of the labels was small and it was difficult to follow the matrix row and columns on the large graphs because of the small size of the squares. The squares and font-size were made as large as possible, but due to the big amount of information that had to be fitted on one screen, some participants still experienced difficulties. One of these participants said that it might have been more difficult for them as they could have eyesight issues. Other participants were explicitly asked after the experiment whether they had experienced any issues and they reported that everything was fine.

Chapter 5

Discussion

5.1 Results

The experimental findings showed that for direct questions matrix graphs have better performance than node-link graphs, but for path questions the opposite was true - node-link graphs outperformed matrix-based representations. Direct questions required the participants to find a label and check which factor is being caused by this label. The results of the experiment suggests that it was easier for the participants to find this label on the matrix-based graph. Considering the layouts of the two representations, the matrix has a row and a column while the node-link layout is rather spread out and unevenly distributed on the screen. Consequently, a possible explanation for the better performance of matrix-based graphs for direct questions can be that it is easier for the human eye to find a label in a row rather than on a flat space where factors are spread out in different directions.

When path questions were concerned node-link graphs outperformed matrix-based graphs. In order to answer a path question in matrix-based graph two selected cells had to be found which could be positioned on two completely different places in the matrix. Therefore, it would take time for the participant to find these two cells. For node-link graphs, however, the participants were required to follow the link between nodes that are positioned next to each other. Therefore, they had to spend time finding only one of the factors. Our results confirmed the findings of Zekian et al [30], who also concluded that node-link representations are better than matrix-based graphs when answering path questions.

The layout analysis did not find a consistent best performing layout in node-link and matrix-based graphs. However, a consistent bad performance from particular layouts depending on the size of the graph and type of representation was found. First, in node-link direct questions, the parallel layout was found to take significantly more time to read than the rest of the layouts (Fig. 4.7). A deeper investigation revealed that a possible reason for these findings is that the edge length in the parallel layout is longer than the rest of the layouts. To illustrate this, edge length in the causal path in the small early age drinking graph between "drinking parents", "early age drinking", and "depression" in the three different layouts was measured. The edge length in the hierarchical layout was 2.9cm, and the edge length in the organic layout was 3, while the length in the organic layout was 11.9cm. In order to show the difference between the layouts an example that shows the biggest difference in length in this particular graph was chosen. It must be noted that the length difference is not that large in all causal relationships. However, the edge lengths in the parallel layout varies greatly, while the edge lengths in the hierarchical and parallel layouts are very similar - between 1cm and 1.5cm between two nodes.

Consequently, it was suggested that the length of the edge is a direct predictor of performance. A controlled experiment examined readability of graphs while manipulating the length of the edges {holten2011extended. There were two edge lengths - short and long. It was found that it took participants significantly longer to answer questions on graphs with long edges when compared to their response time on the short-edge questions. As a

result, this study brings a possible explanation for the bad performance of the parallel layout.

Interestingly, however, these results were not found in the path questions. There the parallel layout out-performed the organic one. It was found that in medium and large graphs, the organic layout has significantly worse performance than hierarchical and parallel graphs. Considering the similarity in the mean values between hierarchical and parallel performance, we can investigate the differences between organic layout and hierarchical/parallel. The main visual difference is that the organic layout is spread out in all directions, and the direction of the edges does not follow a particular patter. However, in parallel (Fig. **??**) and hierarchical (Fig. C.15) graphs the nodes are ordered into parallel rows and the edges always point from the upper level to the lower level

study shows that I am a little kitty

5.2 Node-link readability

The readability of a node-link graph is characterized by readability metrics. These are means that define the understandability of the graph based on number of nodes and overlapping edges. A well-known method for inducing better readability of node-link graphs is spreading out the nodes evenly, keeping the edges away from the nodes, and edges-lengths equal, and avoid overlapping [9]. However, applying an algorithm to make these amendments to a visualisation might lead to a misleading drawing with changed spatial representation.

5.3 Future work

5.4 Conclusion

Appendices

Appendix A

Database Schema

```
participants_answers(question_id, participant_id, answer, time)
participants_info(participant_id,participant_name, email, uni_degree, age)
questions(question_id, question, one, two, three, four, correct, image)
```

Appendix B

Running the Programs

An example of running from the command line is as follows:

```
> java MaxClique BBMC1 brock200_1.clq 14400
```

This will apply $BBMC$ with $style = 1$ to the first brock200 DIMACS instance allowing 14400 seconds of cpu time.

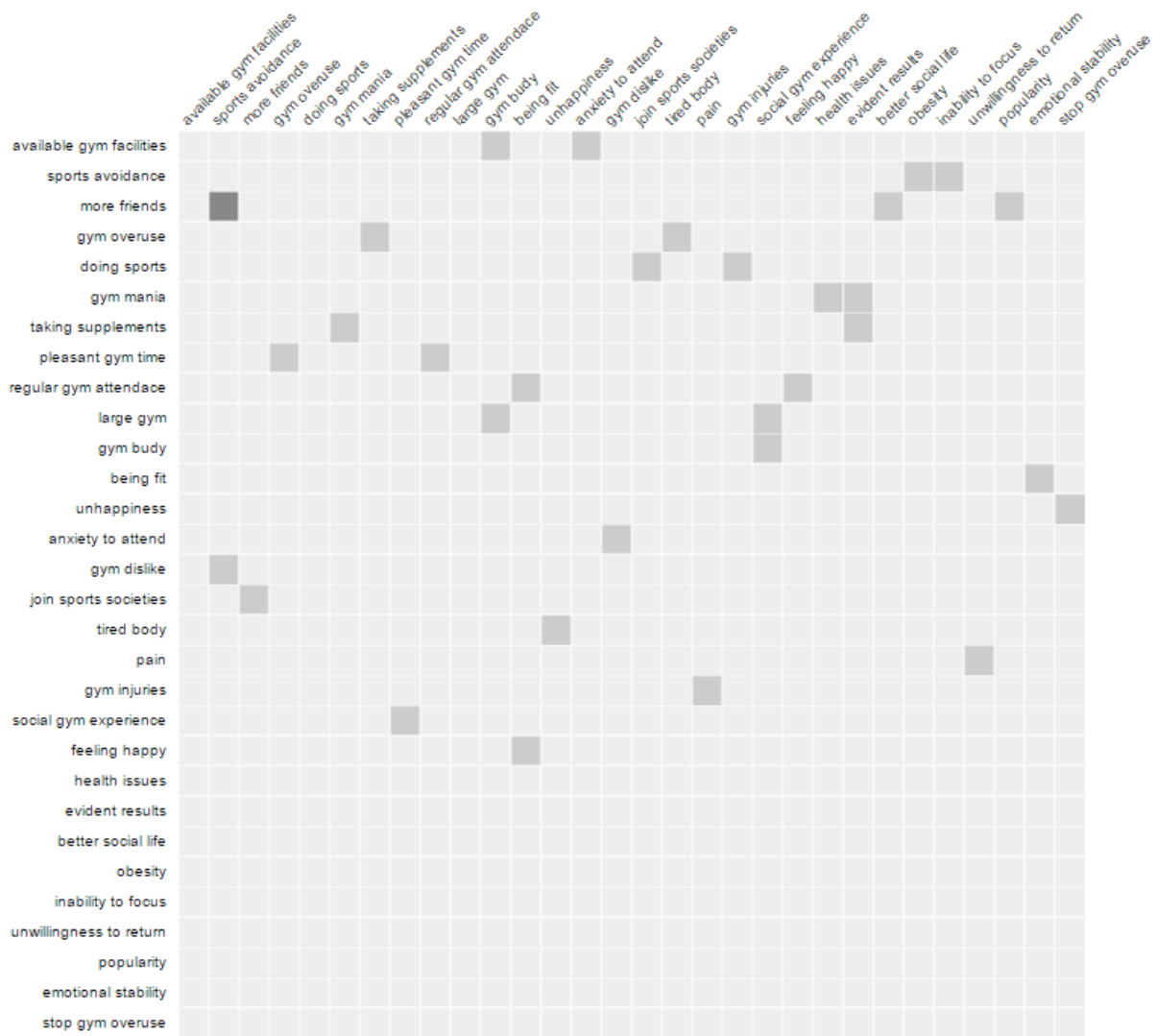# Appendix C

Experimental Stimuli

C.0.1 Matrix-based
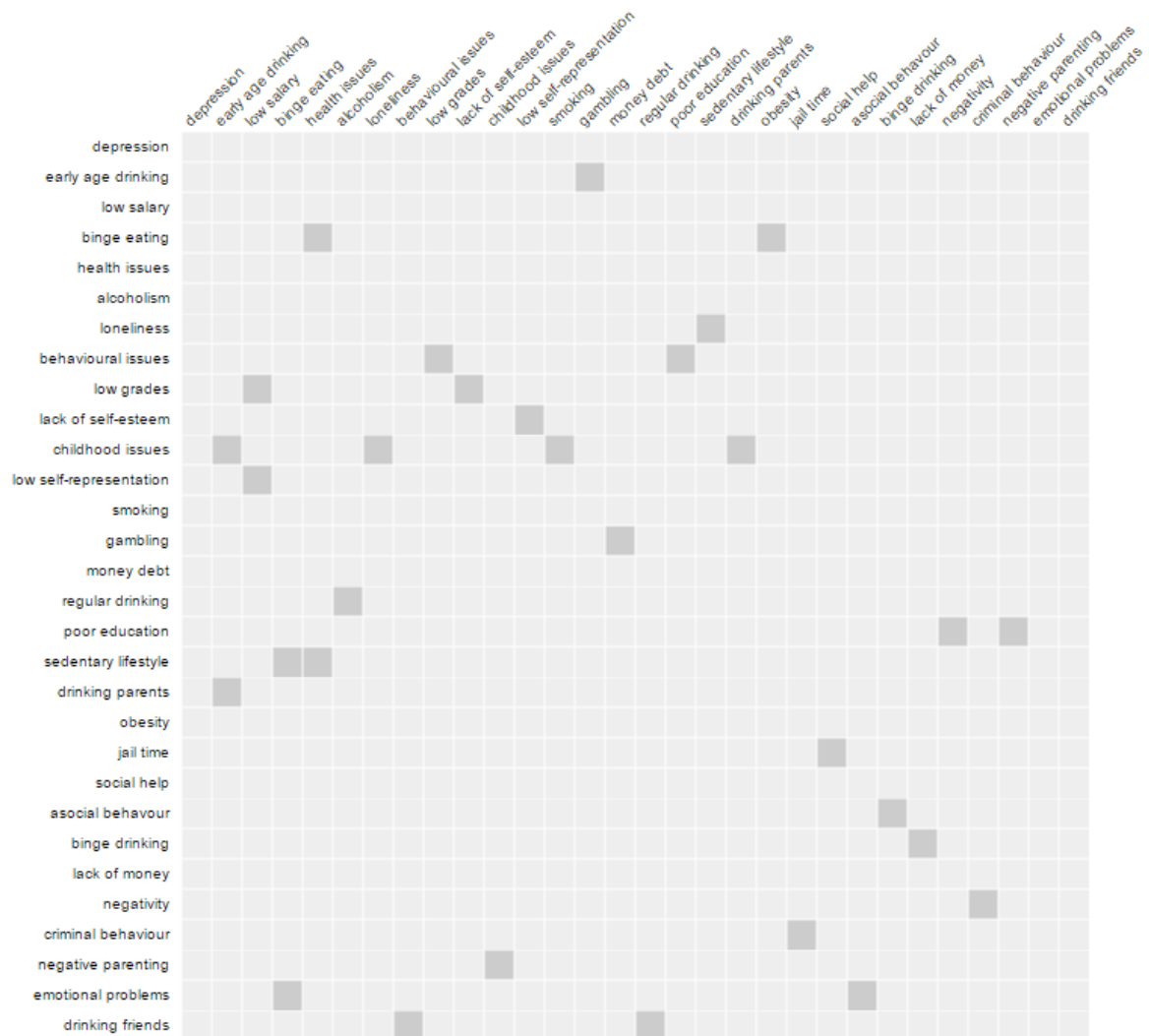
Figure C.2: Large matrix-based graph in in degree descending layout and drinking issues domain question
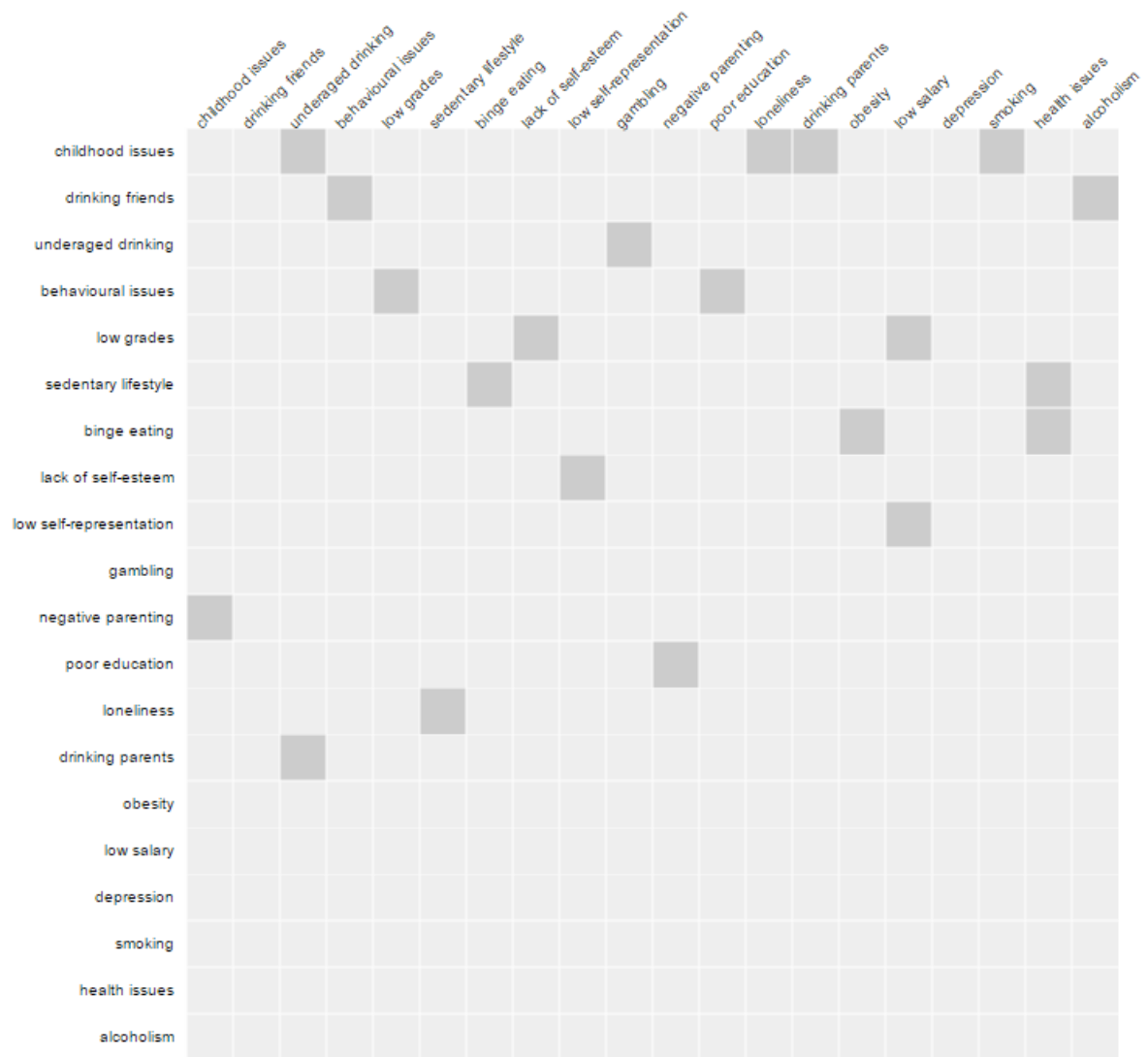
Figure C.3: Medium matrix-based graph in out degree descending layout and drinking issues domain question
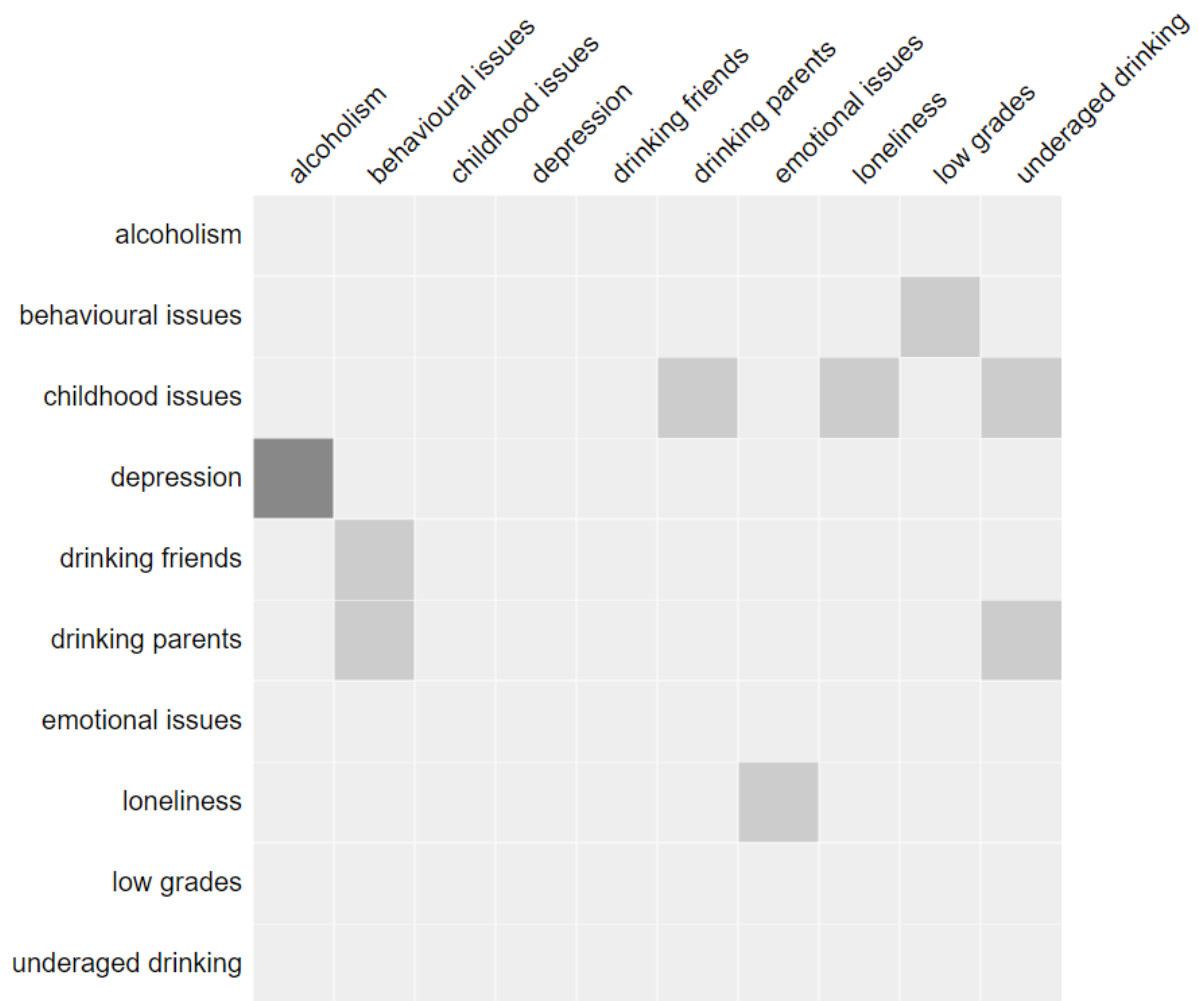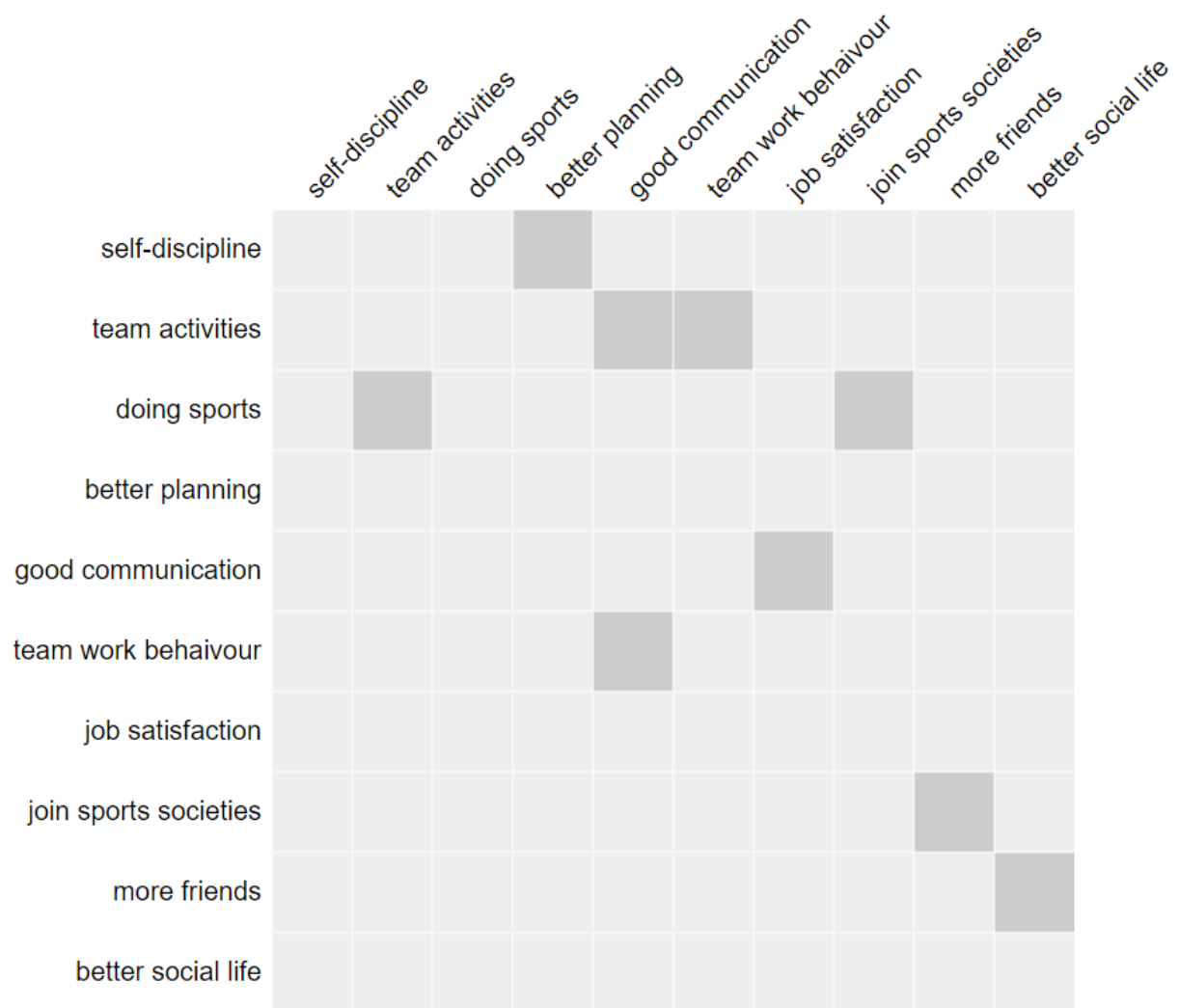
Figure C.4: Small matrix-based graph in alphabetical layout and drinking issues domain question
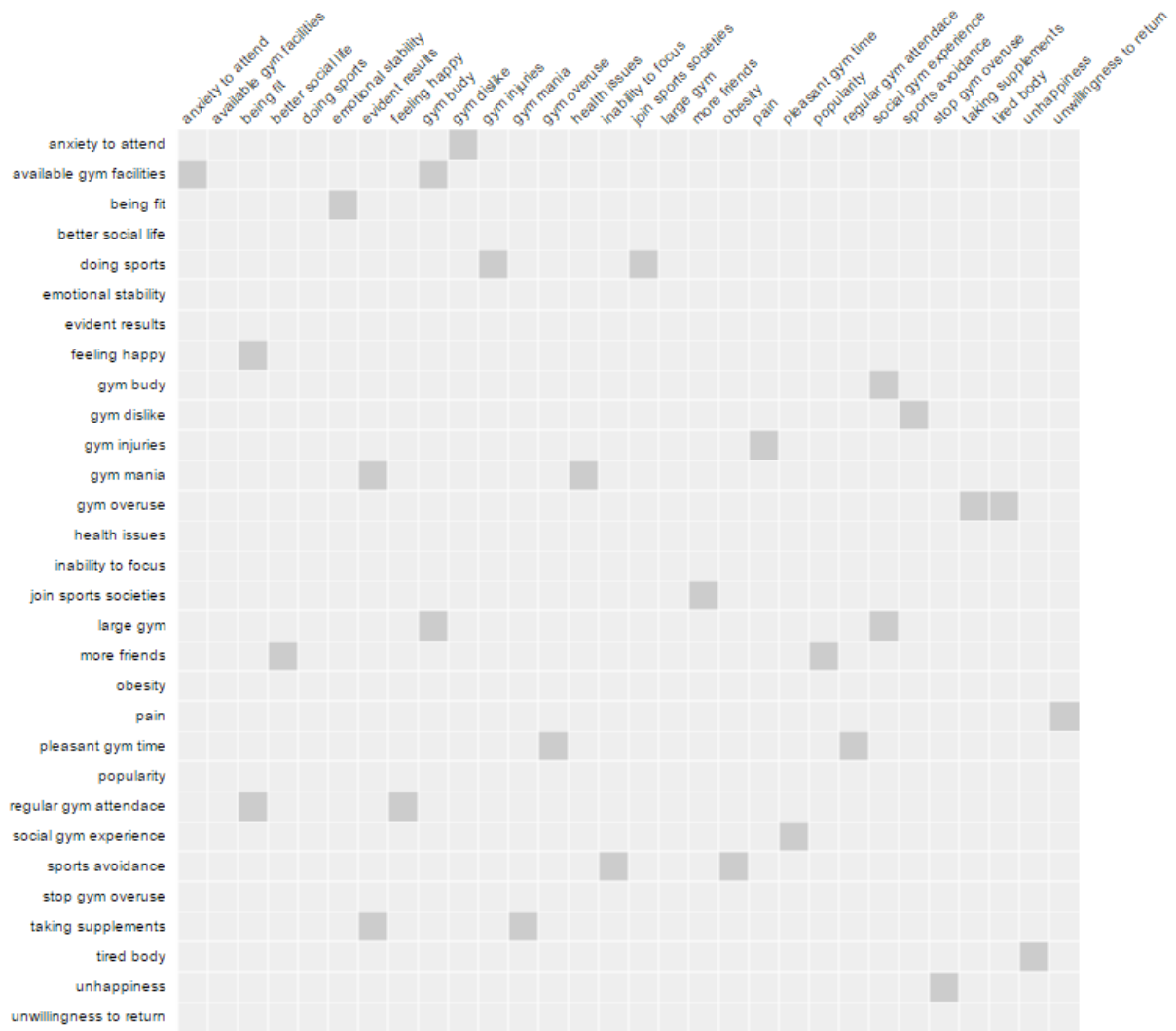
Figure C.5: Large matrix-based graph in alphabetical layout and sport domain question

Figure C.6: Large matrix-based graph in out degree descending layout and drinking issues domain question

Figure C.7: Medium matrix-based graph in in degree descending layout and sport domain question

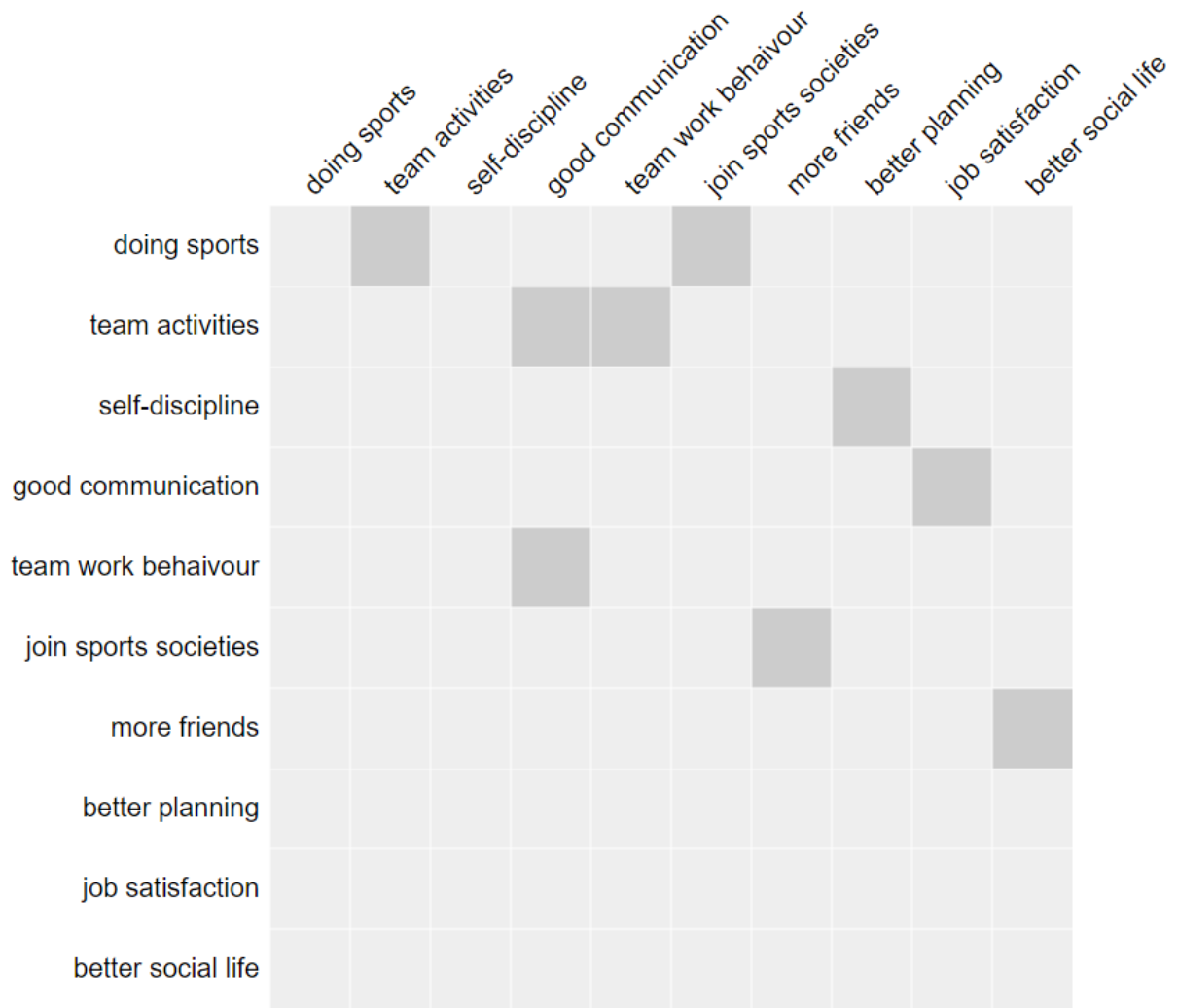Figure C.8: Small matrix-based graph in alphabetical layout and sport domain question

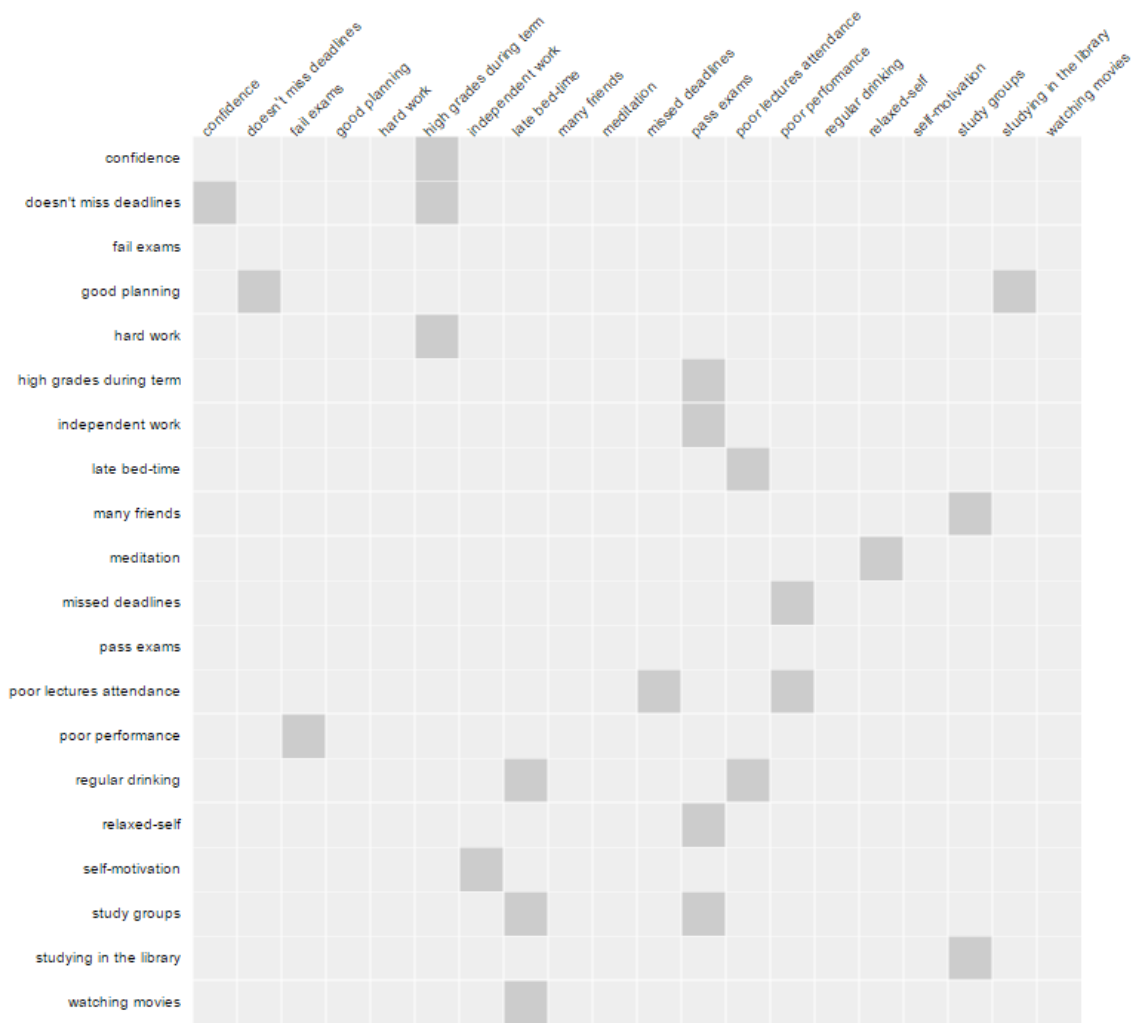Figure C.9: Small matrix-based graph in out degree descending layout and sport domain question

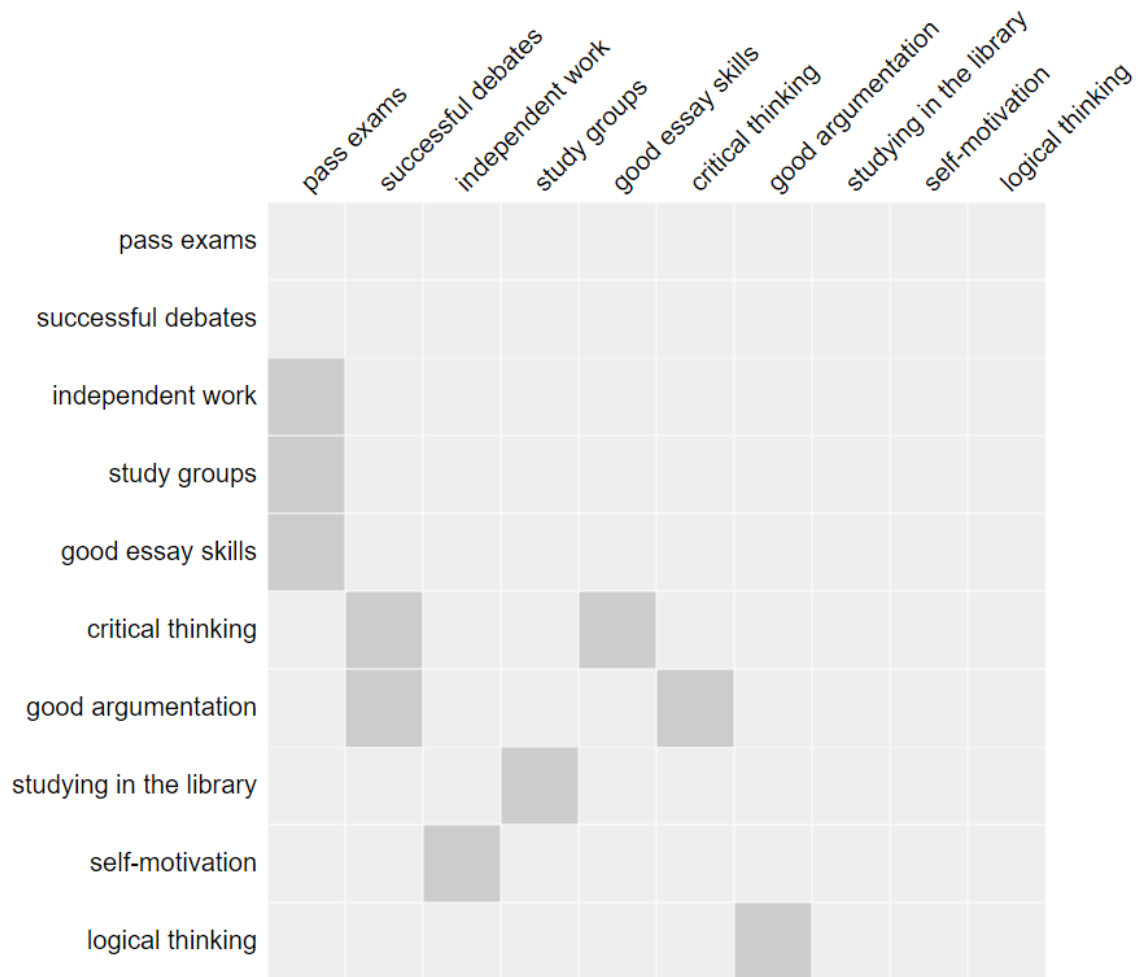Figure C.10: Medium matrix-based representation in alphabetical layout and student domain question

Figure C.11: Small matrix-based representation in in degree descending layout and student domain question
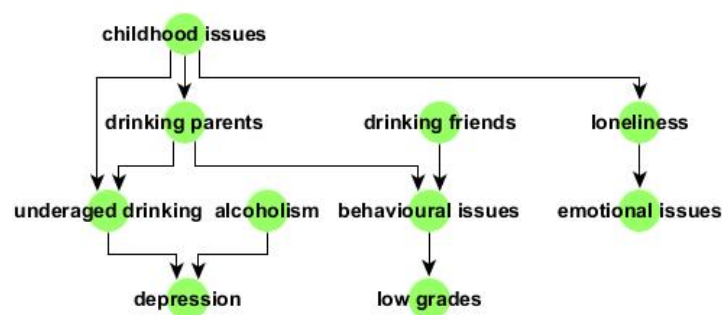
C.0.2 Node-link

Figure C.15: Small node-link graph in hierarchical layout and drinking issues domain question
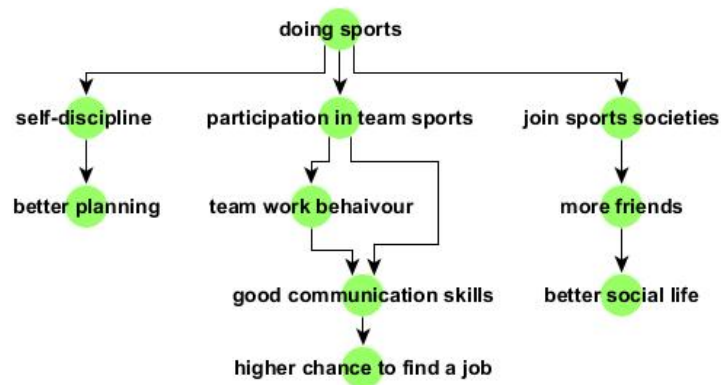
Figure C.18: Small node-link graph in hierarchical layout and sport domain question
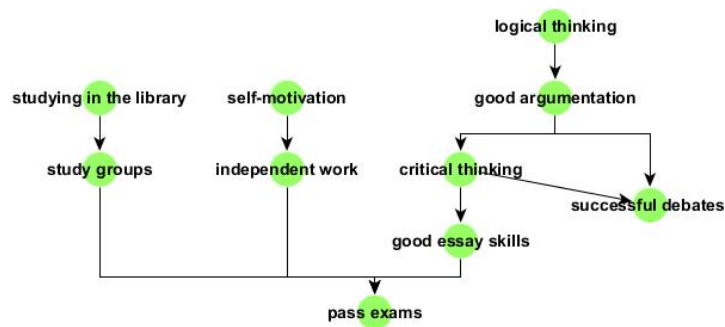

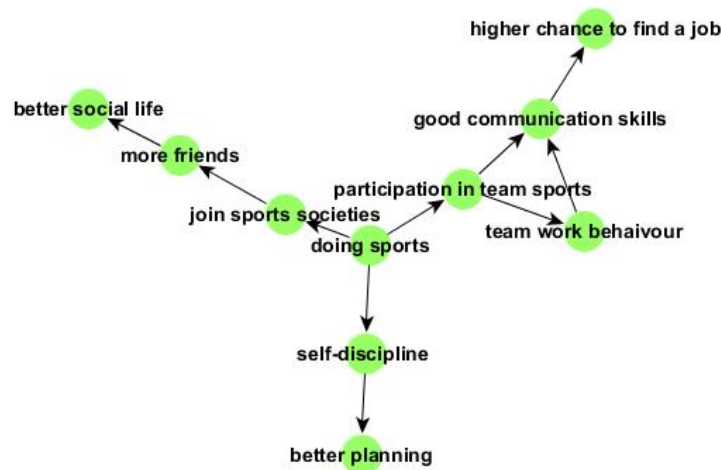Figure C.19: Small node-link representation in parallel layout and student domain question

Figure C.20: Small node-link graph in organic layout and sport domain question

Figure C.21: Large node-link graph in organic layout and student domain question

Bibliography

[1] James Abello, Frank Van Ham, and Neeraj Krishnan. Ask-graphview: A large scale graph visualization system. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):669–676, 2006.

[2] Jasser Al-Kassab, Zied M Ouertani, Giovanni Schiuma, and Andy Neely. Information visualization to support management decisions. *International Journal of Information Technology & Decision Making*, 13(02):407–428, 2014.

[3] Basak Alper, Benjamin Bach, Nathalie Henry Riche, Tobias Isenberg, and Jean-Daniel Fekete. Weighted graph comparison techniques for brain connectivity analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 483–492. ACM, 2013.

[4] Katy Börner, Adam Maltese, Russell Nelson Balliet, and Joe Heimlich. Investigating aspects of data visualization literacy using 20 information visualizations and 273 science museum visitors. *Information Visualization*, page 1473871615594652, 2015.

[5] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[6] A.J. Cardno, P.S. Ingham, A. Lewin, and A.K. Singh. Data visualization methods, July 10 2014. US Patent App. 14/200,903.

[7] Tuan N Dang, Paul Murray, and Angus G Forbes. Pathwaymatrix: visualizing binary relationships between proteins in biological pathways. In *BMC proceedings*, volume 9, page S3. BioMed Central Ltd, 2015.

[8] Kasper Dinkla, Michel A Westenberg, and Jarke J van Wijk. Compressed adjacency matrices: untangling gene regulatory networks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2457–2466, 2012.

[9] Cody Dunne, Steven I Ross, Ben Shneiderman, and Mauro Martino. Readability metric feedback for aiding node-link visualization designers. *IBM Journal of Research and Development*, 59(2/3):14–1, 2015.

[10] Tim Dwyer, Nathalie Henry Riche, Kim Marriott, and Christopher Mears. Edge compression techniques for visualization of dense directed graphs. *IEEE transactions on visualization and computer graphics*, 19(12):2596–2605, 2013.

[11] Ronak Etemadpour, Robson Motta, Jose Gustavo de Souza Paiva, Rosane Minghim, Maria Cristina Ferreira de Oliveira, and Lars Linsen. Perception-based evaluation of projection methods for multidimensional data visualization. *IEEE transactions on visualization and computer graphics*, 21(1):81–94, 2015.

[12] Jean-Daniel Fekete, Jarke J Van Wijk, John T Stasko, and Chris North. The value of information visualization. In *Information visualization*, pages 1–18. Springer, 2008.

[13] Emden R Gansner, Yehuda Koren, and Stephen C North. Topological fisheye views for visualizing large graphs. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):457–468, 2005.

[14] Mohammad Ghoniem, J-D Fekete, and Philippe Castagliola. A comparison of the readability of graphs using node-link and matrix-based representations. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on*, pages 17–24. Ieee, 2004.

[15] Alyssa A Goodman. Principles of high-dimensional data visualization in astronomy. *Astronomische Nachrichten*, 333(5-6):505–514, 2012.

[16] Evgeniy Yur'evich Gorodov and Vasiliy Vasil'evich Gubarev. Analytical review of data visualization methods in application to big data. *Journal of Electrical and Computer Engineering*, 2013:22, 2013.

[17] Lars Grammel, Melanie Tory, and Margaret-Anne Storey. How information visualization novices construct visualizations. *IEEE transactions on visualization and computer graphics*, 16(6):943–952, 2010.

[18] Sander Greenland. Quantifying biases in causal models: classical confounding vs collider-stratification bias. *Epidemiology*, 14(3):300–306, 2003.

[19] Sander Greenland, Judea Pearl, and James M Robins. Causal diagrams for epidemiologic research. *Epidemiology*, pages 37–48, 1999.

[20] Taylor Hanken, Sam Young, Karen Smilowitz, George Chiampas, and David Waskowski. Developing a data visualization system for the bank of america chicago marathon (chicago, illinois usa). *Prehospital and Disaster Medicine*, pages 1–6, 2016.

[21] Chun-Cheng Hsu. Evaluation criteria for blog design and analysis of causal relationships using factor analysis and dematel. *Expert Systems with Applications*, 39(1):187–193, 2012.

[22] Imre Janszky, Anders Ahlbom, and Anna C Svensson. The janus face of statistical adjustment: confounders versus colliders. *European journal of epidemiology*, 25(6):361–363, 2010.

[23] René Keller, Claudia M Eckert, and P John Clarkson. Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? *Information Visualization*, 5(1):62–76, 2006.

[24] John King, Kathy Sonderer, and Kevin Lynch. Cognitive benefits of a simple visual metrics architecture. In *International Conference on HCI in Business, Government and Organizations*, pages 319–329. Springer, 2016.

[25] Xiaotong Liu and Han-Wei Shen. The effects of representation and juxtaposition on graphical perception of matrix visualization. In *Proceedings of the 33rd annual ACM conference on Human factors in computing systems*, pages 269–278. ACM, 2015.

[26] William JR Longabaugh. Combing the hairball with biofabric: a new approach for visualization of large networks. *BMC bioinformatics*, 13(1):1, 2012.

[27] Rudolf Netzel, Michel Burch, and Daniel Weiskopf. Comparative eye tracking study on node-link visualizations of trajectories. *IEEE transactions on visualization and computer graphics*, 20(12):2221–2230, 2014.

[28] Christina Niederer, Wolfgang Aigner, and Alexander Rind. Survey on visualizing dynamic, weighted, and directed graphs in the context of data-driven journalism. *Proceedings of the International Summer School on Visual Computing*, pages 49–58, 2015.

[29] Judea Pearl. An introduction to causal inference. *The international journal of biostatistics*, 6(2), 2010.

[30] Zeqian Sheny and Kwan-Liu Maz. Path visualization for adjacency matrices. In *Proceedings of the 9th Joint Eurographics/IEEE VGTC conference on Visualization*, pages 83–90. Eurographics Association, 2007.

[31] Ben Steichen, Giuseppe Carenini, and Cristina Conati. User-adaptive information visualization: Using eye gaze data to infer visualization tasks and user cognitive abilities. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, IUI '13, pages 317–328, New York, NY, USA, 2013. ACM.

[32] Marit M Suttorp, Bob Siegerink, Kitty J Jager, Carmine Zoccali, and Friedo W Dekker. Graphical presentation of confounding in directed acyclic graphs. *Nephrology Dialysis Transplantation*, 30(9):1418–1423, 2015.

[33] Balaji T Thattai, Karthikeyan Shanmugam, Chi Yao, and Yee Lang Lau. Systems and methods for generating data visualization applications, February 24 2016. US Patent App. 15/052,449.

[34] Stefan Tilkov and Steve Vinoski. Node. js: Using javascript to build high-performance network programs. *IEEE Internet Computing*, 14(6):80–83, 2010.

[35] Colin Ware, Eric Neufeld, and Lyn Bartram. Visualizing causal relations. In *Proceedings of IEEE Information Visualization*, volume 99, 1999.

[36] Kuikui Liuös Xiujin Shiás and Yue Liës. Integrated Architecture for Web Application Development Based on Spring Framework and Activiti Engine. *The International Conference on E-Technologies and Business on the Web (EBW2013)*, pages 52–56, 2013.

[37] Hao Zhang, Maoyuan Sun, Danfeng (Daphne) Yao, and Chris North. Visualizing traffic causality for analyzing network anomalies. In *Proceedings of the 2015 ACM International Workshop on International Workshop on Security and Privacy Analytics*, IWSPA '15, pages 37–42, New York, NY, USA, 2015. ACM.