Università
della
Svizzera
italiana

**Software
Institute**

SPATIO-TEMPORAL VISUALIZATION OF EVOLVING COMPANY NETWORKS

Uncovering Insights From The Registry Of Commerce

Francesco Bresciani

January 2025

Supervised by
Dr. Marco D'Ambros, Prof. Dr. Michele Lanza

Co-Supervised by
Prof. Raphaël Parchet, Dr. Andrea Mocci

MASTER'S THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN SOFTWARE AND DATA ENGINEERING

Abstract

Switzerland has experienced significant economic changes over the past 150 years, transitioning from a primarily local economy, where businesses operated mostly in the primary sector, to one of the world's most competitive economies, largely driven by the services sector. This evolution has attracted the attention of economists who aim to understand the factors behind this shift.

Part of the data needed for economists to analyze the Swiss economy is available due to the Swiss Ordinance on the Registry of Commerce, which requires businesses listed in the Central Business Name Index to provide detailed information about their location, ownership, business purpose, and other relevant aspects. The Swiss Confederation ensures this information is made publicly accessible by publishing daily updates in the Official Gazette of Commerce.

Despite the wealth of this valuable data, its potential remains largely unexploited due to its unstructured nature, compounded by the multidimensionality of the information. Therefore, to date, no large-scale analysis has ever been undertaken.

The raw data extracted from the Official Gazette of Commerce is composed by a set of text snippets, each containing information about a specific event related to a company. By combining these events together, it is possible to reconstruct the life-cycle of a company. Some of these events mention other companies, e.g., when a company acquires another company. By combining these events together, it is possible to reconstruct the networks of companies and their evolution over time. Performing this task is challenging because the data is unstructured and the relationships as well as many other attributes of the single companies change over time making traditional data analysis tools unsuitable to explore and analyse this data.

In this thesis we aim at enabling large-scale spatio-temporal analysis of the data of these evolving company networks. The approach we propose seeks to empower researchers with the means to delve into this wealth of information and gain insights beyond basic details like location, capital, and ownership. We aim to uncover how networks of companies have formed, evolved, and interacted. To address these goals, we harnessed visualization techniques capable of breaking down the complexity of the data at hand, offering researchers the means to perform exploratory and explanatory analyses. Our approach paves the way for addressing previously unanswered questions such as the impact of the series of tax reforms the canton of Luzern embarked on in 2010 with the aim of attracting businesses from other cantons—especially Zug, which at the time had a significantly lower tax rate than any other canton in Switzerland.

We implemented a tool to validate the approach we propose in this thesis and demonstrate its effectiveness. The initial phases of this thesis involved trial-and-error iterations that led us to a final design that leverages a set of numerous custom visualisation instead of a single one-size-fits-all visualisation. This is because the key to understanding the data lies in the ability to interact with the data in order to explore it from different perspectives and test different hypotheses. The tool we developed allows users to create custom populations of companies, apply filters to the dataset, and compare the characteristics of companies across different dimensions, such as canton, tax rate, or type. The tool also supports spatio-temporal analysis, enabling users to track the evolution of companies over time.

Acknowledgements

I am deeply grateful for the opportunity I had to work on this thesis with such an expert and supportive team of advisors. It had the privilege to not only work on this thesis but also to pursue my master's degree while working at the Software Institute, within the CodeLounge group.

Words cannot fully express my gratitude to Dr. Marco d'Ambrosand Prof. Michele Lanza, for their invaluable guidance, unwavering support, and belief in me, especially during the early stages of my journey. I would also like to extend my heartfelt thanks to Prof. Raphaël Parchet and Dr. Andrea Mocci for their insightful feedback and constructive suggestions. This thesis would not have been the same without the individual contributions of each of them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

1.1.1 Domain

The Swiss commercial registry serves as a centralized record for information on legal entities engaged in commercial activities across Switzerland, providing transparency and organization to the business environment. Managed federally but administered by the individual cantons, the registry gathers essential information on all companies conducting business within Switzerland. This information includes each company's corporate name, founding year, location, business purpose, key management and board members, authorized signatories, capital structure, and auditing body, if applicable. Originally published in the Swiss Official Gazette of Commerce in 1883, this registry has been available online since 2001 through the Central Business Name Index (Zefix)[1], which provides digital access to registered information from the various cantonal commercial registries.

The registry also plays a crucial role in outlining business relationships, shedding light on networks that connect different companies. Control and ownership relationships are essential in understanding Switzerland's interconnected business landscape. A control relationship is formed when one company acts as a head company to others (known as subsidiaries), while an ownership relationship is present when one company holds stock in another. Together, these types of connections form expansive business networks that reveal the structural relationships within Swiss business, encompassing both domestic and international linkages.

The dynamics within these networks are constantly evolving, driven by various corporate events and strategic business decisions. For instance, takeovers or mergers can shift the landscape significantly, particularly when large head companies acquire competitors or relocate their headquarters. Such changes can alter the existing network structure, impacting competition, market share, and the overall economic balance within Switzerland. These dynamics make Swiss business networks fluid and responsive to both internal and external influences, with new relationships continuously shaping the business environment.

Tax competition is a powerful factor influencing the formation and relocation of business networks in Switzerland. As a federal state with 26 cantons and roughly 2000 municipalities, Switzerland's political structure allows for a degree of fiscal autonomy at the cantonal and municipal levels. This autonomy includes the ability to set corporate tax rates, creating an environment where cantons compete to attract companies by offering favorable tax conditions. Certain cantons have established themselves as tax heavens, enticing companies from higher-tax areas within Switzerland and abroad. This competition encourages companies to make strategic decisions about where to locate their headquarters, subsidiaries, and other operational branches to benefit from lower tax burdens, thereby impacting the configuration and location of business networks across the country.

[1] https://www.zefix.ch/en/search/entity/welcome

From an economic research perspective, analyzing the historical and current organization of these business networks provides valuable insights into the impact of tax policy and competition on corporate structure and behavior. Key questions arise, such as how inter-cantonal tax competition affects the organization and geographical distribution of companies and their networks. For example, researchers are interested in determining under what conditions companies expand into multiple cantons, whether tax policy shifts have historically influenced corporate relocations, and if tax competition yields a zero-sum effect, where one canton's gain is another's loss, or a positive-sum outcome, where the entire economy benefits. Answering these questions through large-scale, historical data analysis of business networks offers a deeper understanding of the relationship between tax policy, corporate strategy, and economic development in Switzerland.

1.1.2 Problem

In Zefix, the data is presented on a firm-by-firm basis, making it impossible to perform any analysis that requires the aggregation of data of multiple companies. This presentation also complicates understanding the evolution of a single company over time. While historical and large-scale analysis of these networks is economically interesting, it is not feasible with the Zefix web platform. Even with the data that CodeLounge[2] has extracted and structured in a relational database in the context of a project part of the NRP77 program[3] funded by the Swiss National Science Foundation, querying remains challenging due to the complexity, multi-dimensionality, and time-variant attributes of the data. Additionally, the data is not easily accessible to researchers as it requires knowledge of SQL to query it.

1.1.3 Thesis Contributions

In this thesis, we provide researchers with the means to effectively explore and analyze data on companies and their subsidiaries, including their relationships and changes over time, in an intuitive manner. Specifically, we devise a visual analytics approach to support the spatio-temporal analysis of evolving business networks formed by these head companies and their subsidiaries. By leveraging visual analytics techniques, we aim to gain insights into how these networks operate within the Swiss economic landscape, thereby breaking down the complexity of the task.

To achieve this, we integrate various data sources to create a comprehensive dataset that reflects the dynamic nature of business networks. We also employ data processing techniques to clean, merge, and transform the data into a format suitable for analysis.

Furthermore, we address the challenges associated with the visualization of large and complex datasets by implementing interactive visualizations that allow users to drill down into specific aspects of the data. These visualizations are designed to be user-friendly and customizable, enabling researchers to tailor the analysis to their specific needs and preferences. The combination of these techniques provides a powerful toolset for understanding the intricate dynamics of business networks and their impact on the economy.

The tool we developed is packaged as a Docker image that is built using a CI/CD pipeline we define for the purpose. The Docker image can be then be executed on every machine where Docker is installed and where a copy of the original database is available. Our project contains all the processing logic and instructions to recreate the schemas and tables as well to process and fill the data into the newly created tables. This is useful in case the original database would be extended, for example with more recent data. We made available an instance of the web application and an instance of the database at `https://hakken.si.usi.ch`. We decided to shut down this instance for privacy concerns, but CodeLounge has access to the source code, the instance, the instructions, and the virtual machine with the docker compose file to run it.

The project is composed of more than 15k lines of code, comments excluded, of which more than 14k are TypeScript or JavaScript code.

[2] https://codelounge.si.usi.ch/

[3] https://www.snf.ch/en/hRMuYd5Qqjpl1goQ/page/researchinFocus/nrp/nrp77

The contributions of this thesis are:

- we devise an approach to explore, filter, and select the data to be analyzed, as well as several visualizations to represent the aspects of the data that are of interest for economic research. The conceptual approach is described in Section 4.1
- we provide an effective way to model the data of evolving entities by applying HiSMo [1] to graph data. HiSMo is a model that captures the historical evolution of entities over time. The model is described in Section 3.3
- we provide a visual analytics tool to exploit the approach. The tool offers a variety of functionalities that are described in Section 4.2. In Chapter 5 we show the capabilities of the tool by presenting a series of use cases
- we solve the problem of cross-filtering [4] entities with time-dependent attributes its needed to show a single graphical element corresponding to the entities matching the conditions, but the filtering is done considering attributes of the versions that make up the history of the entities to select. The challenges are not rooted in selecting the entities, rather, in adapting all the other visualisations. This is a bigger challenge than may seem at first and we will see why in Section 4.1

1.2 Solution

1.2.1 Conceptual Approach

At a foundational level, we developed a data model to represent the data and a relational model to store the data efficiently. On top of this, we developed a visualization model to enable users to:

- perform explorative analysis of evolving company networks
- perform large-scale spatio-temporal explanatory analysis of sub-populations of companies and their interactions

The first task is complex, as the data is multidimensional and time-dependent. To the best of our knowledge no solution has been proposed to date to perform filtering, aggregation, and visualization of graph data with both properties of nodes changing over time and edges being active only during certain periods. The second task is challenging because we aim to revisit the way analysis is performed. Economists usually perform analysis using statistical methods, which is not the most convenient and effective way to perform exploratory analysis on this data.

For the users of our tool, we provide a main graph visualization they can use to explore the data by filtering and selecting nodes matching their criteria within the time-frame of interest. These nodes can be selected and stored as custom-defined populations. These populations are then used to create visualizations that allow users to:

- compare the evolution of the populations over time
- compare the evolution of the number of active relations between the populations over time
- gain insights into how companies moved from one population to another

This way, users can easily see how the companies in the populations have evolved over time and how they have interacted with each other.

[4] Cross-filtering is the process of filtering data based on multiple criteria e.g., the user filters only companies with *type: head* and *canton: TI or ZH* or based on the selection on another visualisation e.g., adjusting barcharts showing the distribution across the properties of the selected nodes to the current selection. Our contribution solves the issue with respect to the first meaning of the term.

1.2.2 Technical Approach

Our data model is inspired by the HiSMo framework [1], which provides a structured approach to modeling time-evolving entities. This model is implemented in TypeScript. The master data, along with the results of our data processing, is stored in a PostgreSQL database. PostgreSQL was chosen for its robustness, scalability, and ability to handle complex queries efficiently.

The visualization model is also implemented in TypeScript and uses D3.js, a powerful library for creating dynamic and interactive data visualizations. With D3.js, we can build rich visual representations that help users explore and analyze the data effectively. The visualization model is integrated into a web application, making it accessible through any modern web browser.

This web application is developed using React.js and Next.js, a combination that allows us to create highly interactive and performant user interfaces. React.js provides a component-based architecture, while Next.js adds server-side rendering and other optimizations for faster load times and improved user experience.

To ensure seamless communication between the web application and the PostgreSQL database, we use Prisma ORM. Prisma simplifies database access by providing a type-safe and intuitive way to interact with the data.

1.3 Document Structure

In Chapter 2 we present the state of the art in the field of the thesis. More specifically we provide an historical perspective on the evolution of the data visualisation techniques in Section 2.1, an overview of static networks visualisation and time-varying networks visualisation in Sections 2.2 and 2.3, respectively. We later provide and overview of existing visual analytics in Section 2.4 and more specifically of networks visual analytics tools in Section 2.5. We conclude the state of the art chapter by describing some web-based visualisation libraries in Section 2.6, where we focus primarily on d3-js, the library at the core of the visualisations and interactions implemented in our tool.

In Chapter 3 we describe the context of the thesis. We start by introducing the domain of the thesis in Section 3.1, where we explain what the Commercial Registry is, what we mean by business network, the dynamics of business networks we consider in this thesis, what tax competition is, and finally why economic researchers are interested in all these topics. In Section 3.2 we describe the dataset we use in this thesis, and more in particular what data we received, how we processed it and what data we used for the visualisation and analysis in our tool. As mentioned multiple times in this thesis, the data at hand is quite complex and required a deep understanding of the domain to be able to process and visualise it correctly. For this reason we needed to devise a data model, which we describe in Section 3.3. In Section 3.4 we describe in a structured way the problem we aim to solve.

After outlining the context, domain, dataset, and problem, in Chapter 4 we delve into the approach adopted to address the problem. In Section 4.1 we present the conceptual framework, detailing the tasks we aim to support and the guiding principles behind our approach. We also explain the processing pipeline and the architecture of the tool. In Section 4.2, we focus on the tool's implementation, discussing its architecture, the components developed, and the interactions designed to effectively support the identified tasks.

In Chapter 5 we present the evaluation of the tool. We start by describing the evaluation process, where we explain the iterative development and evaluation approach adopted, the role of the domain expert, and the feedback gathered during the evaluation. We then present the basic use cases in Section 5.1, which demonstrate the tool's core functionality. In Section 5.2 we present an advanced use case that showcases the tool's ability to perform more complex analyses.

Finally, in Chapter 6 we present the conclusions of the thesis, summarising the contributions, discussing the limitations of the work, and outlining potential directions for future research.

Chapter 2

State of the Art

2.1 Data Visualisation

Data visualisation and information visualisation are two distinct sciences often used in combination in order to make more accessible data and the information it carries. Data visualisation focuses on the graphical representation of raw data, while information visualisation aims to convey specific insights or messages derived from the data. For the purposes of this section we do not need to distinguish between the two, therefore, we will use the term data visualisation to refer to both.

Graphic visualisation is an excellent approach for exploring data and an essential tool for presenting results [2]. Data visualisation techniques are mainy used to tackle two different analytical tasks: exploratory and explanatory data analysis. Exploratory data analysis refers to a process through which knowledge is iteratively extracted from data to answer questions as they arise. Explanatory data analysis on the other hand is used to communicate the results of an analysis in a clear and concise way. Explanatory visualizations are usually used to communicate findings or inspire action [3].

The roots of data visualisation date back to the first map-like drawings (at least 40 thousands years ago [4]) created to aid in navigation and exploration or to map the position of the stars. Another expression of the most ancient visualisations are cave depictions (at least 64 thousands years ago [5]).

In the 16-17th century data visualisation science rapidly evolved taking advantage of the advancements in the theory and practical applications proposed by scientists of the relevance of Descartes, Fermat, Galileo, Graunt and Petty in the fields of analytic geometry, coordinate systems, probability, and statistics. [6].

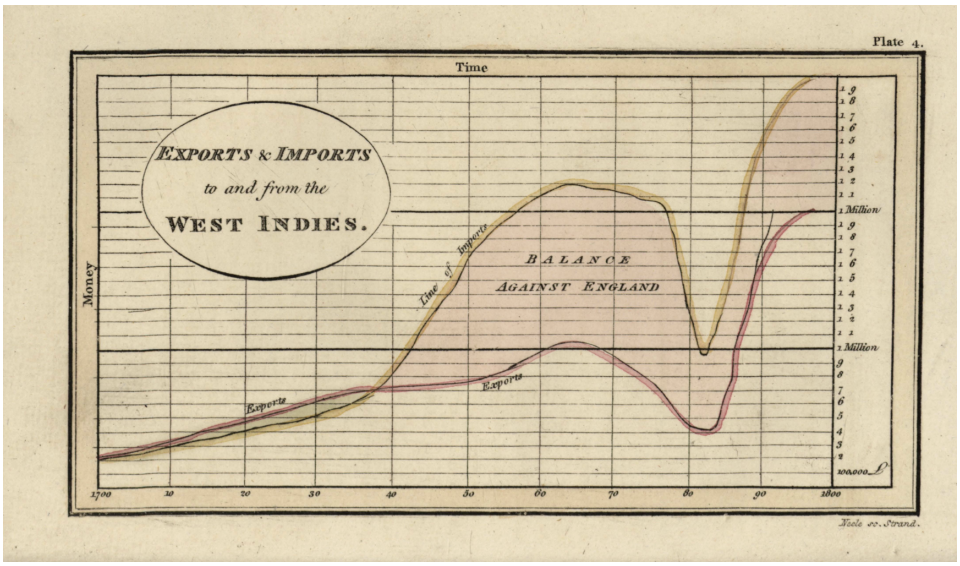

FIGURE 2.1: Playfair's line chart

Thanks to the rise of statistical thinking and widespread data collection, conducted mainly to serve planning and commerce purposes, in the 18th century have been created the first statistical graphs [6].

The inventor of these charts is William Playfair[1758-1823] and they are still extensively used nowadays. To name a few of Playfair's contribution, which have been introduced in two of his greatest books, namely the Commercial and Political Atlas [7] and the Statistical Breviary [8], we can mention the line chart in Figure 2.1, bar chart, area chart, pie chart 2.2, and time-series plot. Playfair's work is described by Spence and Wainer in A Brief History of Data Visualization [9].

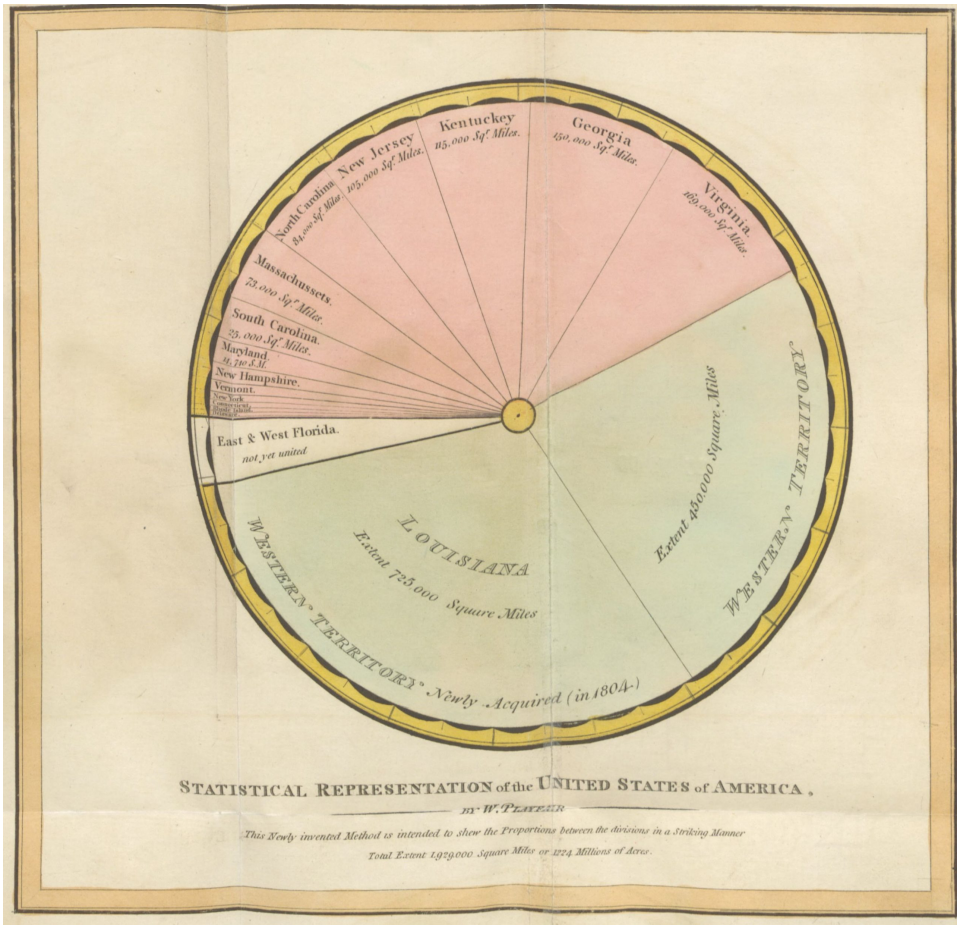

FIGURE 2.2: Playfair's pie chart

The following years are characterised by the evolution in the field of map drawings led by Minard[1781-1870] and Nightingale[1820-1910], and by the contribution of Galton[1822-1911] and Pearson[1857-1936] in the refinement of the statistical graphs introduced by Playfair.

This overview of the major events and contributors of the early days of data visualisation sets the context in which more modern approaches to data visualisation have been developed. Of particular relevance is the work of Edward Tufte[1942-] whose main contributions are in the field of information design and visual literacy. The minimalistic approach of Tufte's work is summarised in his books Envisioning information [10] and The Visual Display of Quantitative Information [11]. Tufte is the proponent of the concepts of:

- data-ink ratio, which is the proportion of ink devoted to the non-redundant display of data-information in a graphic,
- data density, which is the amount of data-information displayed per unit of space, and

- the lie factor, a value to describe the relation between the size of effect shown in a graphic and the size of effect shown in the data.

Tufte's work is inspired by the theoretical principles of visual perception introduced by gestalt psycologists Wertheimer[1880-1943], Kohler[1887-1967], and Koffka[1886-1941] [12] [13] [14]. The gestalt principles are a set of laws describing how humans perceive and group objects. Of our particular interest is the Prägnanz law, Prägnanz is a German word that emphasize the concepts of conciseness and orderliness.

The gestalt principles are

- Law of proximity,
- Law of similarity,
- Law of closure,
- Law of symmetry,
- Law of common fate,
- Law of continuity, and
- Law of past experience.

More recently Few has been one of the most influential authors in the field of data visualisation. In his books Show Me the Numbers [15], Information Dashboard Design [16], and Now You See It [17] Few provides a modern approach to data visualisation inspired by Tufte's work. Few's work is mainly focused on the design of visualisations applied to the business context but the principles he proposes are applicable to any domain.

In the next sections we provide a more detailed description of each principle extracted from What is the origin of the gestalt principles [18].

2.1.1 Law of proximity

The law of proximity suggests that when individuals observe a collection of objects, they tend to group together objects that are near each other. For instance, in the figure demonstrating the law of proximity, although there are 72 circles, we perceive them as grouped. Specifically, we see a cluster of 36 circles on the left side and three clusters of 12 circles each on the right side. This principle is frequently utilized in visual analytics dashboards to highlight which data points are related.

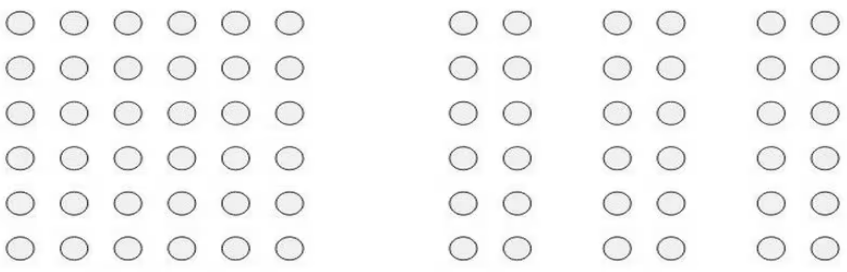

FIGURE 2.3: Law of proximity

2.1.2 Law of similarity

The law of similarity states that objects within a collection are perceived as a group if they share similar attributes. These attributes can include shape, color, shading, or other characteristics. For instance, in the figure demonstrating the law of similarity, there are 36 circles arranged in a square, all equidistant from each other. Among these, 18 circles are shaded dark and 18 are shaded light. We tend to perceive the dark circles as one group and the light circles as another, forming six horizontal lines within the square. This grouping effect is a result of the law of similarity.

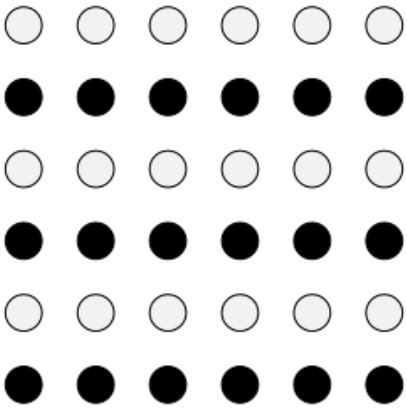

FIGURE 2.4: Law of similarity

2.1.3 Law of closure

Gestalt psychologists posited that humans have a tendency to perceive incomplete objects as complete. For instance, even if a circle is not fully drawn, we still recognize it as a circle. This phenomenon is known as the law of closure. According to this principle, when parts of a visual are missing, our perception fills in the gaps to form a complete image. This tendency helps to enhance the regularity of the stimuli we observe. For example, in the figure illustrating the law of closure, we perceive a circle on the left and a rectangle on the right, despite the gaps in their outlines. Without this perceptual tendency, we would see a collection of disjointed lines rather than cohesive shapes.

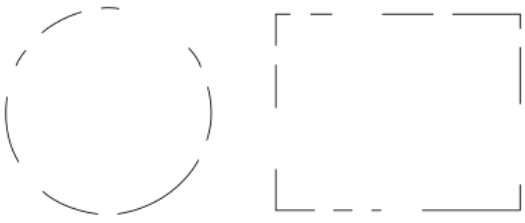

FIGURE 2.5: Law of closure

2.1.4 Law of symmetry

The law of symmetry posits that our minds naturally perceive objects as symmetrical and organized around a central point. This principle suggests that it is visually appealing to divide objects into symmetrical parts. Consequently, when two symmetrical elements are not connected, our perception tends to link them together to form a unified shape. The resemblance between symmetrical objects enhances the likelihood of them being grouped into a single symmetrical entity. For instance, in the figure illustrating the law of symmetry, we see a configuration of square and curled brackets. Our perception tends to group these into three pairs of symmetrical brackets rather than viewing them as six separate brackets.

FIGURE 2.6: Law of symmetry

2.1.5 Law of common fate

The law of common fate suggests that we perceive objects as following the smoothest path of motion. Studies in visual perception have shown that when elements of an object move, we tend to see them as part of a continuous path. This principle indicates that we group together elements that share a common motion trend, perceiving them as part of the same trajectory. For instance, if we observe an array of dots where half are moving upward and the other half downward, we will perceive two distinct groups based on their direction of movement.

2.1.6 Law of continuity

The law of continuity, also referred to as the law of good continuation, posits that elements within objects are perceived as part of a cohesive whole when they are aligned. When objects intersect, people tend to see them as continuous, uninterrupted entities. This principle suggests that stimuli are perceived as distinct even when they overlap. Elements with sharp, abrupt directional changes are less likely to be grouped as a single object. For instance, in Figure 2.7, we see two crossed keys. Our perception tends to view the key in the background as a single, continuous key rather than two separate halves.

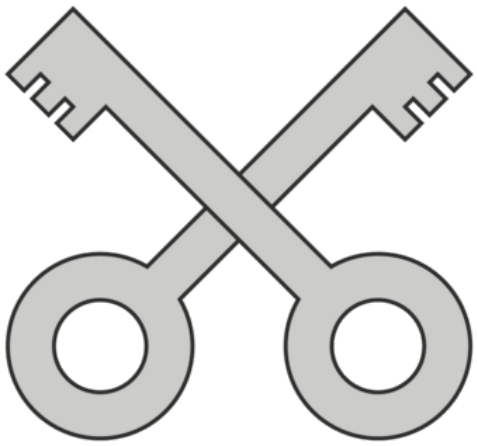

FIGURE 2.7: Law of continuity

2.1.7 Law of past experience

The law of past experience suggests that visual stimuli are often interpreted based on previous encounters. When objects are frequently seen together or within short time intervals, they are more likely to be perceived as a group. For instance, in the English language, 26 letters are combined to form words according to specific rules. When someone encounters an unfamiliar English word, they rely on the law of past experience to distinguish the letters "L" and "I" as separate entities, rather than merging them into a single shape like an uppercase U, which would be an application of the law of closure.

2.2 Static Networks Visualisation

Networks science is backed by graph theory, a branch of mathematics that studies the properties of graphs. In graphs theory a network is modeled as a set of nodes and edges[1]. A network is modeled as a pair $N = (V, E)$ where V is the set of nodes, also called vertices, and E is the set of edges. An edge is a pair of nodes (u, v) where $u, v \in V$.

The simplest categorisation of graphs can be done by distinguishing between weighted, unweighted, directed, and undirected graphs. A graph is weighted if each edge has a weight associated with it and directed if the edges have a direction.

Graphs can also be categorised by their level of connectedness. A graph is connected if there is a path between every pair of nodes, disconnected if it is not connected – i.e., if there exist at least two nodes in the graph such that there is no path connecting them. A graph is strongly connected if there is a path between every pair of nodes in both directions; instead, a graph is weakly connected if there is a path between every pair of nodes in at least one direction.

Furthermore, graphs can also be categorised by their level of density. A graph is dense if the number of edges is close to the maximal number of edges. A graph is sparse if the number of edges is close to the minimal number of edges.

Finally, graphs can be categorised by the presence or absence of cycles, where we can distinguish between cyclic and acyclic graphs.

A special type of graph, which is particularly relevant in the context of this thesis, are bipartite graphs, also known as bigraphs. A bipartite graph is a graph whose nodes can be divided into two disjoint sets U and V such that every edge connects a node in U to one in V. Bipartite graphs are used to model relationships between two different classes of objects e.g., to model the relationship between controlling companies and controlled companies where each controlled company has exactly one controlling company and controlling companies are not controlled by other firms.

Depending on their structure and properties, networks with a static structure are usually visualised as:

- node-link diagram,
- tree,
- chord diagram,
- arc diagram,
- sankey diagram,
- edge bundling, or
- matrix.

[1] Edges are links between the nodes of the network

2.2.1 Node-link diagram

Node-link diagrams are the most common way to visualise networks. In a node-link diagram, nodes are represented as points and edges as lines connecting the nodes. The position of the nodes and the length of the edges can be used to encode additional information about the network. Node-link diagrams are particularly useful to visualise small networks, but they become cluttered and hard to read when the number of nodes and edges increases.

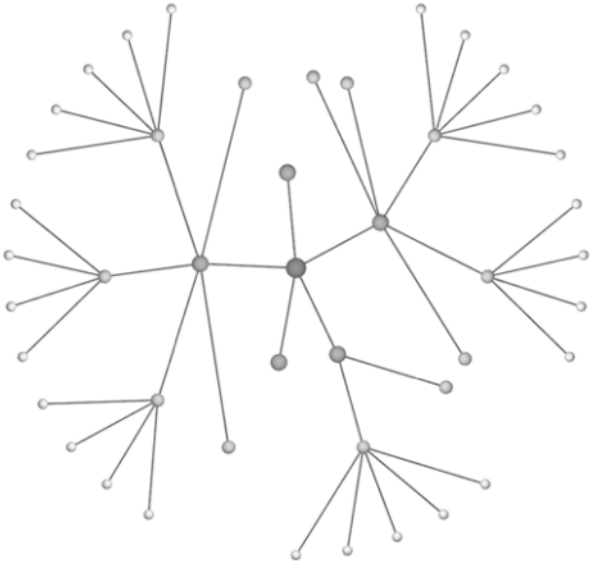

FIGURE 2.8: Node-link diagram
Image from [19]

2.2.2 Tree

Tree diagrams are a type of node-link diagram where the nodes are organised in a hierarchical structure. Trees are used to represent relationships between nodes in a hierarchical way. Trees are particularly useful to visualise the relationships between nodes in a hierarchical structure, but they are not suitable to represent more complex relationships between nodes.

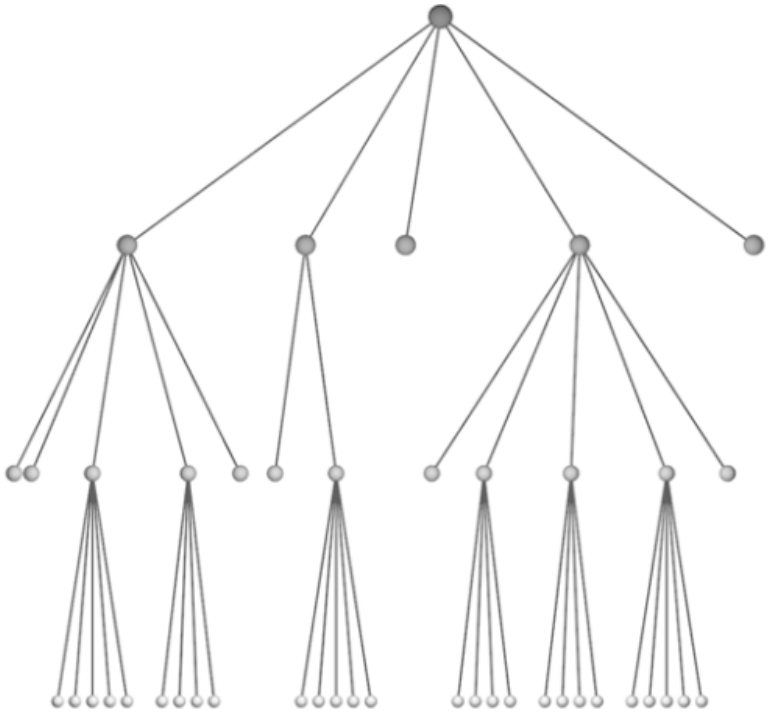

FIGURE 2.9: Tree diagram
Image from [19]

2.2.3 Chord diagram

Chord diagrams are a type of node-link diagram where the nodes are arranged in a circle and the edges are drawn as arcs connecting the nodes. Each node is represented as a segment on the circumference of the circle, and each edge is represented as a curved line (chord) connecting two nodes. Chord diagrams are particularly useful for visualising the relationships between different groups or categories, as the circular layout allows for easy comparison of connections between nodes. However, they can become cluttered and difficult to interpret when the number of nodes and edges is large.

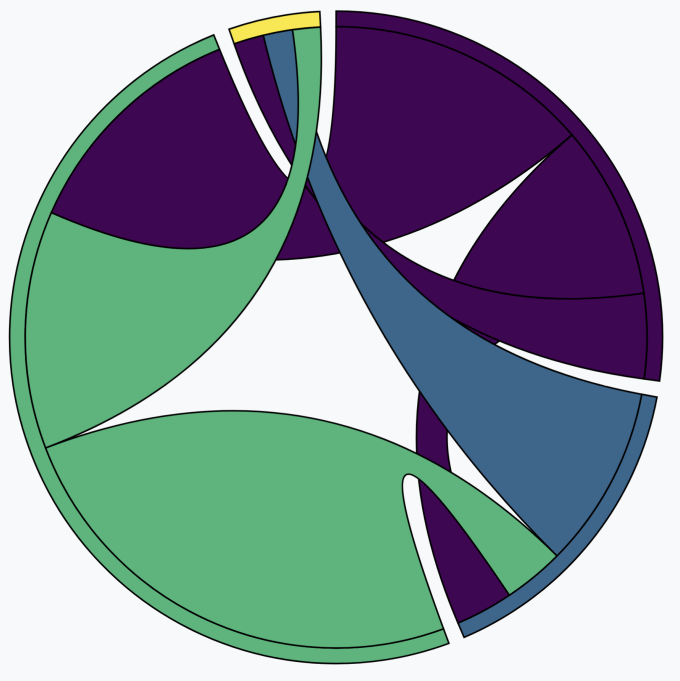

FIGURE 2.10: Chord diagram
Image from https://d3-graph-gallery.com/graph/chord_colors.html

2.2.4 Arc diagram

Arc diagrams are a type of node-link diagram where the nodes are arranged in a line and the edges are drawn as arcs connecting the nodes. Each node is represented as a point on the line, and each edge is represented as a curved line (arc) connecting two nodes. Arc diagrams are particularly useful for visualising the relationships between nodes in a linear structure, as the linear layout allows for easy comparison of connections between nodes. However, they can become cluttered and difficult to interpret when the number of nodes and edges is large.

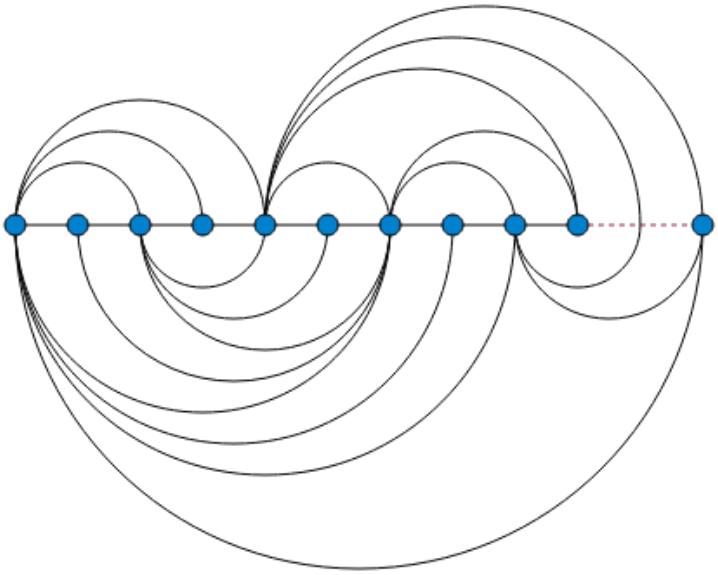

FIGURE 2.11: Arc diagram
Image from https://en.wikipedia.org/wiki/Arc_diagram

2.2.5 Sankey diagram

Sankey diagrams are a type of flow diagram where the nodes are represented as rectangles and the edges are represented as arrows connecting the nodes. Sankey diagrams are particularly useful for visualising the flow of information or resources between different nodes in a network, as the width of the arrows can be used to encode the flow of information or resources. Sankey diagrams are particularly useful for visualising the flow of information or resources between different nodes in a network, as the width of the arrows can be used to encode the flow of information or resources. However, they can become cluttered and difficult to interpret when the number of nodes and edges is large.

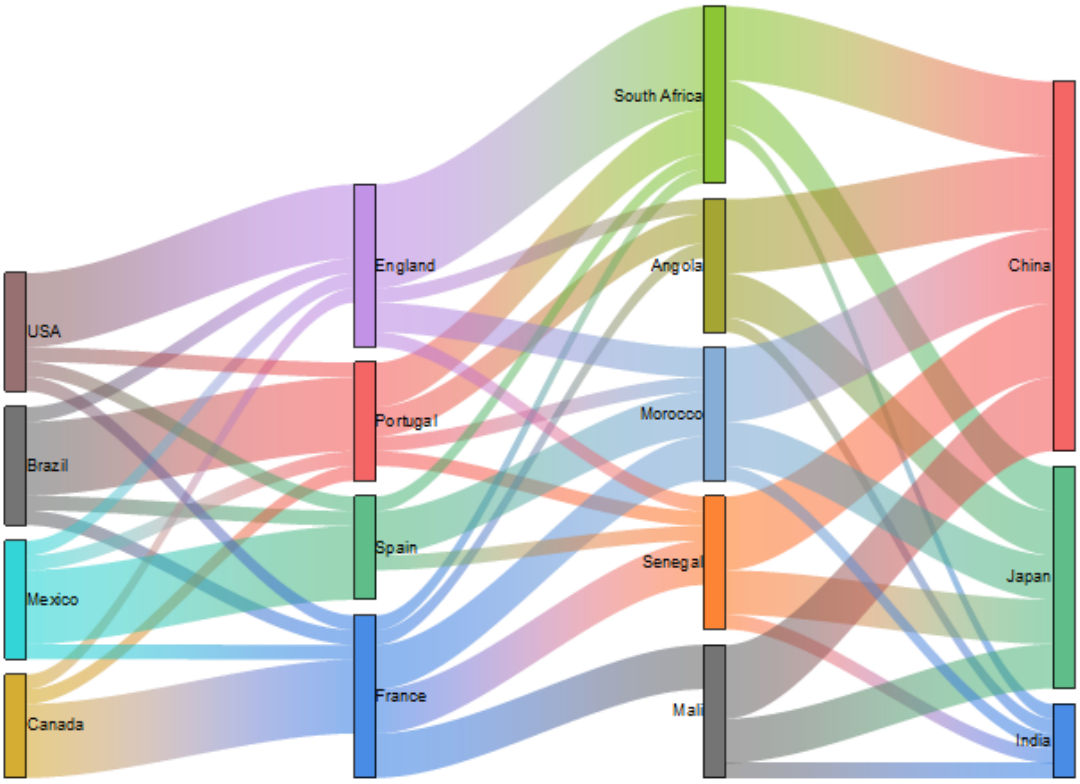

FIGURE 2.12: Sankey diagram
Image from https://www.originlab.com/doc/Origin-Help/Sankey-Diagram

2.2.6 Edge bundling

Edge bundling is a technique used to reduce visual clutter in node-link diagrams by grouping edges together into bundles. In edge bundling, edges that are close together are grouped together into bundles, which are then drawn as curved lines connecting the nodes. Edge bundling is particularly useful for visualising complex networks with a large number of nodes and edges, as it reduces visual clutter and makes it easier to see the overall structure of the network. Often edge bundling is used in combination with other visualisation techniques to improve the readability of the network such as placing the nodes in a circular layout. An example of usage of edge bundling is shown in Figure 2.13.

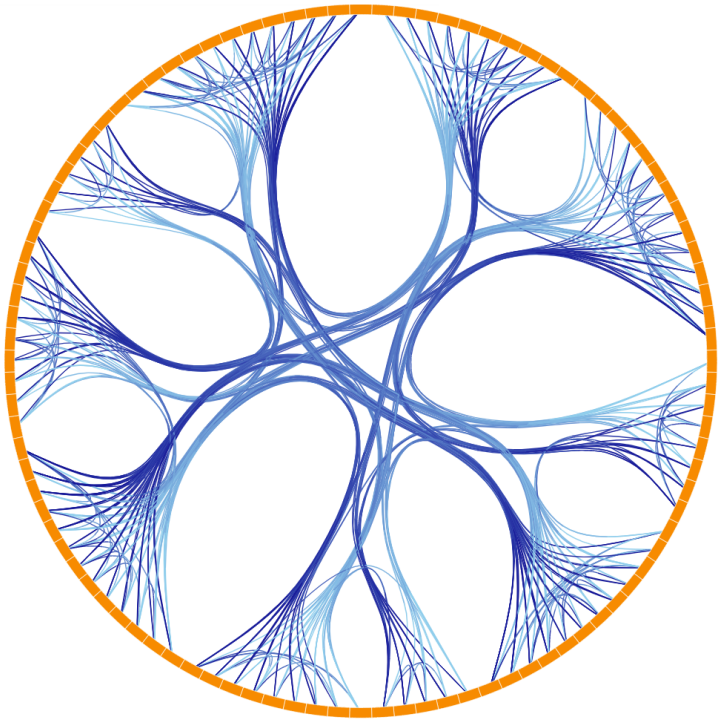

FIGURE 2.13: Example of edge bundling
Image from https://www.yworks.com/pages/interactive-showcase-of-graph-layouts

2.3 Time-Varying Networks Visualisation

Time-evolving networks are networks whose structure changes over time. They can be formalised in the same way as static networks with the addition of the time dimension. A time-evolving network is a sequence of graphs $N = (V, E, T)$ where V is the set of nodes, E is the set of edges, and T is the set of time steps. Each time step $t \in T$ is a graph $G_t = (V_t, E_t)$ where V_t is the set of nodes at time t and E_t is the set of edges at time t.

Visualising time-evolving networks is more challenging [20]; existing approaches to visualise time-evolving networks are mainly based on the use of

- temporal networks [21],
- animations [22], or
- on the use of a sequence of static snapshots [22]

Due to the high number of nodes we will have to deal with in this thesis, we do not consider temporal networks as a viable approach, so we do not discuss them further. In Figure 2.14 we can see an example of a time-evolving network visualised using a sequence of snapshots.

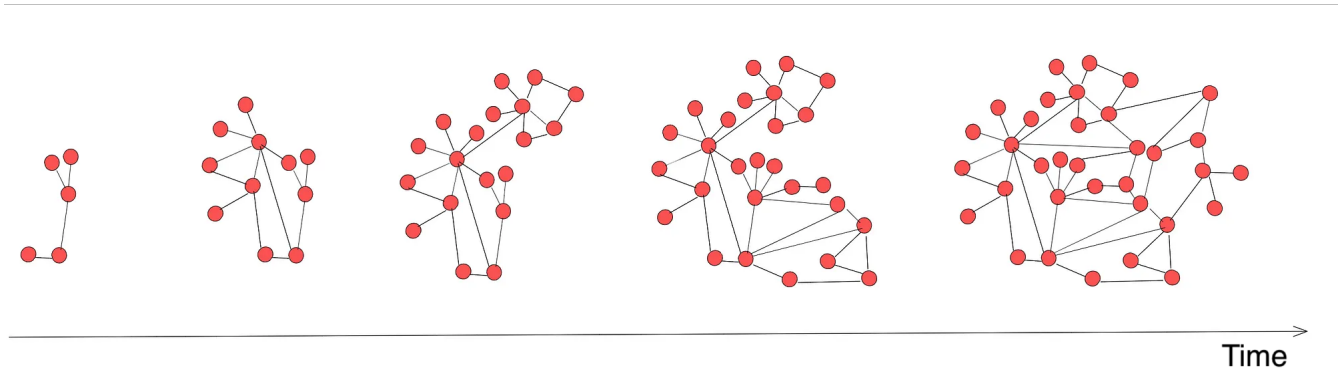

FIGURE 2.14: Example of a time-evolving network visualised using a sequence of snapshots
Image adjusted from https://towardsdatascience.com/temporal-semantic-network-analysis-bd8869c10f10

A way to break down the complexity of visualisasing a time-varying network is by ignoring the time dimension [23], focus on the structure of the network and other aspects of the visualisation. Once the method for visualizing these elements is clear, the time dimension can be incorporated into the visualization using one of the two approaches outlined above.

Both the other two approaches present some limitation i.e., using animations can be difficult to follow the evolution of the network over time [22] and using static snapshots is not always the best approach because it can be difficult to compare the different snapshots [22]. Defining a successful visual analytics approach to allow making sense of complex and time-varying data is difficult and require domain-specific customization [24].

2.4 Visual Analytics

Visual analytics is the science of analytical reasoning facilitated by visual interactive interfaces [25]. This means that employing digital means to support the interaction is a key aspect of visual analytics. Keim et

al. [26] describe visual analytics as "an interdisciplinary field that combines techniques from information visualization, visual data mining, statistics, [...] and human-computer interaction".

Many insights to design clean graphical user interfaces is provided by Schneiderman's Visual Information-Seeking Mantra [27]: overview first, zoom and filter, then details-on-demand. In his work he provides detailed guidelines on how to create visualisation based on the data type at hand and on different tasks the user can be interested in carrying out. The following are the data types and the tasks mentioned above.

Data Types

- 1-dimentional
- 2-dimentional
- 3-dimentional
- temporal
- multi-dimensional
- tree
- network

Tasks

- overview
- zoom
- filter
- details-on-demand
- relate
- history
- extract

2.5 Networks Visual Analytics Tools

Generic visualisation tools such as Tableau, MS Excel, or MS Power BI are not suitable for visualising and analising networks, but there are many tools specifically designed for visualising networks. Some of the most popular are Gephi, Gephisto, VOSviewer, Cytoscape, Kumu, GraphInsight, NODEXL, Orange, Graphia, Graphistry, SocNetV, and Tulip.

The problem with all these tools is that they do not allow to harness the time dimension of the network and they allow to create only basic visualisations and to extract only basic statistics about the data at hand. As mentioned earlier, a successful visual analytics approach require domain-specific customization [24]. The approach we propose in this thesis overcomes these limitations by leveraging the power of web libraries and by providing a customisable visual analytics tool for time-evolving networks. In the next section we provide an overview of the most popular web libraries for visualising networks.

2.6 Web-Based Visualisation Libraries

Some of the most popular web-based libraries that allow creating custom visualisations are Echarts, Vega-lite, Vega, Plotly, and D3.js. Except for Plotly we used all of them in this thesis. We started with Echarts but we found it too limited for our purposes, then we moved to Vega-lite and Vega but we found them too complex for our purposes, so we finally moved to D3.js which we found to be the most flexible and powerful library for creating custom visualisations.

D3.js (Data-Driven Documents) is a JavaScript library for producing interactive data visualizations using web technologies. D3.js is known for its powerful capabilities and flexibility, allowing developers to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven transformations to the document.

D3.js operates at a low level, giving developers fine-grained control over the final visual representation. This means that while it can be more complex to use compared to higher-level libraries like the others mentioned, it offers unparalleled flexibility and power to create custom visualizations tailored to their specific needs.

The library also includes a variety of utilities for working with data, such as functions for scaling, color manipulation, and data parsing. These utilities make it easier to preprocess and transform data before visualizing it.

2.7 Summary

In this chapter, we have reviewed the state of the art in the field of data and information visualisation starting by providing an overview of the driving principles and their origins. We have then focused on the visualisation of networks, starting with the visualisation of static networks and then moving to the visualisation of dynamic networks. In Section 2.3 we explained why the visualisation of networks is a complex task, especially when dealing with large and dynamic networks and why the common visualisation techniques are not always suitable. Finally, we have reviewed some of the most popular tools for the analysis of networks and the most popular web-based visualisation libraries. In the next chapters, we will present the approach we have followed to overcome the limitations of the current visualisation techniques and tools and to provide a solution that is able to analyse large and dynamic networks in an effective way.

Chapter 3

Evolving Business Networks

3.1 Domain

3.1.1 Commercial Registry

The commercial registry is a public registry administered by the Swiss Confederation and maintained by the Cantons. It contains important information on legal entities conducting business activity. One of its purposes is to register and publish legally relevant facts about commercially managed companies such as:

- their corporate name,
- year of establishment,
- location of the head office,
- business purpose,
- names of the partners, of the members of the board of directors, and of the managers,
- authorized signatories,
- capital structure, and
- auditing body, if any [1].

The entries collected from the different cantonal commercial registries are published in the Swiss Official Gazette of Commerce since 1883. Since 2001 they are also available in digital format in a web platform named Central Business Name Index (Zefix).

In the Registry of Commerce each company is identified by an id called EHRAID.

3.1.2 Business Networks

In addition to information about company events, the commercial registry also includes details on relationships between companies. These relationships can be of different types, and the set of companies connected by these relationships form a business network. The most relevant types of relationships are:

- **control:** when a head company has branch companies (the controlled companies are referred to as subsidiaries),
- **ownership:** when a company owns the stocks of another company
- **acquisitions:** when a company acquires another company and possibly all its subsidiaries

[1] https://www.kmu.admin.ch/kmu/en/home/concrete-know-how/setting-up-sme/starting-business/commercial-register%20.html

- **mergers:** when two companies merge into a new company, possibly with all their subsidiaries
- **joint management:** when two or more companies share the same management

In this thesis we focus on control and ownership relationships. Acquisitions and mergers are not considered directly but are inferred from control relationships. Joint management relationships are not considered because at the moment of writing this thesis, the master data provided by CodeLounge does not contain information about people.

3.1.3 Dynamics of Business Networks

The dynamics of business networks resulting from by the ever changing structure of the networks and of the companies part of them are of interest to economists because they can provide insights into the evolution of the Swiss economy. For example, the relocation of a company's headquarters or subsidiary to another canton can have a significant impact on the economic activity of the canton of origin, as well as the canton of destination. This is because the relocation can lead to the creation of new companies, the transfer of employees, and the establishment of new business relationships. The relocation can also have an impact on the tax revenue of the canton of origin, as well as the canton of destination. M&A[2] events can also be interesting as they may cause massive changes in the Swiss economic fabric really quickly, two of the most interesting domestic M&A events are the acquisition of DER Touristik Suisse by Kuoni Immobilien AG and the acquisition of Credit Suisse from UBS.

3.1.4 Tax Competition

Switzerland is a federal state composed of 26 cantons and approximately 2000 municipalities. Each sub-federal jurisdictions has a certain degree of freedom on defining its own tax policy. This leads to a phenomenon known as tax competition. This competition is strategically exploited by cantons, which have the power to set their own corporate tax rates. This has led to a situation where some cantons have become tax heavens, attracting companies from other cantons and even from abroad.

3.1.5 Economic Researchers' Interest

Performing historical and large-scale analysis of these networks is interesting from an economic point of view because it would allow researchers answer questions such as:

- how did a company evolve over time?
- how did a business network evolve over time with respect to its structure and with respect to the location of its head and subsidiary companies?
- how many companies matching specific criteria about their location or tax rate behaved in a specific in a specific point in time, e.g., how many subsidiaries part of a business network with a head company located in a canton with high tax rate moved from the canton of Zurich to the Canton of Ticino in 2019?
- how tax competition between cantons affect the organization of companies e.g., in what circumstances companies form a business network opening subsidiaries located in different cantons?
- if and how changes in cantonal taxation policies affected the relocation of companies?
- does tax competition among sub-federal jurisdictions such as the cantons lead to a zero-sum or a positive-sum game?

[2]Merge & Acquisition

3.2 Dataset

The dataset our tool allows to explore and analyse contains data about ~18'000 swiss companies matching two criteria:

- the company was active between 2002 and 2021 and
- the company has been part of business network.

To obtain the dataset, we reprocessed a dataset of more than 1 million companies from the Swiss commercial register created by CodeLounge in the context of a project conducted in collaboration with the Institute for Economic Research (IRE) at USI, part of the National Research Program NRP77 funded by the Swiss National Science Foundation. The dataset was created by processing pdf files containing the entries of the Swiss commercial register since 2001, when it started to be published in digital form instead of on paper. The entries have been processed and enriched with additional information about the companies, but it is structured in a way that do not allow to easily track or reconstruct the evolution of the companies over time. The CodeLounge dataset provides well structured data about major events regarding the companies such as changes in the company's name, address, legal form, merge and acquisition events, etc.. For each type of event there is a table with the common structure shown in Table 3.1.

start_date	end_date	company_ehraid	new_value

TABLE 3.1: Example of a table in the CodeLounge dataset

We defined a data model, described in the next section, that makes the information contained in the dataset more accessible by representing the evolution of the company as a sequence of company versions. In our dataset there is a single table with all the subsequent versions of the companies, each row representing a version of a company. The table has the structure shown in Table 3.2.

start_date	end_date	company_ehraid	name	canton_id	status

TABLE 3.2: Example of a table in our dataset

Processing the raw data provided by CodeLounge to reconstruct the companies' evolution allowed us to understand and refine the data model by identifying mismatches between the data and a set of assertions about the data model that we defined and encoded in the processing step. This process required a great amount of effort as to query the raw data we had to write complex queries to join the data from many different tables. Harnessing the time dimension in the data was particularly challenging. Ultimately, we were able to fix many inconsistencies and exclude from our dataset the companies for which we could not reconstruct the evolution. By iteratively improving the model, the quality of the data, and by defining strategies to address a specific subset of failures with a common cause, we managed to exclude just a few hundreds of companies. All the tables we defined in our dataset are stored in a separate database schema, which we call `hakken`.

The list of excluded companies is stored in a dedicated table, and the reasons for exclusion are the following:

- companies included in the list of companies but not in the `public.company_offices` table, where the data about a company being head or subsidiary is stored. These companies were excluded because they are not part of a business network
- companies with no active version in the dataset, probably because of a mistake in the dates extracted from the commercial register entries

- companies for which it was not possible to reconstruct the evolution because of inconsistencies in the data
- company offices not present in the `public.companies` table, these are companies we know have been active in the period of interest because they were mentioned in the commercial register entries but only in the context of other companies
- companies showing up both as head and subsidiary of another company. We assumed that this change cannot happen and we excluded these companies because the cases were too few to justify a change in the data model

The data provided by CodeLounge is stored in the `public` schema while the `hakken` schema contains more or less 30 tables we created about:

- master data not included in the CodeLounge dataset, or in a different format
- list of excluded companies
- company versions
- company networks
- data of the analysis session performed by the tool

According to the data model we defined, in the `networks` table we store all the existing edges between a head company and a subsidiary company. Time information about the newtorks are retrieved from the `company_versions` table. An edge was active at a specific point in time if and only if both the head and the subsidiary were active at that time. Moreover, a network may have had different head companies in different points in time. For this reason, each network is identified by its own identifier, and the head company at a specific point in time is derived by the state (i.e. active or inactive) of the head company itself. It is worth mentioning that in special cases a network can have multiple head companies at the same time, but we excluded these cases from our dataset.

In Table 3.3 we show some statistics about the dataset we created.

Attribute	**Count**
Companies	1,325,616
Companies part of a network	18,860
Versions of companies part of a network	49,073
Networks	17,293

TABLE 3.3: Dataset statistics

Tables 3.4, 3.5, 3.7, and 3.6 show the distribution of companies by canton, network size, tax rate, and type respectively. It is important to note that the distributions do not consider companies but company versions. As we are interested in the evolution of the companies over time, this allows us to show time-varying attributes using a simple bar chart. This approach will make even more sense when we will show how we designed interactive components starting from these simple charts to allow the user to explore the data in a more detailed way.

Type	**Count**
Head	10,695
Subsidiary	8,165

TABLE 3.4: Distribution of company versions by type sorted in descending order

Network Size	Count
MICRO	15,185
SMALL	1,732
MEDIUM	1,562
LARGE	221
XLARGE	160

TABLE 3.5: Distribution of company versions by network size sorted in descending order

Canton Tax Rate	Count
HIGH	16,456
MEDIUM	8,298
LOW	5,690

TABLE 3.6: Distribution of company versions by canton tax rate sorted in descending order

Canton	Count
ZH	2,787
VD	2,174
BE	1,724
GE	1,227
SG	1,207
LU	968
AG	897
TI	790
VS	782
ZG	739
FR	725
GR	669
BS	655
BL	597
TG	575
SO	496
NE	399
SZ	355
NW	233
AR	164
JU	154
OW	143
SH	136
UR	124
GL	96
AI	44

TABLE 3.7: Distribution of company versions by canton sorted in descending order

3.3 Modelling

Considering the complexity, vastity, and multi dimensionality of the data, the modelling process is a crucial step in order to make the information accessible. Considering the limited familiarity with the domain, we structured the modelling process in three main steps:

- explore the data to get a general understanding of the entities involved and domain rules,
- define the entities, relationships and how to represent them considering the time dimension, and
- refine the model by integrating the domain knowledge we acquired while facing exceptions to the rules we defined.

The data exploration process was conducted by querying the database using fairly complex ad-hoc SQL queries to extract statistics, patterns, and anomalies. The main entities involved in the data are: *companies*, *networks*, *locations*, and *legal forms*.

The most relevant attributes of the entities are:

- **Companies**: *EHRAID*, *Company Name*, *Legal Form*, *Network*, *State*, *Location*
- **Networks**: *Network ID*, *Head companies*, *Subsidiaries*

Company Name, *Legal Form*, *Location*, and *Network* are all time-variant fields of the *CompanY* entity. This means that a company can be part of many networks, but not at the same time. The *Network* entity is also time-variant, as a company can change network affiliation over time.

At first sight the domain model seems simple, but the complexity arises from the time dimension that is present many dimensions of the companies, which makes also networks time-evolving entities.

We explored and successfully applied a time-variant entity-relationship model inspired by HiSMo [1] to represent time-evolving attributes of the entities composing our model. The HiSMO meta model is shown in Figure 3.1.

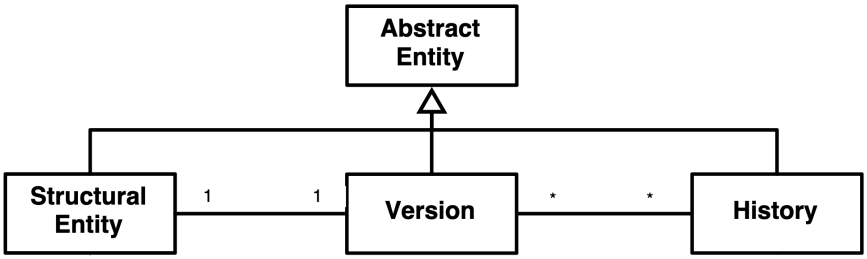

FIGURE 3.1: HiSMo meta model
Image adjusted from [1]

The meta model we devised inspired by HiSMo is shown in Figure 3.2 and a detailed version of the model is shown in Figure 3.3.

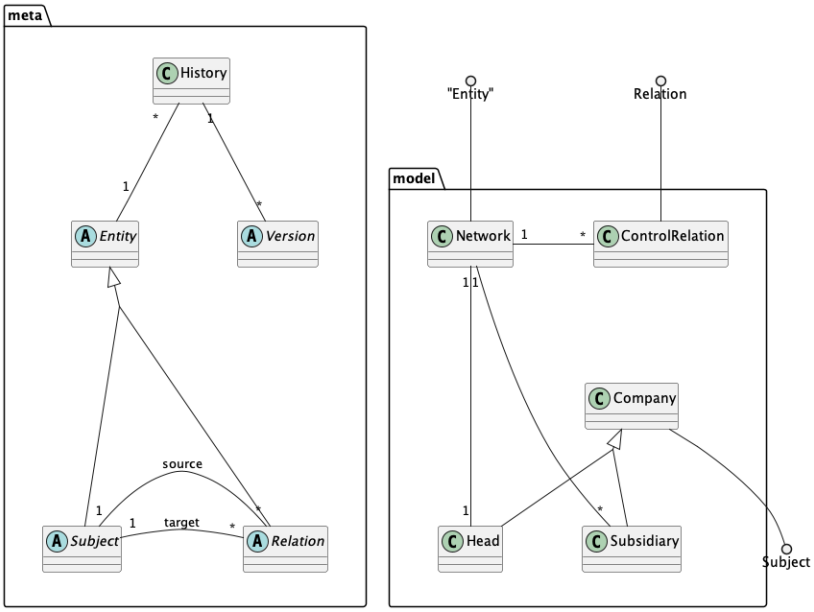

FIGURE 3.2: UML model of evolving company networks

The approach leverages the concept of History State Modelling (HiSMo) to reduce the complexity by splitting the main entity from its time-evolving attributes, which becomes concern of an instance of the `History` entity. Basically, instead of trying to represent the network as set of edges between company versions, where a new company version is a separate instance with one or more time-variant attributes changed, we represent the network as a set of edges between companies, and then we retrieve the company versions that are part of the network at a given time by the history of such companies. Also the structure of the network is time-variant, as companies can join and leave networks over time and the composition of the network at a specific time is the union of the active companies that are part of the network at that time.

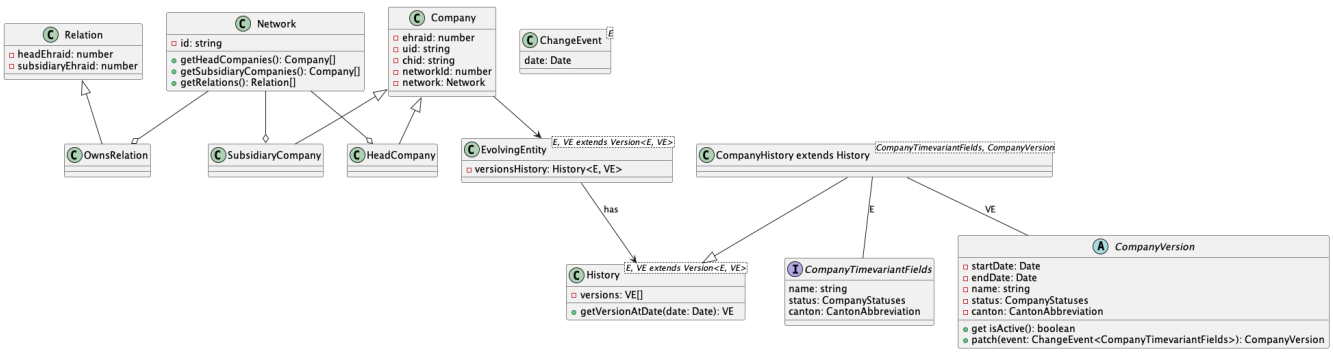

FIGURE 3.3: UML model of evolving company networks with attributes

3.4 Problem

This thesis aims to simplify the process of performing exploratory and explanatory analyses on the data at hand. Exploratory analysis involves understanding the data, calculating basic statistics about its distribution, and filtering or selecting companies based on user-defined criteria. Explanatory analysis focuses

on creating charts and tables that help users present their findings, with an emphasis on illustrating the evolution of selected data over time.

Conducting these types of analyses is challenging for several reasons. The complexity of the data often requires users to have both a solid understanding of the dataset and proficiency with analytical tools. For non-experts, relying on plain SQL queries or programming languages like R or Python is particularly difficult. Even for experts, these methods can be overly time-consuming and lack interactivity, reducing their effectiveness for iterative exploration.

The main challenges include:

- Time-varying data: Handling changes over time requires tracking historical versions and structures.
- Complex cases: Analyzing intricate histories, such as company mergers and reorganizations, can be difficult.
- Answering nuanced questions: Addressing complex, high-level questions often requires significant effort and sophisticated tools.

3.4.1 Examples of Challenges

Time-varying data

Determining the largest business network ever recorded would require merging data on network structures, the companies involved, and how these elements evolved over time.

Complex cases

Consider Cellere Bau, a company that has undergone multiple ownership changes, acquired numerous other firms, and changed its name several times. Tracking its evolution on platforms like Zefix is extremely difficult, and even with tools like Python or R, understanding its network would require significant effort.

Nuanced questions

An example of nuances question is examining the impact of tax changes across Swiss cantons on business networks. For example:

- Do tax rate changes lead to the creation of new companies?
- Do they attract headquarters or subsidiaries?
- How does a canton's centrality within the network shift compared to cantons that did not change their tax rates?

Answering such questions is particularly challenging without an interactive tool. In the following chapters, we will demonstrate how our approach can help users tackle this kind of challenges by analyzing a specific case study.

3.5 Summary

In this chapter we have introduced the domain we operate in this thesis. More specifically we described what the commercial registry is, what we mean by business networks, what are the business dynamics we consider, what tax competition between cantons is and why all this is of interest to economic researchers. We have also introduced the dataset we use, the modelling approach we take and the problem we want

to solve with this thesis. In the next chapter we will present our approach to solve this problem by applying the principles, techniques, and technologies described in Chapter 2 to the dataset and the problem described in this chapter.

Chapter 4

Approach

4.1 Conceptual Approach

The approach we propose to solve the problem described in the previous chapter leverages visual analytics techniques to provide researchers with a tool that allows them to explore and analyze the data from the Swiss registry of Commerce about companies part of a business network that have been active in the time period 2002-2021.

The approach we propose is based on the findings of an extensive literature review we conducted, which is summarized in Chapter 2. In particular we applied the principles of Schneiderman's Visual Information-Seeking Mantra:

- Overview first
- Zoom and filter
- Details-on-demand

We support most of the tasks Schneiderman identified:

1. overview,
2. zoom,
3. filter,
4. details-on-demand,
5. relate,
6. history, and
7. extract.

In this chapter we describe how we applied these principles to the problem at hand by providing screenshots of components of the tool we developed. In the next chapter we will show how we combined these components to provide a coherent tool that supports the tasks researchers need to perform.

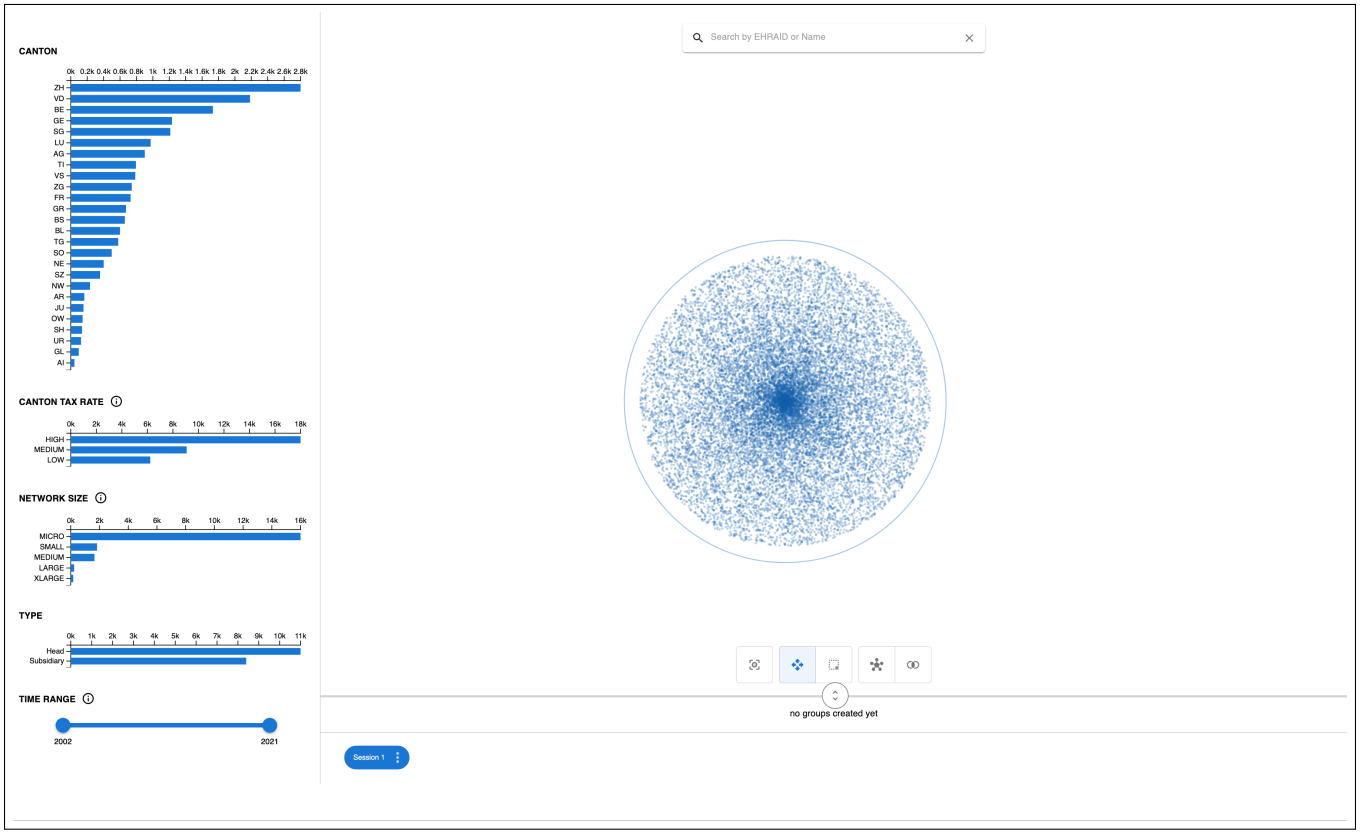

FIGURE 4.1: The main graph of the tool

The first principle we applied is overview first. This principle states that the user should be provided with an overview of the data before they can zoom and filter. In our case, we provide an overview of the data in the form of a network graph that shows nothing more than a node for each company like in Figure 4.1 and a set of charts that show the distribution of companies by different attributes like in Figure 4.2.

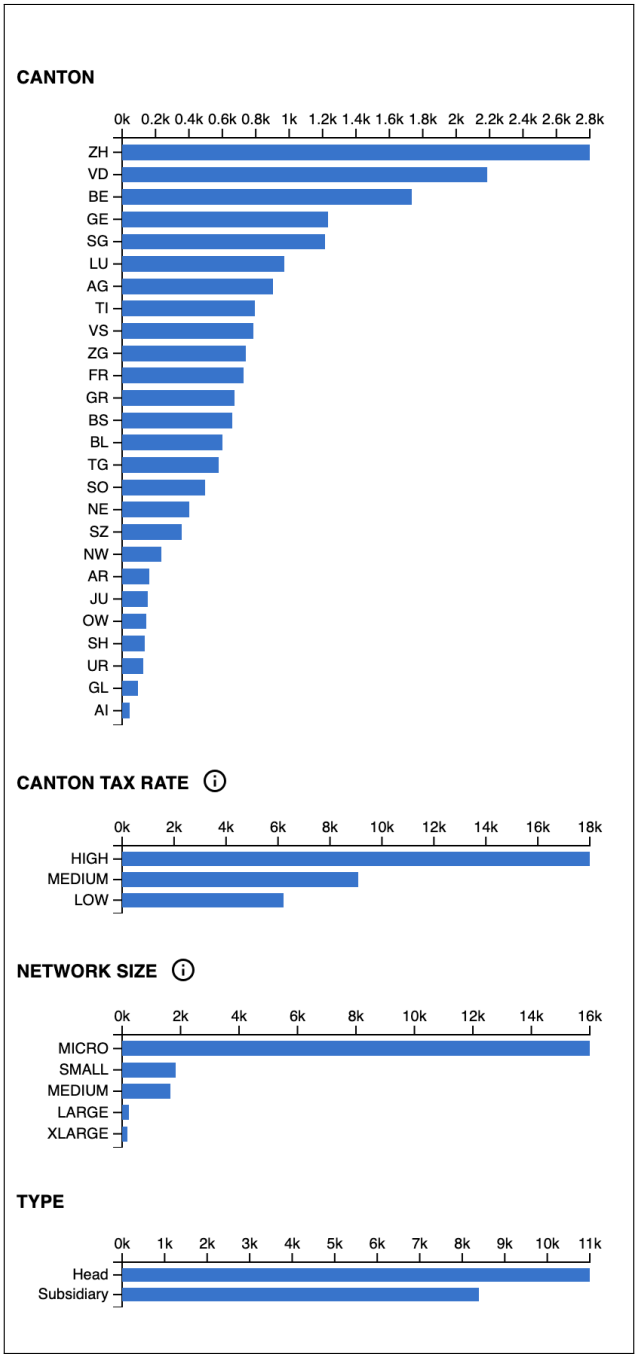

FIGURE 4.2: The distribution charts of the tool

The next principle we applied is zoom and filter. This principle states that the user should be able to zoom in on the data and filter it to focus on the data of interest. In our case, we provide the user with the ability to zoom in on the data by selecting a node in the network graph and see the details of the company represented by that node. An example of this is shown in Figure 4.4. We also provide the user with the ability to filter the data by selecting a range of values in the distribution charts like in Figure 4.3 or by searching for a specific company in the search bar.

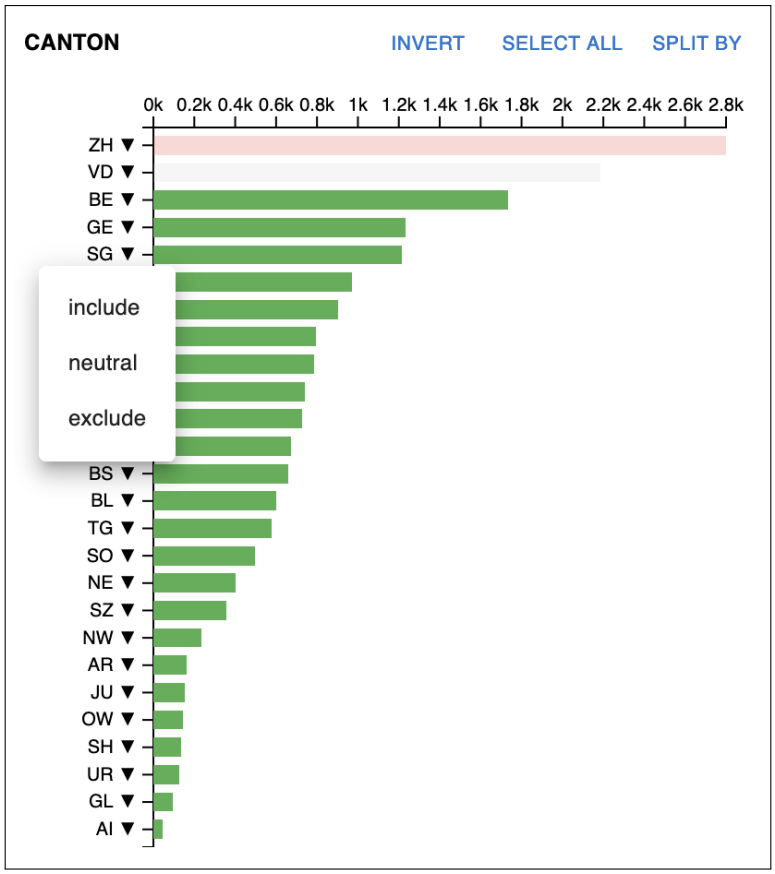

FIGURE 4.3: Filtering the data

The distribution charts in fact act as filters that allow the user to include or exclude companies matching a certain criteria. In Figure 4.3 we show the options the user can select to apply these filters. There are three options:

- include,
- neutral, and
- exclude.

What is the neutral option for? To answer this question we need to make a consideration about the fact that the data we are working with is time variant, reason why we chose to map the data in the barchart to the number of companies with at least one version matching the criteria. The easiest way to understand this is by considering just the include and exclude option, and experience the limitation of this approach first hand. Let us imagine that we have all criteria of the `canton` dimension set to include but we want to exclude all companies that have been in a certain canton. The user may be tempted to set the corresponding criteria to exclude, but this would not exclude all companies that have been in that canton, because some companies may have been in that canton at some point in time but not in the time period we are

considering. For this reason the user can select the *neutral* option to forcefully excludes companies that have been in a certain canton, or just not include them unless they match another selected criteria.

The third principle we applied is details-on-demand. This principle states that the user should be able to see the details of the data they are interested in. In our case, we provide the user with the ability to see the details of a company by selecting a node in the network graph like in Figure 4.4. The details of a company are shown as a series of versions in a timeline, the history mentioned in Section 3.3. We have a new version everytime an attribute of the company such as the name or location changes. The detailed view of a company shows also information about the variation of count of subsidiaries companies if the company is a head company and M&A events.

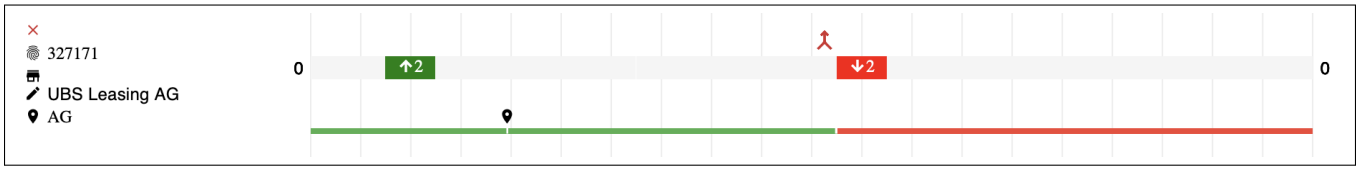

FIGURE 4.4: The details of a company. In the left-most column are shown the ehraid, an icon representing a head or subsidiary company (which is not a time-variant attribute), the name and canton at the beginning of the considered time frame. The rows above the rectangles representing the versions of the company show, from top to bottom: (1) the M&A events, (2) the initial count, the variations, and the final count of subsidiary companies, if the company is a head company, and (3) variations in name or location. Each version is coloured in light gray if the company was not founded yet,in green if the company was active, in red if the company was cancelled. Companies in liquidation are coloured in green because from an economic point of view a company is still active during the liquidation phase as it can run business operations in that period.

In the next chapters we will see how we combined these and other components and leveraged interactivity to support the analysis of our dataset. Before doing that, we describe how what tasks are supported by our approach and how.

1. **Overview**: the user can get an overview of the data by looking at the main graph and the distribution charts
2. **Zoom**: the user can zoom in on the data by selecting a node in the main graph
3. **Filter**: the user can filter the data by selecting a range of values in the distribution charts or by searching for a specific company in the search bar
4. **Details-on-demand**: the user can see the details of a company by selecting a node in the main graph, the user can also see the relationships between nodes as well as the intersection between goups of nodes. The user can also see details of a selection of companies by selecting a node, a relationship, or the intersection of groups of nodes
5. **Relate**: the user can see the relationships between nodes as well as the intersection between groups of nodes. The user can also create custom groups of companies and compare each group with the others in order to see the differences and similarities between them
6. **History**: the user can see the history of a company by looking at the timeline of versions of the company. The user can also see the history of a selection of companies by selecting a node, a relationship, the intersection of groups of nodes, or custom groups of companies
7. **Extract**: we do not support this task yet, but we defined how this could be done in the future. The user could extract the data by exporting the raw data in csv format, or by exporting the visualizations as an image

4.2 Implementation

The definition of the approach and the implementation of the tool have been interleaved activities we carried out in an iterative process that involved the following activities:

- identify meaningful questions about the data and identify interesting aspects about it
- perform data analysis to find answers to the questions and refine our understanding of the data and of the data model
- update the data model to better represent the data and update the data processing pipeline to extract and store the information we needed
- develop visualisations to answer the questions and to better understand the data
- integrate the visualisations in a user interface to allow the user to interact with the data
- refine the visualisations and the user interface to better answer the questions and to better support the user in the analysis of the data

We started by performing data analysis on the data stored in the database provided by CodeLounge to answer rather basic questions and understand the domain we were about to operate in. By looking at the results we obtained, more specific questions arose, questions whose answer could oftentimes be better understood by visualizing the data. We developed visualisations to answer these questions. Eventually, we wanted to be able to modify these visualisations on the fly to discover more, for example to dig deeper into a portion of the data shown in the general visualisations we created. This has been the moment when we decided to add the interactivity capabilities to the visualisations to allow filtering the data, zooming in, and selecting a portion of the data to see it in more detail.

The tool we present in this section accesses data we stored in a PostgreSQL relational database filled by a data processing pipeline we created, whose aim is to recreate and fill data into these tables from the original database by executing all the queries needed to process the data to extract the information we needed.

The execution of the data processing pipeline is orchestrated by a TypeScript application that leverages Prisma ORM to run simple queries, plain SQL queries to perform more complex operations, and a set of classes that implement the model we described earlier to manage the complexity of the data processed.

4.2.1 Processing Pipeline

The processing pipeline shown in Figure 4.5 is composed of the following steps which are executed in sequence and are better described below:

1. import master data
2. define companies to exclude
3. import data from CodeLounge database
4. parse companies to create networks
5. create company versions
6. create conditions and conditons on companies

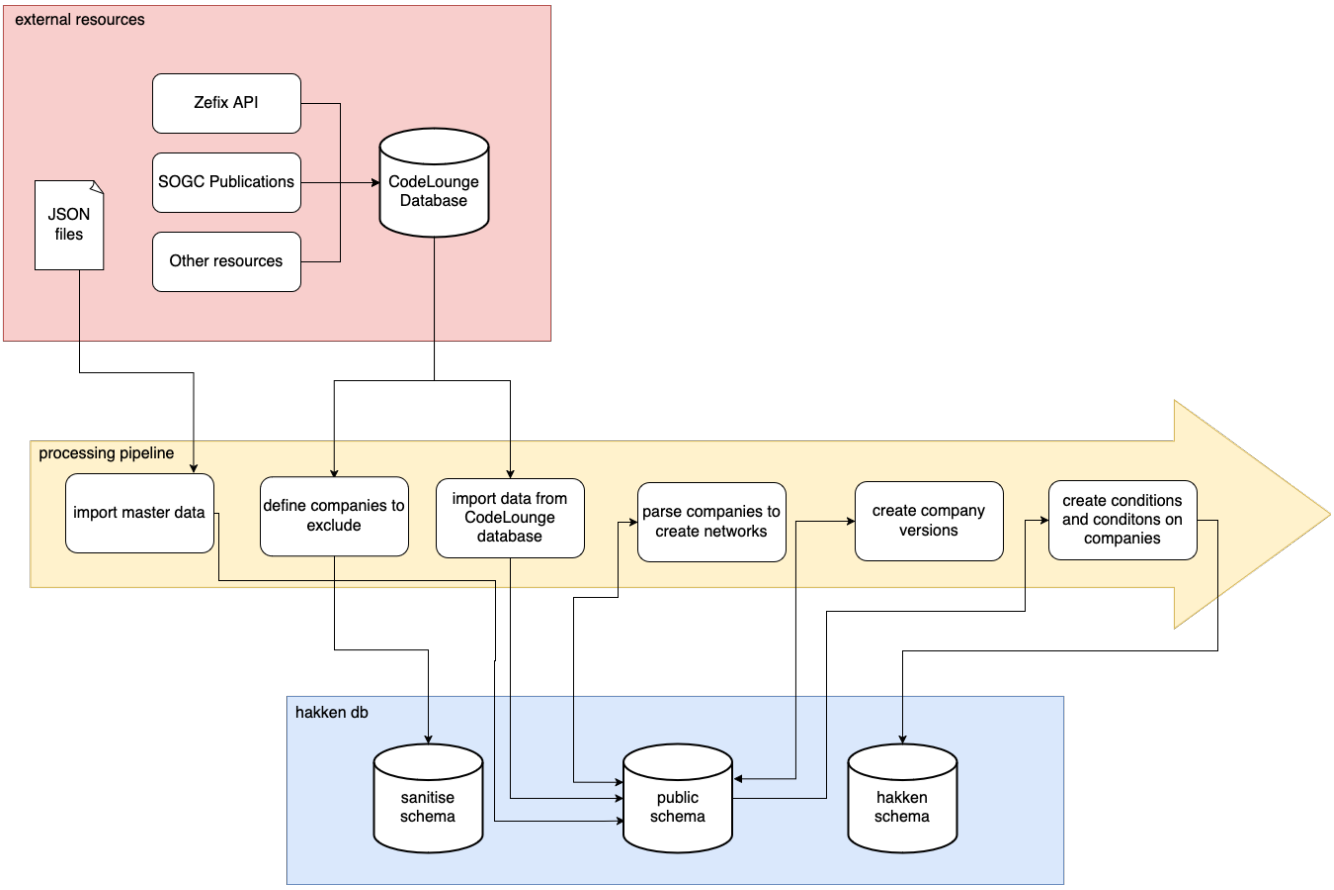

FIGURE 4.5: Processing pipeline architecture
Elements in the red area are externale resources. Elements in the yellow area are the steps of the pipeline. Elements in the blue are are part of the application database. The steps in the pipeline are executed in sequence following the order of the big yellow arrow.

Import Master Data

The first step of the pipeline is to import the master data, which is simple resources needed to process the data such as the list of cantons and the list of tax rates. This step is executed as a SQL script that inserts the data in the database.

Define Companies to Exclude

The second step of the pipeline is to define the companies to exclude. This step is executed as a SQL script that inserts the data in a dedicated schema named *sanitise*. The ehraid of the companies to exclude is stored in a dedicated table for each exclusion condition they match. The conditions are:

- companies that appear both as head and subsidiary
- company offices whose ehraid is not present in the *companies* table. This means that the company was mentioned as head or subsidiary in an entry of the Registry of Commerce but there is no data about it in the CodeLounge database
- companies with legal seat not associated to any canton, basically with the data available we cannot determine the canton where the company is located either because the data is missing or because the data is not consistent

- companies not in company offices. This means that there is data about the company but there is no data about the link with another company. This is a rather common case because all the companies that are not part of a network match this condition. Some companies though are part of a network but the data is missing. This is a case we want to exclude from the processing

Import data from CodeLounge database

The third step of the pipeline is to import the data from the CodeLounge database. This step is executed as a SQL script that inserts the data in the database. Part of the data imported in this step is the data already used in the previous step, but we import it after the exclusion of the companies to exclude. The data imported in this step is the following:

- companies
- company offices
- legal forms
- legal seats
- noga codes
- noga info
- company name changes
- company relocations

Parse companies to create networks

This and the next steps of the processing pipeline are more complex than the previous ones and for this reason we had to model the data and the operations to perform in a more structured way using a dedicated model implemented in TypeScript. In this step we aggregate information from different tables to reconstruct the networks of companies. The hardest part of this step is to handle the fact that the relations are time dependent. This means that a company can be a subsidiary of a network in a certain time range and then not be part of the network anymore because:

- the company was liquidated
- the company went through a merge or an acquisition
- the company became independent
- the head company was liquidated
- the head company went through a merge or an acquisition

Create company versions

In this step we create the company versions. A company version is a snapshot of the company at a certain point in time. The company version is created by aggregating the information about the company from the different tables. In this step we also use a dedicated model implemented in TypeScript to parse all the information about the company and create a list of events. The events are the changes that happened to the company. The list of events is then used to create the company versions.

Create conditions and conditons on companies

In this step we create the conditions which are basically the list of all possible values a property can have. The properties we are interested in are the following:

- canton
- tax rate
- type
- network size

The values of the canton property are all the swiss cantons.
The values of the tax rate property are buckets of tax rates we defined:

- high (greater than 20%)
- medium (between 15% and 20%)
- low (lower than 15%)

The tax rate values have been computed by Prof. Parchet aggregating the federal tax rate, the cantonal tax rate, the municipal tax rate, and other rates. Each company version has a tax rate which depend on the tax rate of the canton where the company was located in the time range when the version was valid.

The values of the type property are head and subsidiary. A company cannot be both head and subsidiary so all the versions must be consistent with the same value.

The values of the network size property are buckets of network sizes we defined: xsmall, small, medium, large, xlarge. The network size of a company is the size of the network the company was part of in the time range when the version was valid. The values correspond to the following ranges that we defined by looking at the distribution of the network sizes in the dataset and that we refined over time to have a meaningful distribution:

- xsmall lower than 5
- small between 5 and 10
- medium between 10 and 50
- large between 50 and 100
- xlarge greater than 100

4.2.2 Web Application

The other main component is the React web application written in TypeScript using NextJS. NextJS is a convenient-to-use framework that allows developers to create a hybrid application characterised by some server-side components and some client-side components. By using server actions, in NextJS we can perform operations on the server without writing a full fledged server application. For example the creation of the REST APIs endpoints is handled by the framework, reducing the overhead of creating a server application while maintaining all the advantages of a server application. The React application is a single-page application that uses the D3.js library to create the visualisations we developed.

In Figure 4.6 we show the architecture of the application. The application is composed of the following components:

- the database, which is a PostgreSQL database that stores the data processed by the pipeline in the *public* schema and the user data in the *hakken* schema
- the NextJS/React application that that uses the Prisma ORM to access the database and serves the user interface
- the partially pre-rendered pages that are served by the NextJS server and uses client side rendering and d3js to serve the interactive visualisations

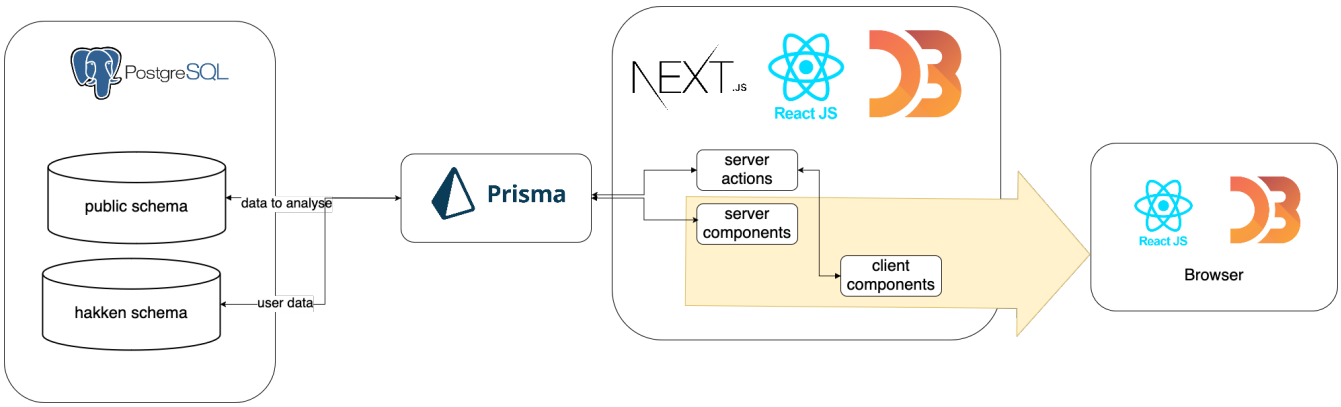

FIGURE 4.6: Application architecture

The first visualisations we created were created using Apache e-charts, a simple-to-use JavaScript library that allows to create a wide range of visualisations. The conciseness and simplicity of echarts comes with a price: the charts are not as flexible as lower level libraries like D3.js. The main limitations we encountered were:

- customisation of the visualisations e.g., position nodes in a force-directed graph,
- the ability to interact with the visualisations e.g., select a portion of the data to see it in more detail,
- the ability to create custom visualisations e.g., the evolution charts and the visualisation of the history of a company

We also tried vega, a declarative language to create visualisations, but we found it not really intuitive nor flexible enough for our needs. We then decided to switch to D3.js. The main limitation of D3.js is the steep learning curve, but the flexibility and the power of the library are unmatched and empowered us to create the visualisations we needed to not mention the ability to interact with the visualisations. There is a deep mismatch between the approach of D3.js and React, but we found a way to make them work together well. The mismatch is due to the fact that D3.js is a library that manipulates the DOM directly, while React is a library that creates a virtual DOM and then updates the real DOM. The default behavior of D3.js is to reuse as much as possible the existing elements while React's default behavior is to rerender the components whenever the state changes.

The user interface is composed of a set of panels that allow the user to explore the data, create custom groups with a selection of the companies, and compare the evolution of some properties of these groups.

The user interface is composed of twho main panels: the main graph and the evolution analysis. The main graph is the entry point of the application and allows the user to explore the data and create custom groups of companies. More specifically using the main graph the user can:

- search a specific company by ehraid or name (past names are considered as matches)
- show relations and intersections between populations
- add nodes to the temporary selection
- create custom groups with companies arbitrarly selected from the graph or from the temporary selection
- open the inspection panel to see the history of a single company by clicking on a node
- open the inspection panel to see the history of arbitrarly selected companies using the brush and the dedicated button
- open the panel where the evolution charts can be accessed

The evolution analysis panel allows the user to compare the evolution of the count of companies and the count of relations between the companies in the custom groups.

The main graph

The main graph visualisation shown in Figure 4.7 is the panel presented to the user when the application is opened first. In this panel each company in the dataset is mapped to a dot. The user can select a company by clicking on the corresponding dot. To find a specific company among the more than 18'000 shown the user can search by name or ehraid using the search bar on top. The dots are all enclosed in a big circle which represent the population of companies matching all the criteria listed in the distribution charts. Initially, all conditions are selected, so the population contains all the companies in the dataset. The charts on the left provide information about the distribution of the companies with respect to their properties. By clicking on a population the distribution charts act as filters 4.8 and the population can be refined by changing the state of the conditions mapped to each bar of the filters. E.g., a user may exclude companies matching certain conditions. The user can also split a population by a dimension e.g., splitting a population by canton creates a new population for each canton whose condition is marked as `included`.

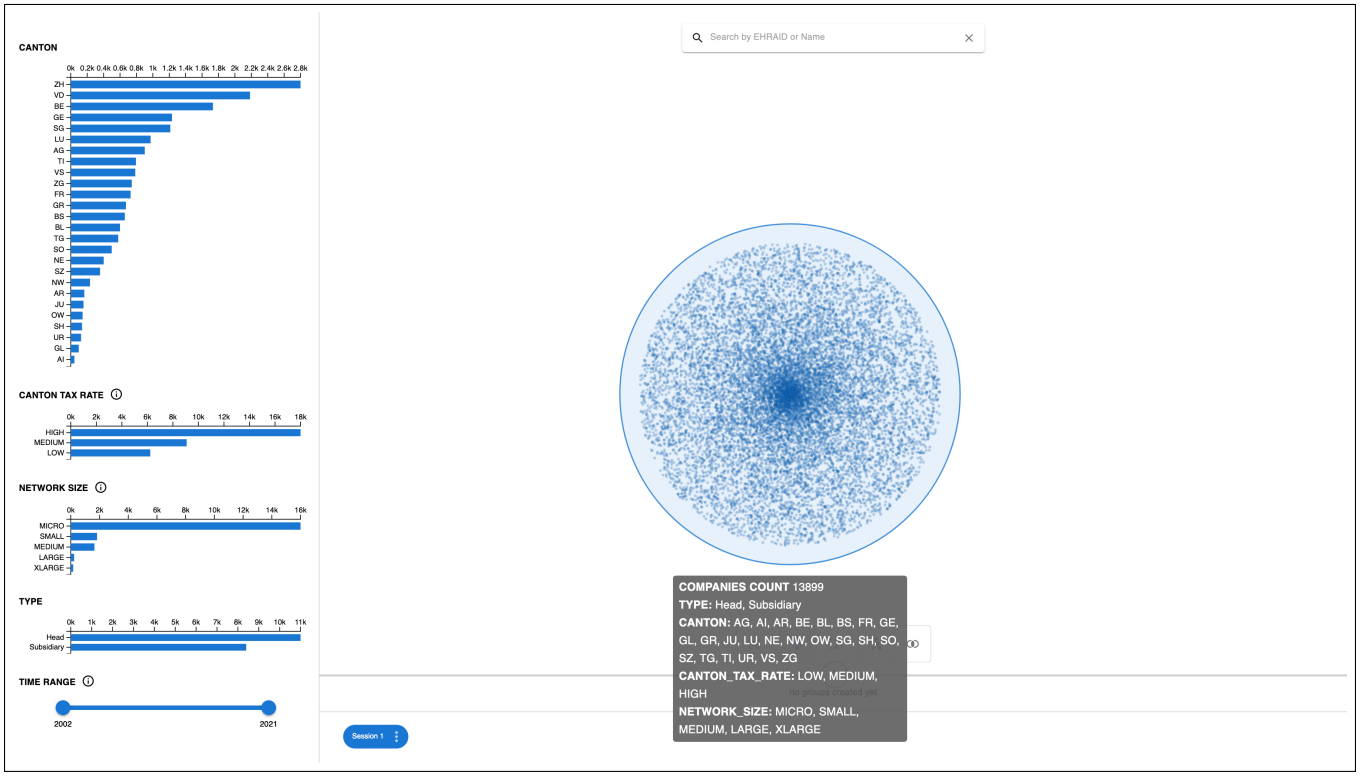

FIGURE 4.7: The entry point panel

The bars of the distribution charts can be colored with different colors, depending on the context:

- blue is used when the charts are used to show the distributions and do not act as filters
- green is used when the charts are used as filters and the condition is marked as `included`
- gray us used when the charts are used as filters and the condition is marked as `neutral`
- red is used when the charts are used as filters and the condition is marked as `excluded`

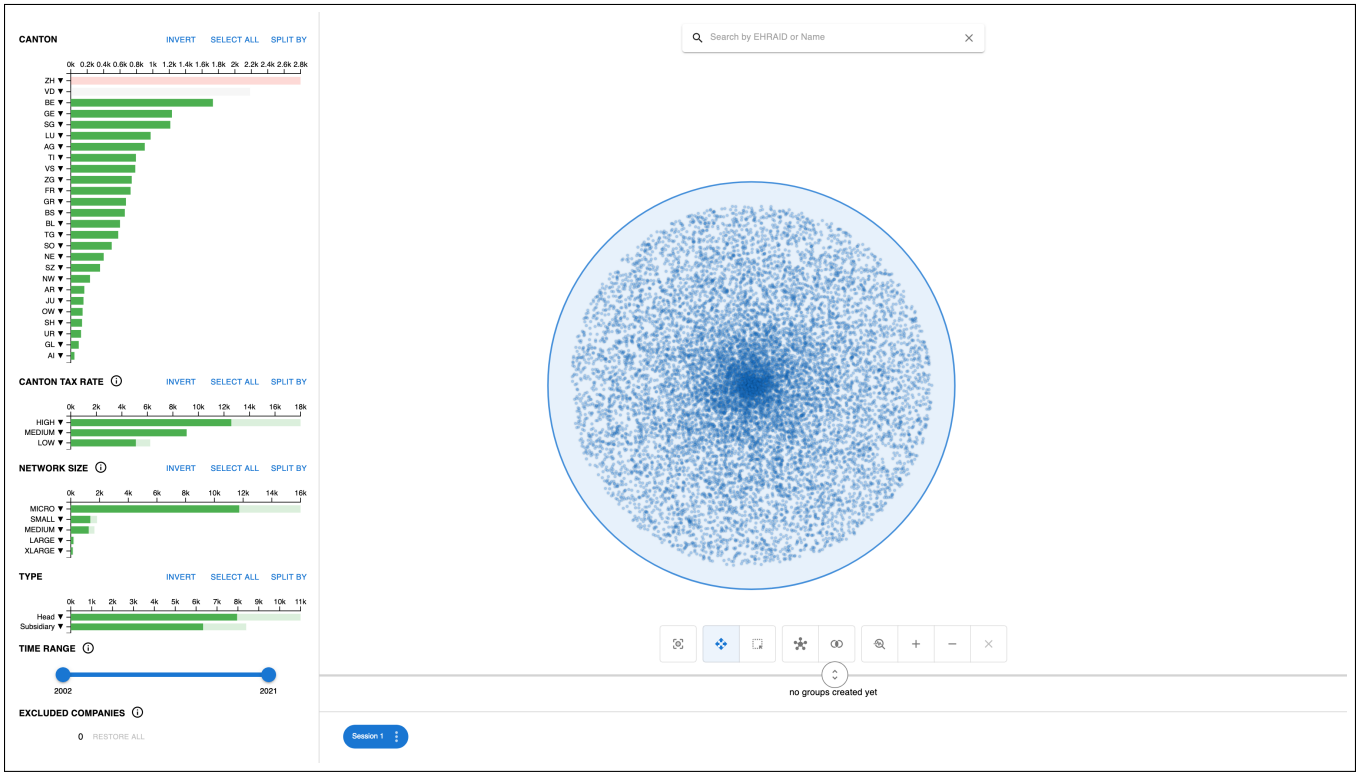

FIGURE 4.8: The entry point panel with a population selected

Additionally, in the filters there is a time range selector which can be used to show only the companies that have been active in the selected time range. As the distributions charts are repurposed when the user selects a population, the time range selector is also repurposed and allows to filter the companies in the selected population by the time range. This may come handy for example if the user wants to compare two populations with same conditions but different time ranges.

Relations and Intersection

By default, relationships and intersections are not shown in the main graph to reduce visual clutter. The user probably do not want to focus on those aspects at first and focus on the creation of different populations based on different criteria first. We decided to leave the user control over what to show in addition to the dots representing the single companies. The user can toggle on and off the relationships and the intersections using the buttons in the control bar placed at the bottom.

As the reader may notice by looking at the relationships visualisation shown in Figure 4.9, we do not show relations between single companies, but only between populations. We show the count of outgoing, ingoing, and self referencing relations. The latter being relations among companies in the same population. The reasons for this choice are

- too often showing so many relations between single companies would create excessive visual clutter,
- the user is probably interested in selecting the bunch of companies part of the relations at this stage of the analysis and dig deeper into the relations using more advanced visualisations which we decribe later.

In this visualisation the user can select the companies part of a relation by

- creating the populations according to their criteria,

- toggle the relations on,
- click on the count placed on top of the relation line (ingoing, outgoing, or self referencing).

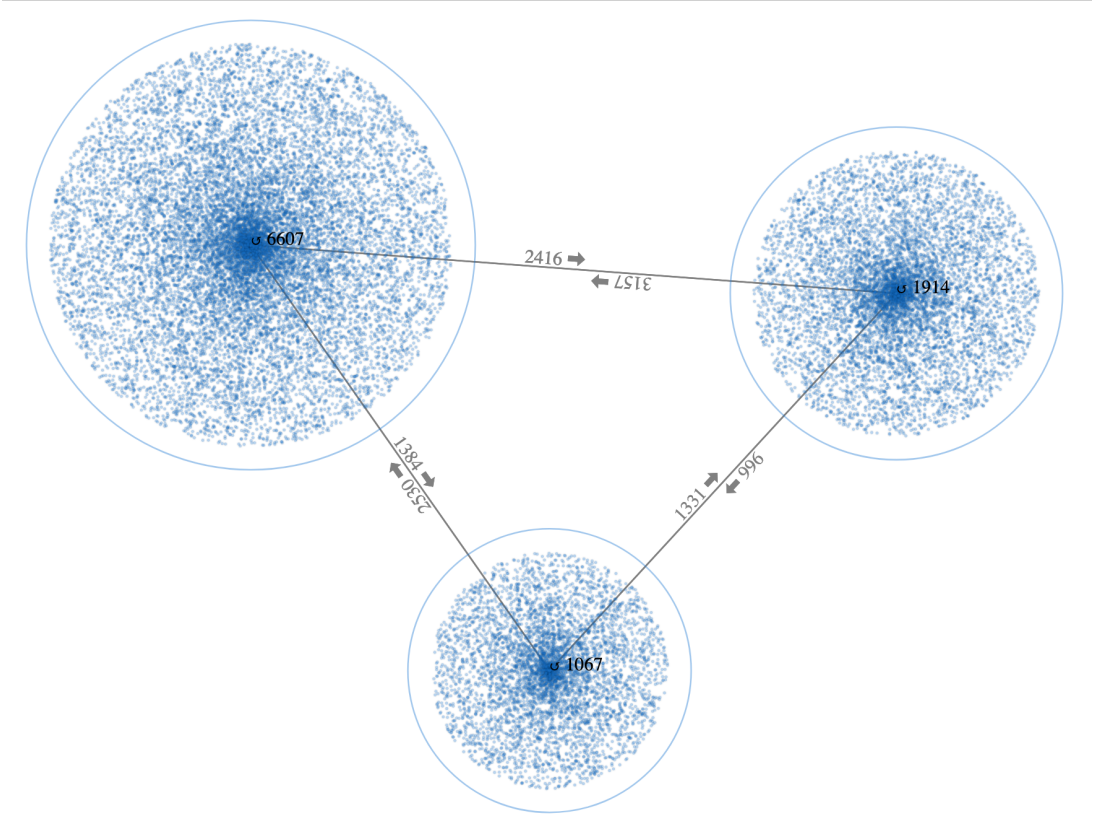

FIGURE 4.9: The entry point panel with relations shown

A similar worflow can be used to select the companies part of the intersection of two populations, that are represented as shown in Figure 4.10.

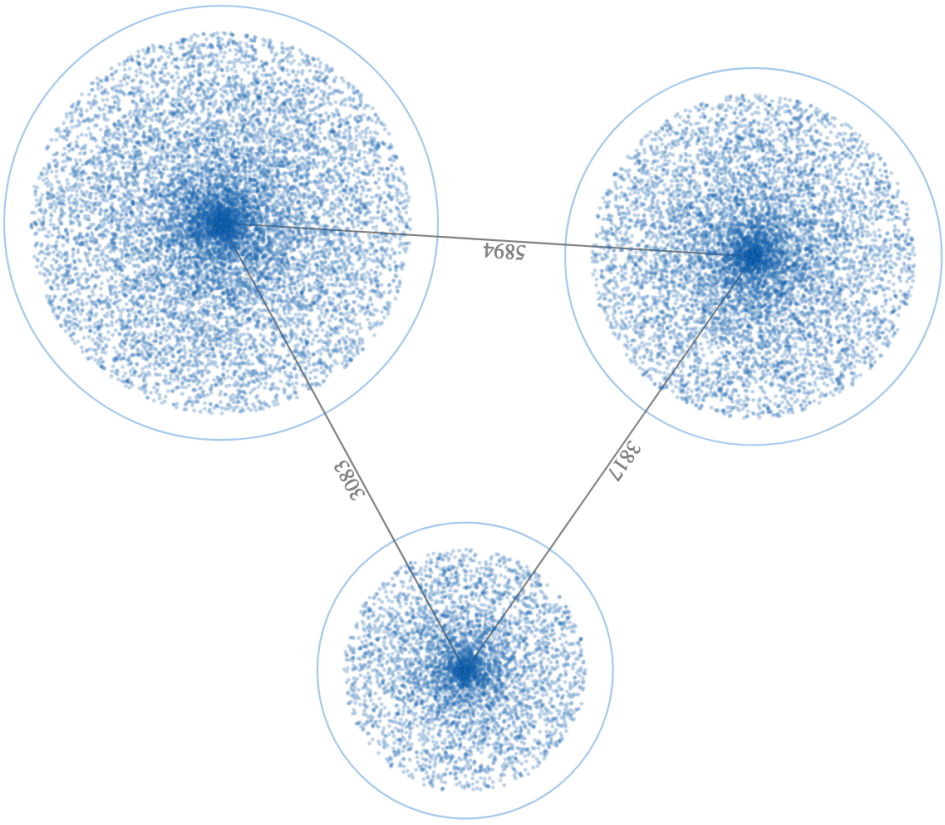

FIGURE 4.10: The entry point panel with intersections shown

Inspection Visualization

The main graph visualisation with optionally shown relations and intersections, which we described in the previous sections, is of great support to the user in the exploration of the dataset but it lacks the ability to:

- show many details that are present in the dataset but hidden in the main visualisations, and
- to provide an historical perspective on the selected companies.

For this reason we designed a component that allows the user to see the history of one or more companies with all the changes, events, etc. that happened to them. This component is shown in Figure 4.11 and can be accessed by clicking on a multitude of other components, making it a central component of the analysis. The user can for example inspect:

- a single company by clicking on the corresponding dot,
- a whole population by clicking on the circle enclosing the dots,
- a relation by clicking on the count of the relation,
- an intersection by clicking on the count of the intersection,
- the results of a research done using the search bar, and

- a custom selection of nodes by selecting them with the brush tool as shown in Figure 4.12.

FIGURE 4.11: Inspection of multiple companies

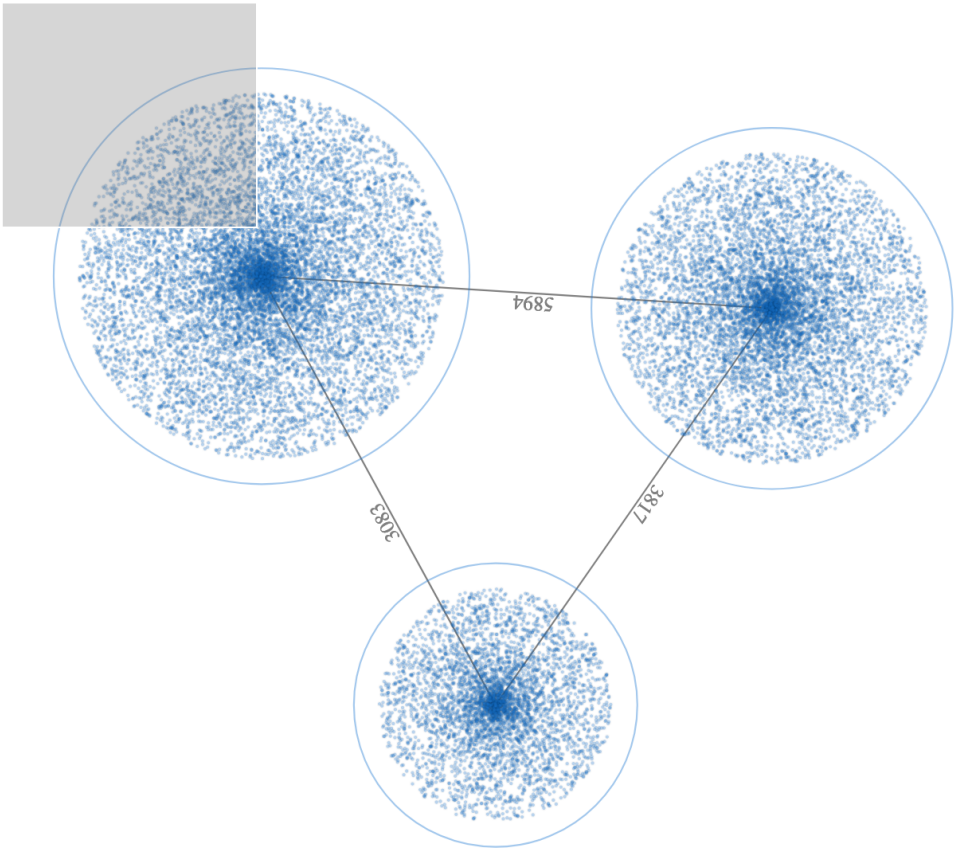

FIGURE 4.12: Brush selection

Temporary Selection and the Creation of Custom Group

Despite having a visualisation to explore all the companies and a visualisation to inspect each of them in detail, with the visualisations described until now the user cannot answer advanced questions about the dynamics of the companies in the dataset. For example, the user may want to compare the Year-over-Year evolution of the count of companies in the dataset that are active in the canton of Ticino with the companies in the canton of Zurich. To answer this question we allow the user to create custom groups of companies and perform more advanced analysis on them. These groups can match the populations the user created in the main visualisation, or they can be created by the user by selecting the companies with the brush tool. The user can add companies to a temporary selection, show in the top right corner in Figure 4.13, and then create a custom group, shown at the bottom also in Figure 4.13.

FIGURE 4.13: Temporary selection and custom group creation. The count of companies part of the temporary selection is shown in the top right corner fo the interface, along with the buttons to create a group out if it and the button to clear the temporary selection. The custom created groups are shown in the bottom part of the interface. In this screenshot there is a custom group called *custom-group-1*. The coloured dot shows the color assigned to the group itself which is used to visually link other elements that we will show later such as the lines of the line charts described in Section4.2.2. Close to the name of the group we show the counts of head companies, subusidiaries, and the total amount of companies part of the group.

Evolution Analysis

As mentioned in the previous section, on the custom groups the user can perform a more in-depth analysis, focussing on the evolution over time of certain properties of the groups. The data to conduct these analysis is shown as linecharts, heatmaps, or with the companies evolution table used to show the history of the companies described in Section 4.2.2.

Comparing Population Count Evolution

This chart shows the evolution of the count of companies in the selected custom groups. The data can be shown as a linechart (Figure 4.14) or as a heatmap (Figure 4.15). The user can optionally normalise the data, this is useful when the user wants to compare the percentage variation with respect to previous point in time instead of the absolute variation.

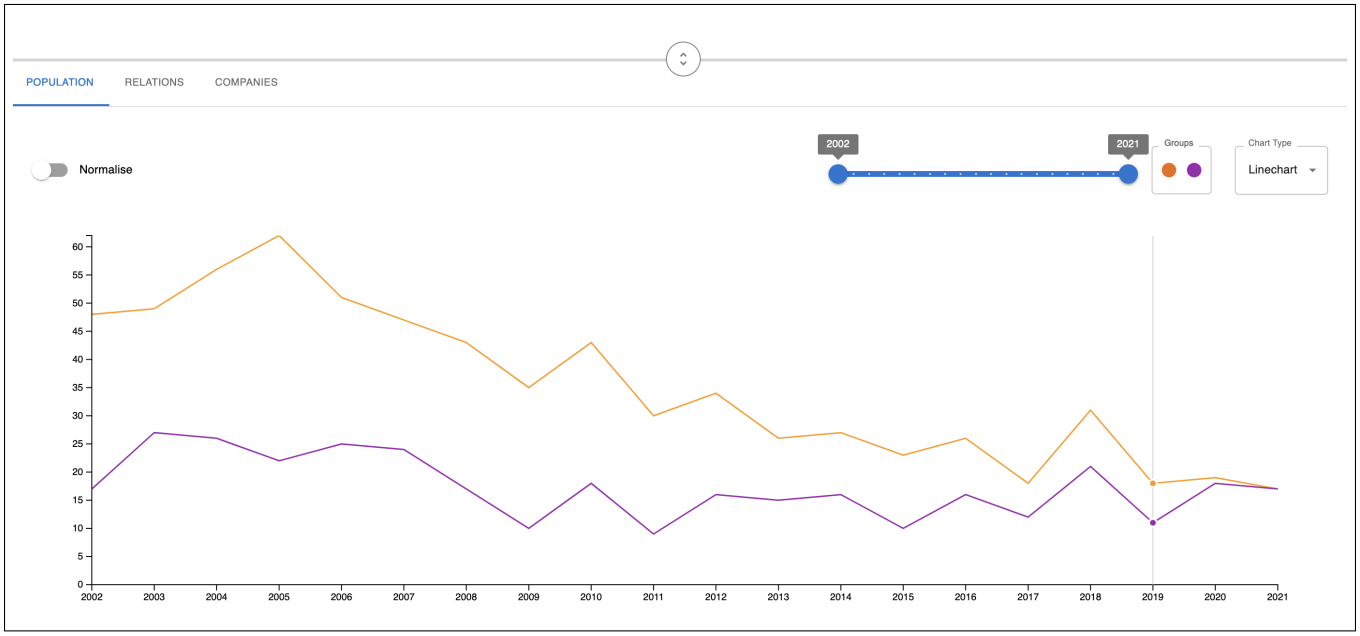

FIGURE 4.14: Population evolution linechart

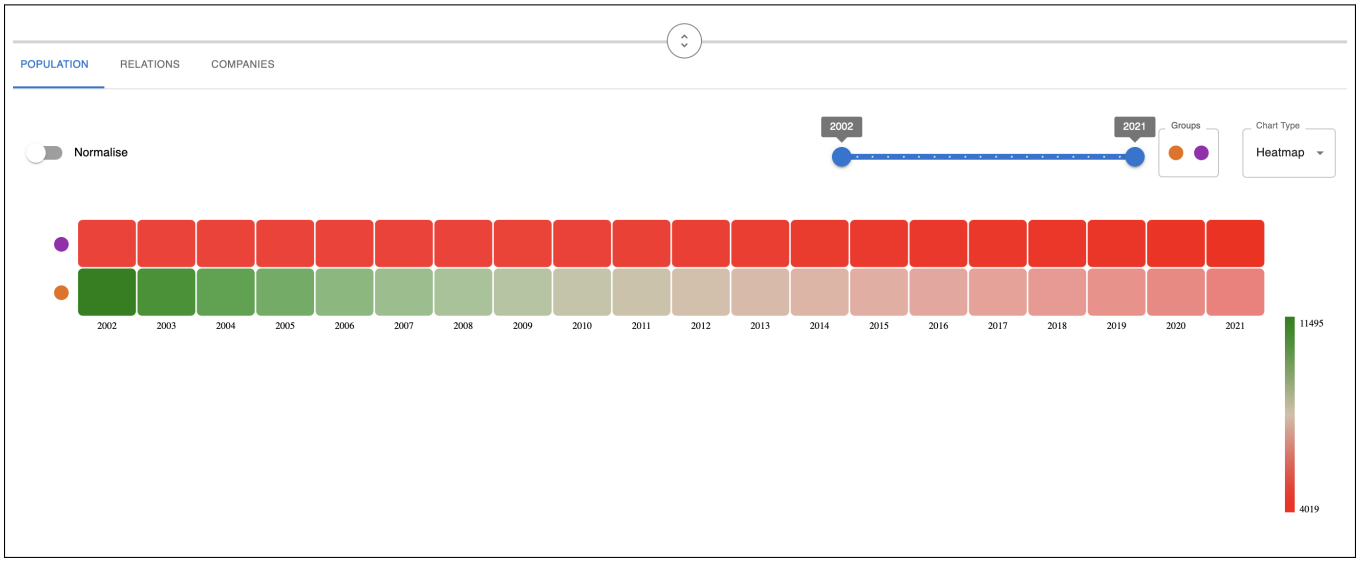

FIGURE 4.15: Population evolution heatmap

Both the linechart and the heatmap are interactive charts. The user can:

- adjust the time range,
- quickly hide certain groups, or
- select companies making up the count at a specific point in time of a line or a cell of the heatmap.

Comparing Relations Count Evolution

Another aspect of particular interest is the evolution of the relations between the companies in the custom groups. Again, the data can be shown as a linechart 4.16 or as a heatmap and the same interactions available

for the charts described in the previous section apply to these charts as well. Additionally the user can optionally hide self-refering relations as well as "always zero" relations. These are relations having always a count of zero.

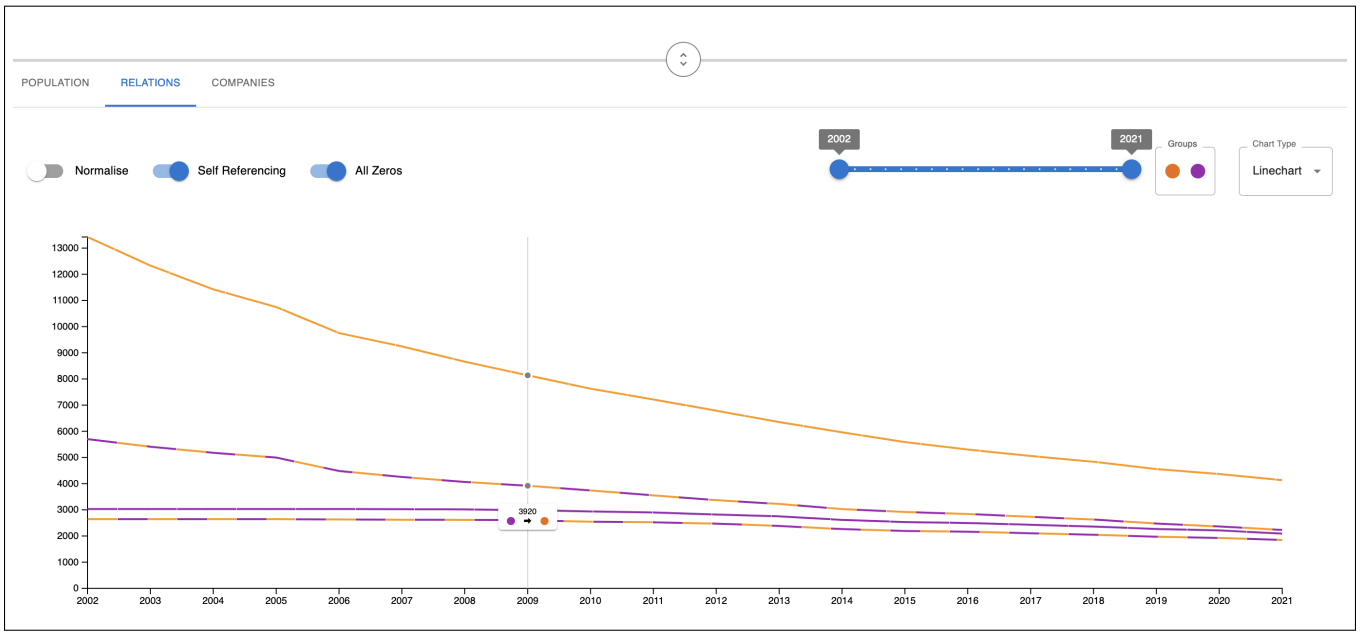

FIGURE 4.16: Relations evolution linechart

If a relation has as a source group a group containing only subsidiary companies, we hide the line by default because there is no semantics in the relation.

To distingish between the lines of self referencing and between-groups relations we use a color encoding that works as following: if the relation is self referencing the line is colored with the color assigned to the group itself. If the relation is between two groups the line is colored with a dashed pattern, with the first section colored with the color of the source group and the second section colored with the color of the target group.

Sessions Management

In the bottom part of the user interface there are one or more tags similar to the one shown in Figure 4.17. Each of these tags represent an analysys session that the user can create as a copy of the session they are working on in order to save a check point and continue the analysys in a different session. This mechanism allows the user to jump back and forth between the created checkpoints. In addition the user can:

- restore a session to the initial state i.e., all the companies are shown and the conditions in the filters are all selected
- detete a session
- rename a session

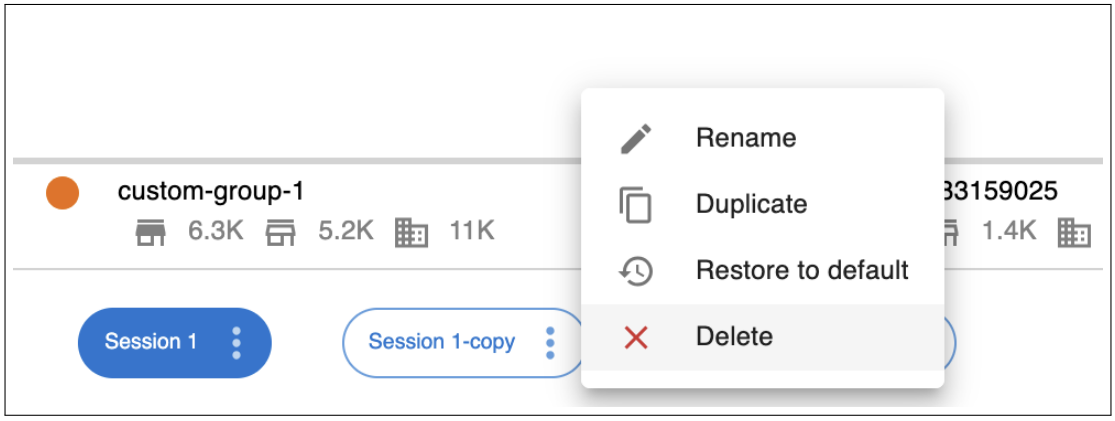

FIGURE 4.17: Sessions tags with open menu to show the menu options

4.2.3 Summary

In this chapter we presented the tool we developed following the principles and techniques described in the previous chapter in order to solve the problem at hand. The tool is composed of a data processing pipeline that extracts the information we need from the data provided by CodeLounge and a web application that allows the user to interact with the data and to perform analysis on it. The tool is composed of two main components: the main graph visualisation and the evolution analysis. The main graph visualisation is the entry point of the application and allows the user to explore the data, create custom groups of companies, and inspect the companies. The evolution analysis allows the user to compare the evolution of the count of companies and the count of relations between the companies in the custom groups. In the next chapter we provide an evaluation of the tool we developed.

Chapter 5

Evaluation

The evaluation phase is a critical component of this thesis, aimed at assessing the effectiveness, usability, and applicability of the approach we propose. To ensure the tool met the specific requirements of the domain, an iterative development and evaluation approach was adopted, incorporating continuous feedback from a domain expert. This approach enabled a dynamic cycle of development, reflection, and refinement, ensuring that the tool evolved in alignment with the needs and expectations of its intended users.

Weekly meetings with the domain expert formed the backbone of the evaluation process. These meetings served as checkpoints to discuss progress, present solutions to challenges encountered, and validate the functionality and relevance of the tool. Feedback gathered during these interactions played a pivotal role in shaping the overall design of the tool.

It is important to note, however, that the evaluation process did not include a formal experiment with real users due to time constraints. While such a study would provide valuable insights into the tool's practical impact and usability, it was not feasible within the timeframe of this project. Instead, the effectiveness of the tool is demonstrated through a detailed presentation of some use cases. This includes screenshots and descriptions illustrating the tool's functionality and how it addresses the needs of its intended users.

In this chapter we provide an anecdotal evaluation of the tool's effectiveness based on the provided use cases, offering insights into its strengths, limitations, and potential areas for future enhancement.

5.1 Basic Use Cases

The tool was designed to support a range of use cases, from simple queries to more complex analyses. To demonstrate its effectiveness in this Section, we present a series of basic use cases that showcase the tool's core functionalities, in the next Section

5.1.1 Use Case 1: Single Company

The first use case involves tracking the evolution of a specific company over time. By entering the name of a company or a specific ehraid in the search bar the user filters the population of companies to display only those matching the search criteria. An example is shown in Figure 5.1.

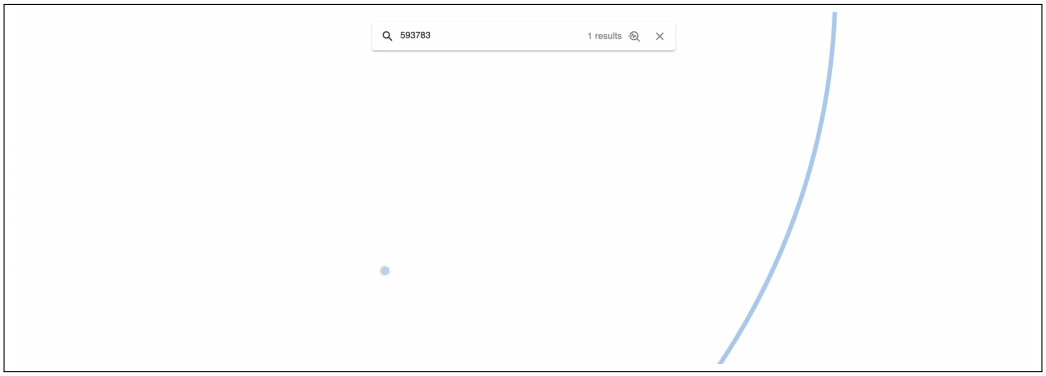

FIGURE 5.1: Using the search bar in the upper part of the interface, the user filtered the nodes searching for the company matching the ehraid 593783. The main graph visualisation is zoomed to show clearly the single result shown as node in the graph.

At this point, the user can click on the company of interest to view its details. This visualisation shows the evolution history of the company as described in more detail in Section 4.1. The history of this company is shown in Figure 5.2.

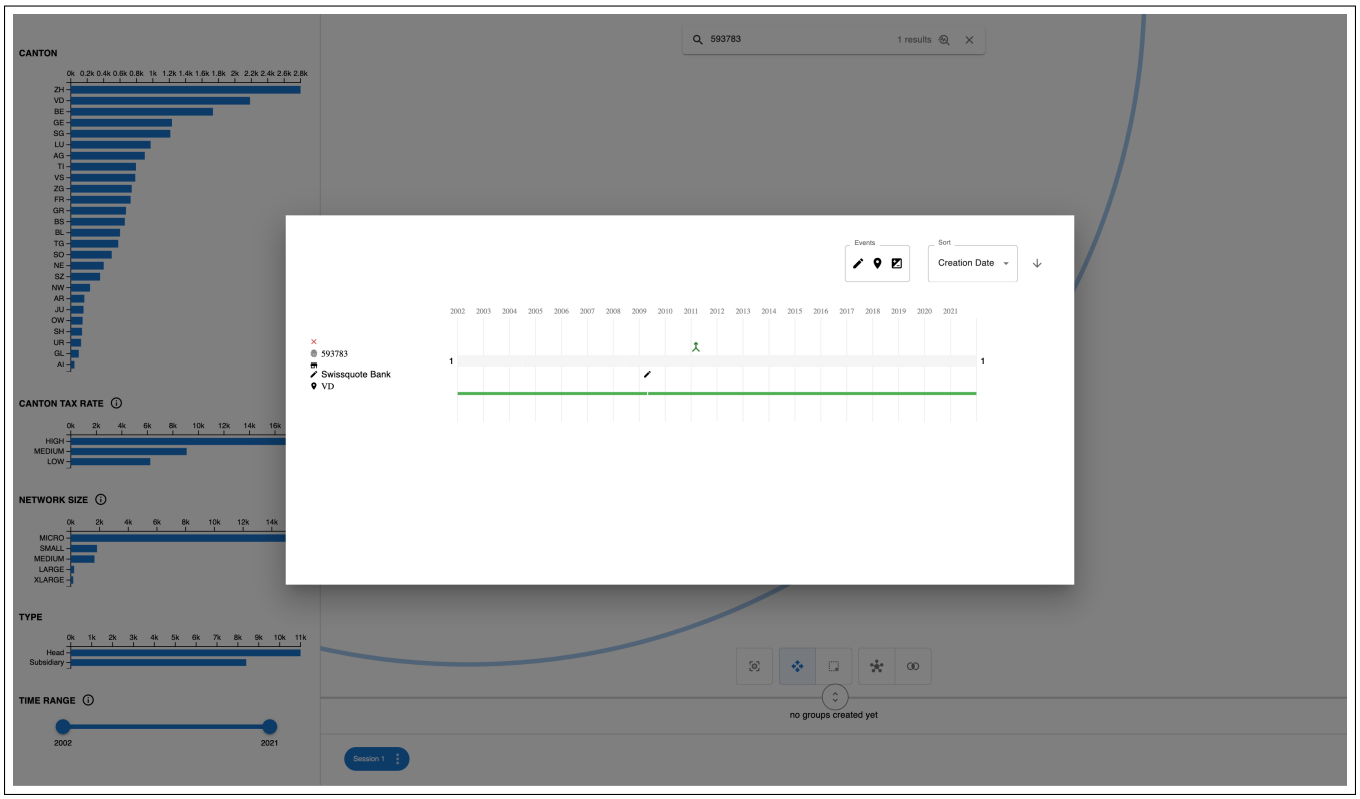

FIGURE 5.2: Using the search bar in the upper part of the interface, the user filtered the nodes searching for the company matching the ehraid 593783. The main graph visualisation is zoomed to show clearly the single result shown as node in the graph.

5.1.2 Use Case 2: Business Network

The second use case focuses on identifying and exploring the largest business network in the dataset.

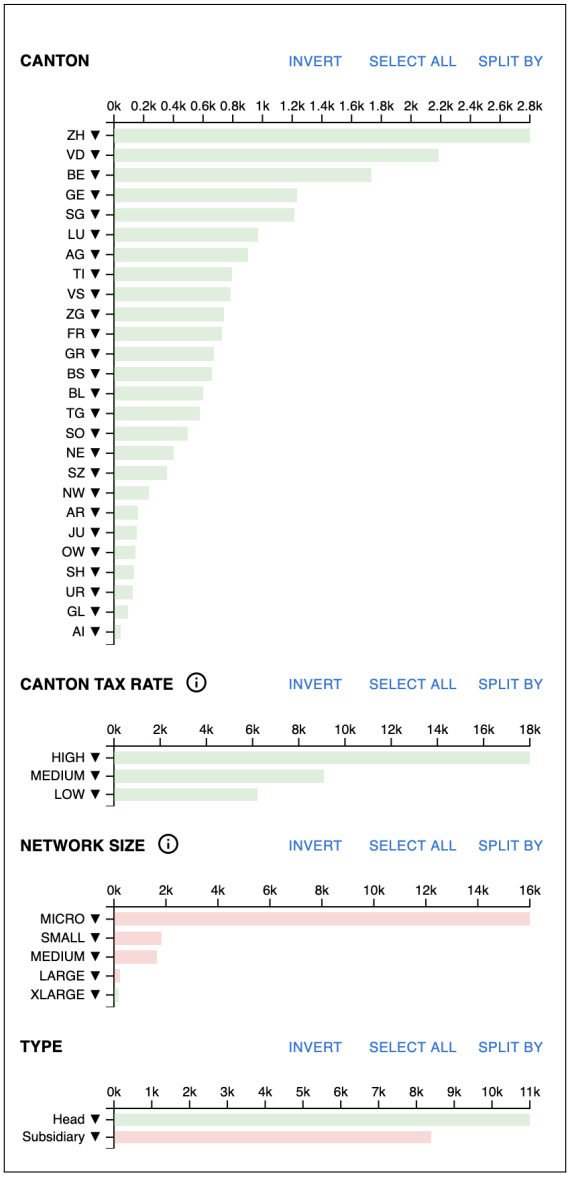

FIGURE 5.3: Use case 2: Filters applied to isolate the head companies of the biggest networks

By applying filters to the dataset as shown in Figure 5.3, the user can isolate the head companies of the largest networks and then use the inspection tool to compare the evolution of the count of subsidiaries. The head company of the biggest network is Kuoni Immobilien AG, whose evolution is shown in Figure 5.4.

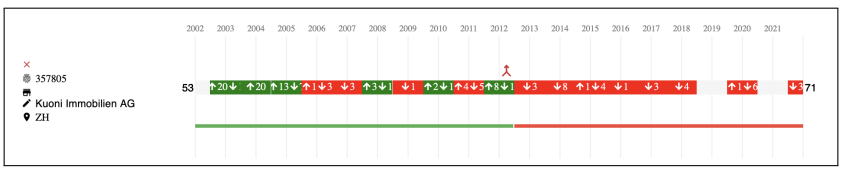

FIGURE 5.4: Use case 2: The evolution of the head company of the biggest network whose head company has the ehraid 357805

5.1.3 Use Case 3: Domain Comprehension

The third use case involves gaining understanding of the domain by leveraging the tool's exploratory analysis features. By creating different populations, the user can compare the characteristics of companies across different dimensions, such as canton, tax rate, or type. As an example we show here how to find out if there is any company that changed type i.e., from head to subsidiary. The first step is to create two populations, one containing all head companies, the other containing all the subsidiaries. This can be achieved by selecting the default population that contains all the companies and then splitting it by type. This operation can be performed by clicking on the *Split By* button in the distribution charts as shown in Figure 5.5. Performing this operation splits the selected population anc creates one for each value of the property the user is splitting on. As shown in Figure 5.6, in this case two populations are created because the property *type* has two possible values.

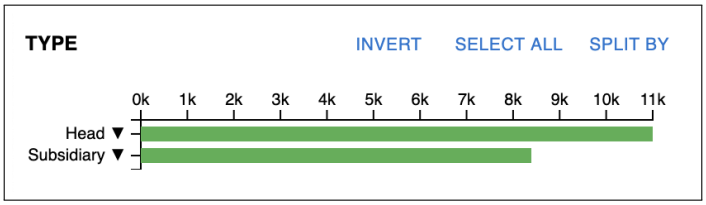

FIGURE 5.5: Use case 3: The Split By button in the distribution charts

After having created the two populations, the user can show the intersections between all the populations. Since there is no line connecting the two populations, any company has ever changed type from head to subsidiary or vice-versa. An example of existing intersection is shown in Figure 4.10. The result of this task is aligned with the assumptions we did in the preprocessing step. In fact, according to the data in the dataset there were 16 companies that changed type but we manually inspected them and that was due to errors in the data.

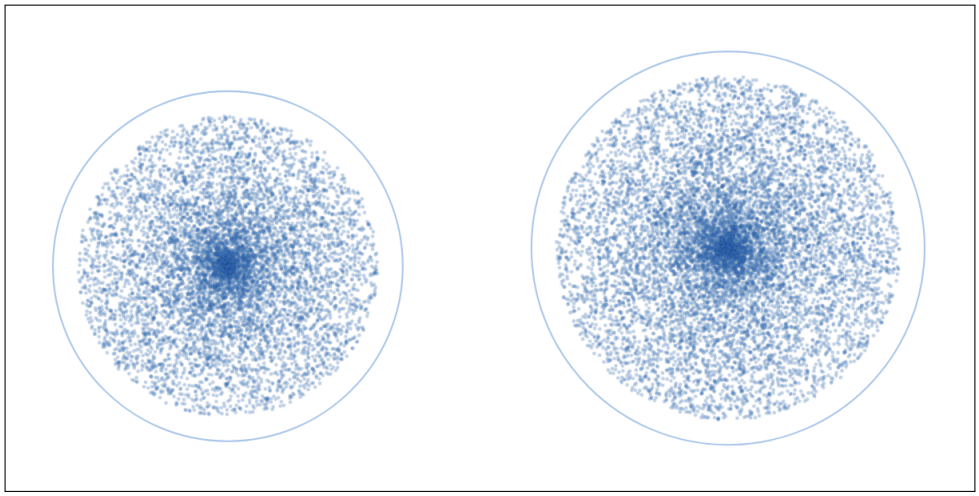

FIGURE 5.6: Use case 3: Result of splitting the default population by type

5.2 Advanced Use Case

In addition to the basic use cases mentioned in the previous section, the tool supports more advanced analyses that require a deeper understanding of the data and its underlying structure. These use cases leverage the tool's evolution analysis features to uncover insights that would be difficult to obtain through manual exploration.

As an example, we present a use case about regional variations in corporate tax responsiveness in Switzerland described by Krapf and Staubli [28].

Before 2010, the canton of Luzern embarked on a series of tax reforms aimed at lowering effective tax rates to attract businesses from other cantons—especially Zug, which at the time was significantly cheaper than any other canton in Switzerland. The competition with Zug was deliberate and explicitly stated, as noted in the newspapers from that period.

To evaluate the real-world impact of these reforms, we conducted a two-step analysis:

1. Exploratory Analysis: Using the exploratory analysis features of the tool, we identified relevant companies and created three custom groups by applying filters to the dataset that includes all companies that have ever been part of a business network active in Switzerland show in Figure 5.7
2. Evolution Analysis: Using the evolution analysis features of the tool, we compared the growth in the number of active companies within these groups over time show in 5.8

The three groups for the analysis included companies active between 2008 and 2012 in:

1. the canton of Zug,
2. the canton of Luzern, and
3. the canton of Basel-Stadt, which served as the control group

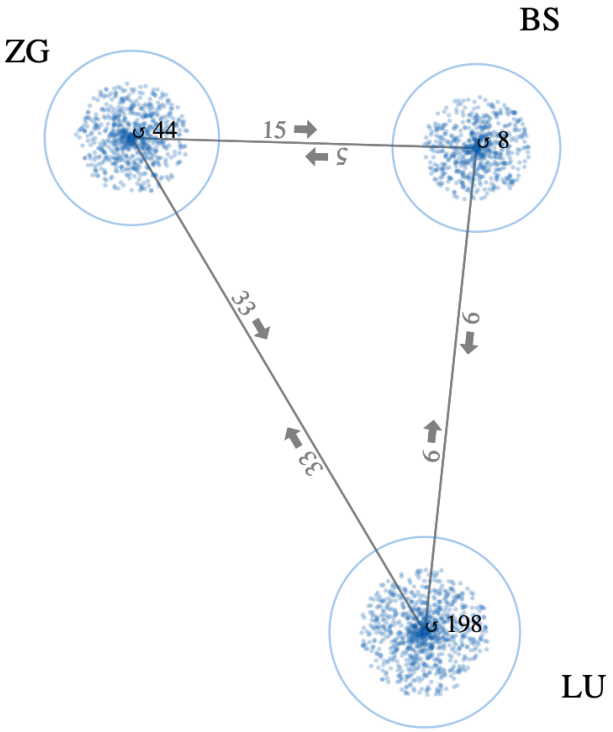

FIGURE 5.7: Exploratory Analysis: Custom Groups

The results show a clear spike in the number of active companies in Luzern during 2009 and 2010. This increase did not occur in either Zug or the control group. This trend suggests that Luzern achieved its goal with the abovementioned tax reforms.

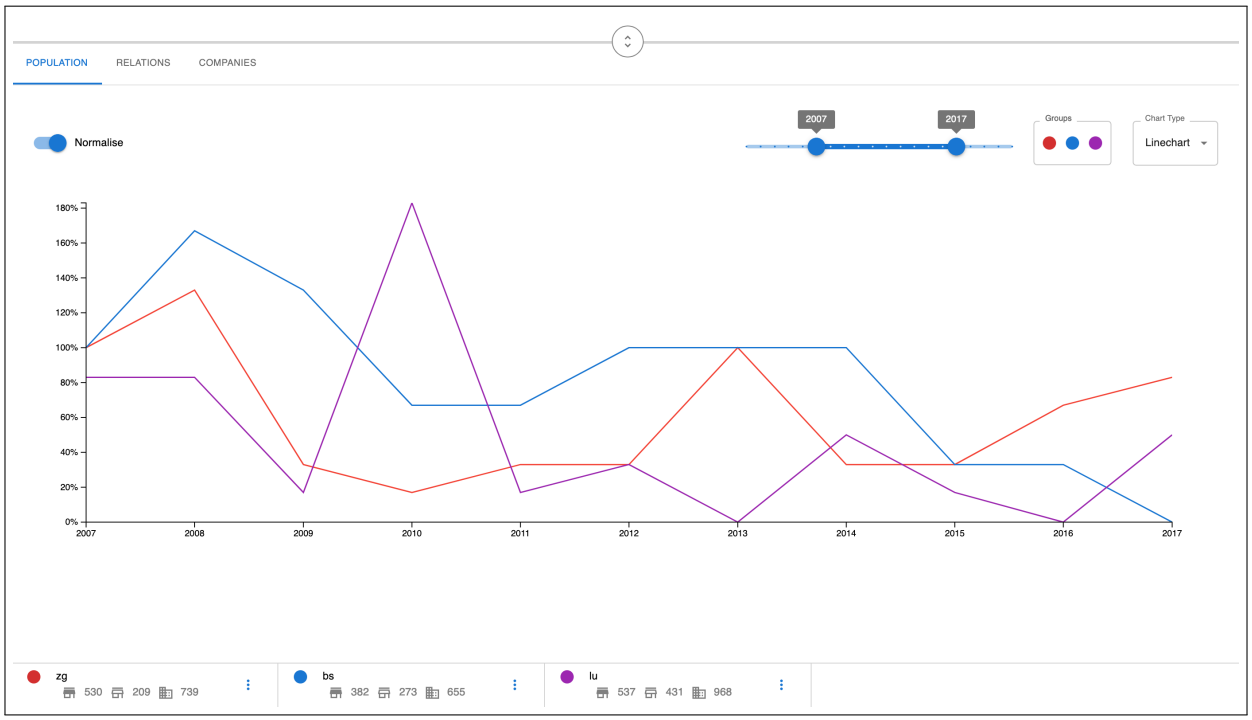

FIGURE 5.8: Evolution Analysis: Growth in Active Companies

5.3 Summary

The use cases presented in this chapter demonstrate the tool's effectiveness in supporting a range of analyses, from simple queries to more complex investigations. The tool's ability to filter, group, and visualize data enables users to explore the dataset in depth, uncovering insights that would be difficult to obtain through manual exploration alone.

The main conclusion to draw from the evaluation is that the tool's effectiveness does not rely on a single visualisation but on the combination of different visualisations that through interaction allow the user to explore and analyse the data in depth.

In the next chapter, before drawing the conclusions of the thesis, we discuss potential areas for future research and development, in particular how to address the limitations of the tool.

Chapter 6

Conclusions

6.1 Summary

The aim of this thesis is to enable large-scale spatio-temporal analysis of the data of evolving company networks exported from the Registry of Commerce that have been active between 2001 and 2021. The data contains information about both head and subsidiary companies part of a business network, including their relationships, location, name, and status as well as the evolution of these properties over time. The complexity of the data makes analysing the data a challenging task, which we made more accessible by developing a dedicated visual analytics tool.

The main activities required to reach our goal have been:

- devise and implement a custom data model
- preprocess the data to make it suitable for analysis
- define a conceptual approach to overcome the challenges of the data, and
- implement a visual analytics tool to exploit the approach

The main contributions of this thesis can be summarised as follows:

- we provide an effective way to model the data of evolving entities by applying HiSMo to graph data
- we devise an approach to explore, filter, and select the data to be analyzed, as well as several visualizations to represent the aspects of the data that are of interest for economic research
- we provide a visual analytics tool to exploit the conceptual approach we defined
- we designed interactive components to allow cross-filtering entities with time-dependent attributes

In Chapter 5, we presented an anecdotal evaluation demonstrating that, while leaving room for further develpment and refinement, the tool we developed enables the analysis of the dataset and allows to gain valuable insights as well as answering the posed questions effectively. We believe our work is in the right direction. We also believe that performing a more structured evaluation could strengthen our belief while providing even valuable feedback to improve the tool even further the future work we propose in Section 6.3.

6.2 Limitations and Threats to Validity

Considering our observations, the feedback we got from the domain expert, and the evalution we presented in Chapter 5, the biggest limitations of the tool we identified are:

- the lack of information about the sector of the companies

- the set of charts tailored to the questions that we initially defined
- the fact that we consider only control relationships and not ownership and joint management

The main threats to validity are:

- the few hundreds companies excluded in the preprocessing step
- an high-level anectotal evaluation instead of a more structured validation

6.3 Future Work

The scope of this thesis is so broad that we could not cover all the aspects we idetified during the feedback sessions and the development of the tool. Moreover, the work can be extended with many features that have always been considered out of the scope of this thesis. The following list of future work is a summary of the most important aspects we think could be addressed in the future to improve the tool and empower even better the researchers in the analysis of the data. Some of these ideas are described more in detail below.

- implement a visualisation to show the flow of company versions in each group over time,
- implement Year-over-Year normalisation in the evolution charts,
- improve sessions management,
- include foreign branches in the db,
- enrich the dimensions set,
- extend the dataset with data from the creation of the Registry of Commerce, allowing the analysis over 150 years of history,
- extend the dataset to all companies to broaden the scope of the tool from business network to the whole economic landscape,
- add data from other sources such as corporate websites, dataset about financial activity and employees of the companies,
- run a controlled experiment with a broader set of questions and two groups, the control group using the tools economists use to run similar analysis (R, Stata) and the treatment group using our tool

6.3.1 Improve Sessions Management

In Section 4.2.2 we introduced the concept of sessions, a mechanism to allow users to create checkpoints during their analysis and come back to it later. The current implementation of sessions is just a collection of sessions that can be created as copy of the state of an existing analysis. The management of the sessions can be improved by defining a hierarchy between the checkpoints, allowing users to:

- remove a whole branch of the hierarchy
- test different scenarios by creating different branches of the hierarchy

The model and the database schema are already designed to support this feature, so the implementation of this feature is just a matter of implementing the user interface to allow users to interact with the hierarchy of sessions.

6.3.2 Include Foreign Branches In The DB

We initially planned to include the foreign branches of the companies in the database but during the processing of these companies we realised that the completeness of the data related to this companies was not enough to include them in the analysis. The inclusion of the foreign branches in the analysis would allow a more comprehensive analysis of the data allowing researchers to study the behaviour of the companies in the international context.

6.3.3 Enrich the Dimensions Set

The current implementation of the tool allows users to filter the data based on the dimensions defined in the data model wich are:

- canton,
- type,
- network size, and
- tax rate

Thanks to the feedback received during the development of the tool, we identified the need to enrich the dimensions set with more advanced criteria:

- sector, and
- tax differentials

The sector dimension can be used to filter the data based on the Noga code of the companies. The noga code is a classification of the economic activities of the companies. For more information about the noga code we suggest the reader to refer to the official documentation of the Swiss Confederation [1]. The tax differentials dimension can be used to filter the data based on some criterias such as:

- delta between the average of the company tax rates and the average of the head tax rates,
- delta between average of the company tax rates and the average of the tax rates of its subsidiaries, and
- delta between average of the company tax rate and the average of the network tax rate.

Such dimensions allow a more detailed analysis of the data based on their behaviour in response to the changes in the tax rates, which is a key aspect of the analysis of the data of evolving business networks under a tax competition perspective.

6.3.4 Implement a Visualization To Show The Flow Of Company Versions Over Time

The conclusion we drew in the advanced use case described in Section 5.2 could be improved by showing how companies flow between different groups over time. To solve this we propose to implement a visualization similar to a Sankey diagram to illustrate the movement of companies. This visualization would show whether a company existed and moved to a different group as well as if a company was created or deleted at specific points in time. Such a visualization would provide a clearer understanding of the dynamics within the company networks.

[1] https://www.kubb-tool.bfs.admin.ch/en/noga/2025

6.4 Closing Words

In conclusion, this thesis has introduced an approach to the large-scale spatio-temporal analysis of evolving company networks. Through the development of a custom data model, ad-hoc preprocessing of the data, and a visual analytics tool, we have shown the viability of our approach and effectiveness of our tool. The contributions of this work establish a foundation for future research and development in this domain. We anticipate that the proposed enhancements and future work will further augment the tool's capabilities and empower researchers in their analysis of data sets whose complexity revolves around the evolution over time of the entities of interest.

Bibliography

[1] S. Ducasse, T. Gîrba, and J.-M. Favre, "Modeling software evolution by treating history as a first class entity," *Electronic Notes in Theoretical Computer Science*, vol. 127, no. 3, pp. 75–86, 2005.

[2] C.-h. Chen, W. Härdle, and A. Unwin, *Handbook of Data Visualization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[3] A. C. Dzuranin, "Data visualizations can be used to effectively communicate the results of analyses and guide decision making.," *Strategic Finance; Montvale*, vol. 103, no. 7, p. 42–49, 2022.

[4] A. Godlewska, "The idea of the map," *Ten geographic ideas that changed the world*, pp. 15–39, 1997.

[5] D. L. Hoffmann, C. D. Standish, M. García-Diez, P. B. Pettitt, J. A. Milton, J. Zilhão, J. J. Alcolea-González, P. Cantalejo-Duarte, H. Collado, R. de Balbín, M. Lorblanchet, J. Ramos-Muñoz, G.-C. Weniger, and A. W. G. Pike, "U-th dating of carbonate crusts reveals neandertal origin of iberian cave art," *Science*, vol. 359, no. 6378, pp. 912–915, 2018.

[6] M. Friendly, "A brief history of data visualization," in *Handbook of Computational Statistics: Data Visualization* (C. Chen, W. Härdle, and A. Unwin, eds.), vol. III, pp. 16–48, Heidelberg: Springer-Verlag, 2006. (In press).

[7] W. Playfair, *The commercial and political atlas: Representing, by means of stained copper-plate charts, the progress of the commerce, revenues, expenditure and debts of England during the whole of the eighteenth century*. J. Murray, 1786.

[8] W. Playfair, *Statistical Breviary; Shewing, on a Principle Entirely New, the Resources of Every State and Kingdom in Europe*. 1801.

[9] I. Spence and H. Wainer, *Introduction to Playfair's Commercial and Political Atlas and Statistical Breviary*, pp. 1–35. 01 2006.

[10] E. R. Tufte, "Envisioning information," *Optometry and Vision Science*, vol. 68, no. 4, pp. 322–324, 1991.

[11] E. R. Tufte, *The visual display of quantitative information*, vol. 2. Graphics press Cheshire, CT, 2001.

[12] E. Willis, *Laws of organization in perceptual forms. A source book of Gestalt psychology*. Routledge, 1938. original work of Max Wertheimer was published in 1923.

[13] M. Henle, *The selected papers of Wolfgang Kohler*. Liveright, 1971.

[14] K. Koffka, *Principles of Gestalt psychology*. Harcourt, Brace, 1935.

[15] S. Few, "Show me the numbers," *Analytics Pres*, 2004.

[16] S. Few, *Information Dashboard Design: Displaying data for at-a-glance monitoring*, vol. 5. Analytics Press Burlingame, CA, 2013.

[17] S. Few, *Now you see it: simple visualization techniques for quantitative analysis*. Analytics Press, 2009.

[18] D. Todorovic, "What is the origin of the gestalt principles," *Humanamente*, vol. 17, pp. 1–20, 2011.

[19] D. Holten, "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," *IEEE transactions on visualization and computer graphics*, vol. 12, pp. 741–8, 09 2006.

[20] D. Soetanto, "Examining change in entrepreneurial networks: Using visualisation as an alternative approach," *European Management Journal*, vol. 37, no. 2, p. 139–150, 2019.

[21] M. Latapy, C. Magnien, and T. Viard, *Weighted, Bipartite, or Directed Stream Graphs for the Modeling of Temporal Networks*, p. 49–64. Computational Social Sciences, Cham: Springer International Publishing, 2019.

[22] C. D. G. Linhares, J. R. Ponciano, J. G. S. Paiva, B. A. N. Travençolo, and L. E. C. Rocha, *Visualisation of Structure and Processes on Temporal Networks*, pp. 83–105. Cham: Springer International Publishing, 2023.

[23] P. Holme and J. Saramäki, *A Map of Approaches to Temporal Networks*, pp. 1–24. Cham: Springer International Publishing, 2019.

[24] J. Heer, S. K. Card, and J. A. Landay, "Prefuse: A toolkit for interactive information visualization," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 421–430, Association for Computing Machinery, 2005.

[25] A. Chopra and R. M. Samant, "Overview of visual analytics," in *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, ICWET '11, (New York, NY, USA), p. 1358, Association for Computing Machinery, 2011.

[26] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon, *Visual Analytics: Definition, Process, and Challenges*, vol. 4950, p. 154–175. Springer Berlin Heidelberg, 2008.

[27] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE Symposium on Visual Languages*, (Boulder, CO, USA), p. 336–343, IEEE Comput. Soc. Press, 1996.

[28] M. Krapf and D. Staubli, "Regional variations in corporate tax responsiveness: Evidence from switzerland," *European Economic Review*, vol. 171, p. 104891, 2025.