Bauhaus-Universität Weimar
Faculty of Media
Degree Program Human-Computer Interaction

# Towards Visualization of Relationships Between Highly Branching Hierarchies

## Master's Thesis

Erika Patricia Garcés Fernández
b. 30th August 1985 in Mexico City, Mexico

Matriculation Number 118047

First Referee: Prof. Dr. Bernd Fröhlich
Second Referee: Jun-Prof. Dr. Martin Potthast

Submission date: 18.10.2019

# Declaration of Academic Honesty

I hereby confirm that I worked on this thesis with the title **Towards Visualization of relationships between highly branching hierarchies** and did not use sources of any kind other than the ones that I have acknowledged and cited.

Furthermore, I confirm that this is the first time that this thesis - in any form - is presented to a supervising staff.

Weimar, 18.10.2019

Erika Patricia Garcés Fernández

*"Without data you're just another person with an opinion."*

**– W. Edwards Deming**

# Abstract

This thesis presents a visualization system that supports identification of content and structure text reuse. The system developed in this thesis consists of three Levels of Detail (LoD). The first LoD presents a parallel node-link diagram and an adjacency matrix, these visualizations show Wikipedia category Highly Branching Hierarchy (HBH) structure and relationships between categories based on text reuse cases. A method was implemented to generate hierarchical structures from a network consisting of the Wikipedia categories. The "Text Reuse Cases Within Wikipedia" [1] dataset was reorganized for efficient real-time access and lower latency. The text reuse cases were mapped and aggregated based on their categories to visualize them in Wikipedia HBH structure. A Transfer Function Tool was implemented, where a user can filter aggregated cases based on heuristics proposed by Alshomary [2]. The tool allows the user to create and test their own heuristic rules. The second LoD is a Parallel Coordinate plot showing the features of text reuse cases in an aggregated co-occurrence between two categories. In the last LoD, the two Wikipedia articles involved in a selected text reuse case are displayed.

# Table of Contents

# 1 Introduction

Visualizing relationships between large branching hierarchies, e.g., digital libraries, biological taxonomies and DNA, is a challenging problem, but of great importance in several fields, such as biology, taxonomy, ontology, library science, legal, societal, economic, etc. Due to both significant breadth and depth of some hierarchies (later referred to as highly branching), new visualizations are required.

Highly Branching Hierarchies (HBH) are defined in this work as deep hierarchies (e.g., 10 levels). Each node can have a large and irregular number of children. These characteristics make it a defying task to visualize.

Data exploration tasks require visualization of not only highly branching hierarchical structures, but also the complex relationships between them. This combination of different aspects of data makes it even harder to provide a proper visualization. Nevertheless, as mentioned by Spence, the user can benefit if many aspects of the data are included in the representation [3].

The aim of this thesis is to enable interactive exploration of relationships between highly branching hierarchies to support complex user tasks. This work is intended to support identification of content and structure text reuse. The case study for this work is to enable the exploration of structural and semantic similarities within Wikipedia dataset [1].

Wikipedia is organized as a collection of articles, attributed to a network of categories. The challenge of navigation within such structure is not only that articles can be attributed to multiple categories, but also that the amount of children nodes within these hierarchies can vary significantly from node to node. One extreme example is the category "Year" with 2731 children.

The simplest example of text reuse is repeating or borrowing text. The intention behind text reuse may vary between quotations, translations, paraphrases and summaries [4, 5].

Wikipedia text reuse corpus [1] shows high intrinsic diversity, e.g., a source passage may be paraphrased, summarized, or translated; one may reuse individual facts, entire paragraphs, or merely the sentence structure, which increases the complexity of analysis. To facilitate better performance of such analysis, a detailed exploration of outliers is necessary on a per-case basis, often in the context of overall structure of the corpus.

This thesis sets the goal to enable such exploration within the Text Reuse corpus [1], in order to answer the following questions within an integrated visualization environment:

- Would we expect structure reuse to be more between siblings or between parent and child? is that different for content reuse?
- Is it common to find relationships that are outside their family branch?
- Which categories have more text reuse cases?

The research questions of the thesis are:

- How to visualize relationships between highly branching hierarchies without losing context?
- How to encode and visualize the properties of such relationships?

## 1.1 Structure overview

The remaining chapters of this thesis are structured as follows:

- **Chapter 2.** This chapter provides an overview of the related work in the field of hierarchical structure visualizations, text reuse, text reuse visualization and Wikipedia visualization, where they are reviewed and discussed extensively.
- **Chapter 3.** The data used in this work is discussed in depth in this chapter, as well as cleaning, preparation and prepossessing done in it. The main technical challenges that were faced and the way they were overcome.
- **Chapter 4.** All the Visualization design decisions are reviewed and discussed in depth. The visualization design followed the Visualization Analysis Framework [6], while the interaction design was categorized following the work "Toward A Deeper Understanding Of The Role Of Interaction In Information Visualization" [7].
- **Chapter 5.** The most significant choices in the implementation are described in this chapter, such as the technologies selected for data pre-processing, database and visualizations involved in this work.
- **Chapter 6.**The main contributions and achievements of this work are summarized. Possible direction of future work is discussed

# 2 Challenges & Related Work

Wikipedia is often used in the field of information visualization as a test case due to size, complexity, and accessibility. This work also uses Wikipedia as case study. This chapter provides an overview of related work of Wikipedia and hierarchical structure visualizations, text reuse, text reuse visualization. Additionally, A Pipeline for Scalable Text Reuse Analysis with Applications to Wikipedia and the Common Crawl [2] is thoroughly discussed, the output of this work is used as the dataset to visualize in this thesis. The last section explains the main challenges for this work.

## 2.1 Hierarchical Structures

Müller et al. [8] systematically studied visualization commonly used to represent hierarchical data. For many practitioner the worth of new visualizations is unclear, and they are unsure which visualization would fit them best. This issue becomes more prominent on hierarchical data that is used largely in science and economic field. These visualizations influence highly on decisions. The study was conducted on treemaps, icicle plots and node-link representations. Radial techniques were excluded because they assumed to be less intuitable based on Burch [9].

**Node-link diagram** for expressing trees is intuitive, it replicates a botanical tree. This type of representation has been used for a long time. However, there are many representations that produce less empty space. In 1991 **Treemaps** were introduced by Johnson and Shneiderman [10]. This visualization technique uses 100 % of the available display space. It maps the full hierarchy on rectangular region filling the space. **Icicle plots** proposed by Kruskal [11] is a method to represent hierarchical clustering. One of the main benefits of this layout is that it facilitates data analysis process by making it easy to read objects that are part of the same cluster, but to appreciate this advantage it is required to have

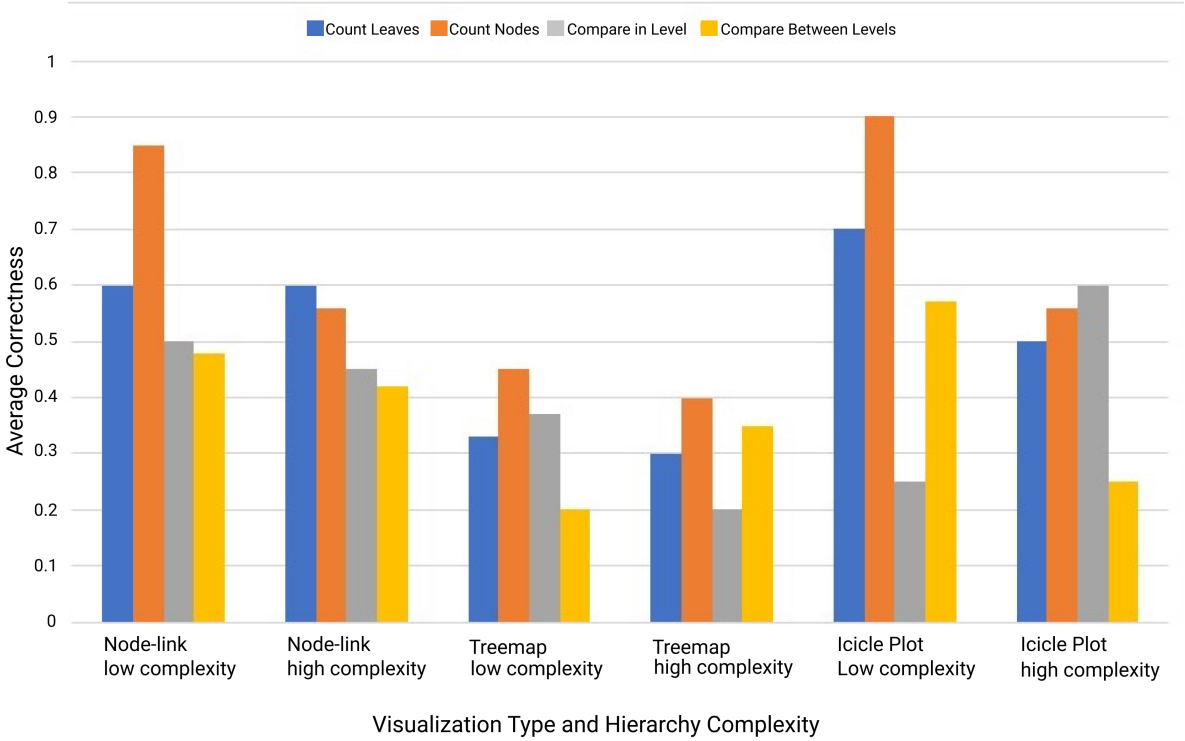


**Figure 2.1:** Plot of the average correctness of participants' answers to the tasks performed in [8] with respect of visualization and type and hierarchy complexity (Low or High)

hands on experience, especially in large hierarchies. Icicle plots combines strength from node-link diagram and treemaps visualization, intuitive top-down design and implicit hierarchy encoding [8].

The study measured user performance in terms of correctness and time, also eye movements were tracked for each participant. Four tasks were performed, where participants were asked to count all leaf nodes, all nodes of the hierarchy, compare combined area of two pairs of nodes within one level and compare combined area of two pairs of nodes across different levels of the hierarchy. The results stated that node-link diagrams and icicle plots performed as expected and in a comparable way, as shown in Figure 2.1. The performance of node-link diagrams was significantly better when using complex hierarchies.

## 2.2 Visualization of Large Hierarchies and Edge Bundling

**Concept Relationship Editor** [12] is a visualization tool intended to aid exploring and editing relations between hierarchical taxonomic classification. This tool adopts interactive space-filling adjacency layout, allowing the user to expand multiple lists with common parents. Each classification is color-coded, and the current taxa selected is colored in a lighter hue than its classification. The labels of the names of taxa are assigned enough space to be readable. The priority is given to the selected taxa, then the children, after the siblings and at the end the ancestors. If a selected list contains too many items, the user can move using 'lens mode' or 'scroll mode'. Using 'lens mode' allows the classification and all relationships visible on-screen. A fish-eye technique is implemented to allow the name of the items with less space to be legible. Scroll mode assigns enough space to the labels for all the items in the selected list to be legible. Lens mode provides context and allows for more direct comparison of relationship throughout the classifications, while the scroll mode allows to add relationships more efficiently.

**ProvenanceMatrix** [13] is a tool that allows a user to explore and understand the outcomes of taxonomic alignments. This tool uses a matrix layout and glyphs in each cell to represent articulation sets and alignments. Each side of the matrix displays a taxonomy, and it uses arcs to indicate the hierarchical information by connecting parents with direct children. ProvenanceMatrix support several interactions, such as filtering the matrix by articulations, brushing and linking; and collapsing/ expanding sub-hierarchies.

On Visualization of Large Hierarchical Data by Circle Packing [14] is described an approach to visualize large hierarchies. This visualization used the idea of nesting rectangles from Treemap, but instead of using rectangles it used circles. In the visualization, all the branches and leaves are visible. All the siblings nodes at the same level and sharing the same parent are represented by circle. The sibling circles are placed around an origin and are connected by a front-chain. Different node levels are represented by nested circles. The root parent is a big white circle, and each of its direct children has a color. This visualization has dominance over rectangle layouts because it allows very small nodes to remain visible without distorted ratios.

**Degree of Interest Tree browser (DOITree)** [15] interactively displayed large hierarchies (up to 10,000 nodes). DOITree used the Fisheye view proposed by Furnas [16] to determine

sizes for each node. The entire tree is displayed by scaling algorithm and layout. When a user clicks on a node that is not the focus of interest, the tree shrinks or grows to show the new view of the tree.

Munzner et al. proposed a system; **TreeJuxtaposer** [17], designed to support structural comparison of large trees of several hundred thousand nodes. This tool provides the user with the ability to compare and browse large trees. The TreeJuxtaposer relies on three techniques: structural comparison, guaranteed visibility and accordion tree navigation. Structural comparison is automatic detection of structural differences, where each node in one tree is associated to its most *similar* node in the other tree. The system guarantees visibility by the marked areas are always visible no matter the navigation type. The latter technique is an efficient Focus+Context navigation and layout. This system allows interaction with detailed structural comparison between trees of over 100,000 nodes each.

A visual analytical approach for comparison of multiple trees was introduced by Bremm et al. [18]. This approach was focused on global and local structures, where automatic data analysis with interactive visualization was combined. This combination is very useful for data analysis, the results of automatic analysis are used for highlighting interesting patterns in the data. Also, a new tree comparison score for pattern identification was created . A new distance measure was introduced to compare rooted trees, they claim that their measurement indicates difference in tree structures better than other measurements.

**DAViewer** [19] is an interactive visualization system for discourse analysis. This system is aimed to computational linguistics researchers to explore, compare, evaluate and annotate the results of discourse parsers. DAViewer shows an overview panel with statistics about a collection of discourse trees. It has a detail panel, where it is shown the full structure of the discourse tree, and a text panel that shows the content of the active document. Two main visualizations were from the discourse tree, an icicle plot and a dendrogram. The icicle plot is a hybrid representation, it displays the nodes in a rectangular form, but the layout imitates a dendrogram by aligning all the levels to the same rightmost level. By doing this, the user can clearly see the clustering of elementary discourse units at each intermediate stage.

Graham and Kennedy in [20] implemented a system for visualizing multiple taxonomic hierarchies by incorporating synonymy information, allowing taxonomic knowledge be

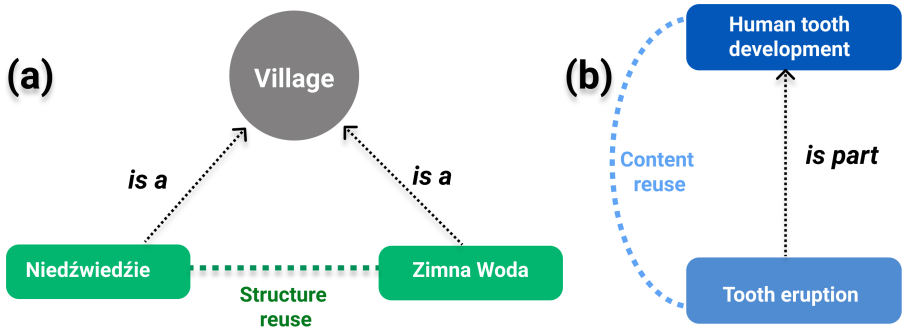


**Figure 2.2:** Ontology Tree (a)Structure text reuse often happens, when article concept are on same level in an Ontology tree. Village is a concept, Niedźwiedźie and Zimna Woda are two Polish villages, both villages are in the same level being part of the concept of Village. (b) Content text reuse can happen when an article is part of another. The example of Tooth eruption is part of the article Human tooth development.

noticed in the visualization. They also developed a navigation technique for discovering structural reorganizations between hierarchies and for revealing information about nodes not shown on the display resolution. The multiple hierarchies visualization is done by stacking trees one above the other. The taxas are represented as group of nodes below their parent node, which are placed below the root, by doing this the explicit links between parent-children are removed.

Holten in 2006 proposed a new method to visualize compound graphs, known as **Hierarchical Edge Bundling** (HEB) [21].Compounds graphs refer to relations between items and a hierarchy is defined on the items. His approach is based on visually bundling adjacency edges together. The hierarchical structure is presumed to be shown through any of the common visualization methods (treemaps, node-links, radial tree, balloon tree). Each adjacency edge is bend, modeling a B-spline. The polyline uses the path along the hierarchy from one node to another where there is an adjacency relation. The main feature of this visualization is the flexibility to use in conjunction of an existing tree visualization. It reduces visual clutter and provides an intuitive and continuous way to control the strength of bundling.

## 2.3 Text Reuse

The Internet offers wide access to text-based platforms such as Wikipedia. This facilitates the ability to reuse text, which may be unwanted (e.g., in case of plagiarism, absence of original author attribution, etc.). In the same time, there are several research contributions and tools developed towards facilitating identification, classification and further analysis of text reuse at scale. Large collections have been analyzed in e.g., electronic Text Reuse Acquisition Project (eTRAP) [4], A Pipeline for Scalable Text Reuse Analysis [2] or PicaPica[1].

A corpus of text reuse inside Wikipedia and outside the digital library (Comparing Wikipedia against other sites) was collected by Alshomary [2]. The identification of text reuse was done as pairs of document with sufficient similarity on text spans. First, the source retrieval was carried, where candidate document pairs were identified. A document d1 from a collection D1 is compared against the collection D2, this is done by ranking all the document that are member of the collection D2 in an descent order of the likelihood of having reused text from the document d1. After, these pairs were compared in detail in text alignment process. The steps involving the text alignment process were first to identify short exact matches (seed generation), the clustering of matches in groups (seed extension), and post filtering.

Alshomary found nearly 70 times more text reuse cases inside Wikipedia than outside, but involving only one third of the number of articles. Two different kinds of text reuse inside Wikipedia were identified, and ontological relationship between the topics of articles correlated with the type of text reuse were observed. Structure text reuse makes up for the majority of the cases, and it uses the same text structure with different facts. For example, explaining geographical locations on relation to its surroundings. Structure text reuse occurs more often when articles concepts are on the same level, e.g., siblings, in an ontology tree, as shown in Figure 2.2. This type of text reuse is non-problematic, specially inside Wikipedia. The articles follow certain structure and are very similar to each other, but showing the unique facts of each article [2].

The other text reuse is more questionable, articles may contain almost identical passages, as copy-paste from one to another. This type of the text reuse was considered content text reuse. When articles concepts are vertically aligned in an ontology tree, following a case

1 http://www.picapica.org

of "is a" or "part of" might exhibit content text reuse. The main issue of content text reuse within Wikipedia is that there should be one single authority source for all the section sharing the same text, by doing so the article would be consistent from one to other and reduce redundancy[2].

Alshomary proposed to classify text reuse cases using heuristics based on the percentage of reused text in the articles, Jaccard similarities between named entities, and word N-grams (2,5,10 and 15). Table 2.1 shows the proposed heuristics by Alshomary [2].

To validate the performance of these heuristics, they were applied in two samples of 100 random cases from data that was already labeled as structure text reuse case and another 100 cases where the cases were labeled as content text reuse. The validation was repeated with a sample of minimum of 200 tokens. The precision was manually computed, which was the number of cases correctly classified of the 100 instances. The results were 100% precision for structure reuse, but only 57% for content reuse.

| **Content reuse** | **Structure reuse** |
|---|---|
| H1 | H2 |
| NE sim $\in (0.5, 1.0] \wedge$ | 2 grams sim $> 0.5 \wedge$ |
| 10 grams sim $> 0.5 \wedge$ | 5 grams sim $< 0.5 \wedge$ |
| (source percent text reused $< 0.5 \vee$ | (source percent text reused $> 0.5 \vee$ |
| target percent text reused $< 0.5$) | target percent text reused $> 0.5$) |

**Table 2.1:** The heuristics use the features of Jaccard n-grams, Jaccard NE similarities, and percentage of text reuse. In order to combine the selected features, the heuristics use Boolean operators to concatenate them.

## 2.4 Wikipedia and Text Reuse within Wikipedia Visualizations

Riehmann et al. [22] proposed a visualization for similarities in Wikipedia articles. The similarities were classified in terms of structure and content. Stop word removal; eliminating the most frequently used words, and stemming; detach inflection of a word to get its root, were applied in the articles. Then vectors of the articles were created and weighted with tf-idf measure. Semantic similarity between articles was estimated using cosine similarity, where the similarity ranged from "0" to "1". The visualization shows articles as

vertices and relations as edges. If the similarity between two articles' vertices exceeded a threshold, and edge was created. Therefore, the resulting visualization consisted of isolated sub-graphs with circle layout. A rectangular placement is used to arrange the sub-graphs per row from left to right following descendant order on number of articles per sub-graph. The radial layout assigns the sub-graph with the largest number of articles in the center and order the sub-graphs in a descending order by the number of vertices, increasing the diameter per level. Edges can be seen across sub-graphs when the global similarity threshold is reduced. The user can interact with this tool by zooming, panning and dragging sub-graphs.

Castiglia's work [23] presented Wikipedia in the form of a radially arranged tree. This visualization allows the user to interact and explore the tree structure. Based on Riehmann et al. [22], he followed stop word removal, stemming process, word vector creation and weighted with tf-id measure. His work allowed the user to change the arrangement of tree by adding nodes. The user can interact directly with graphical user interface, or the content in the main window. The search bar allows the user to find specific category and the number of children to be displayed. The number of articles in a category is shown by the size of the node in the category. The user determines a new threshold using a slider. The slider sets a minimum number that must be exceeded by the similarity value between two articles to be displayed. Categories are colored in yellow if the articles comply with the new threshold value.

Biuk-Aghai and Hou Cheang designed and implemented a visualization tool using a radial layout of Wikipedia articles and categories. In order to avoid duplication, they converted the category graph into a tree. The process was done based on similarity between linked categories. The similarity calculation was based on the co-assignment in articles; in other words, the larger the number or co-occurrences the stronger the similarity. After the category similarity calculation was done, the conversion of category graph to tree took place. The graph was traversed starting from the root node to leaves nodes using breadth-first search. A list was kept with all the visited nodes, if a node was previously visited, then they kept one and eliminated the other vertex. The radial visualization represents nodes and edges. The labels for the nodes are placed outside the circle, and lines were used to connect nodes. The circle is divided into arcs that represents the top-level categories and the sub-categories are placed outside the corresponding category [24].

The visualization from Harrison [25] and **WikiGalaxy**[2] show the complexity of the Wikipedia structure. Harrison's graph layout was done using a spring model, where spring length could be adjusted. This visualization shows Wikipedia articles up to three levels in hierarchy. WikiGalaxy is a 3D visualization of Wikipedia articles as a galactic web. This visualization only shows 100,000 most popular articles of 2014.

## 2.5 Main challenges

The aim of the thesis is to facilitate interactive exploration of relationships between highly branching hierarchies. The case study for this work is the corpus of Text Reuse Cases Within Wikipedia collected by Alshomary [2].

The main data challenge for this work is the highly complex data structure, the combination of text reuse dataset and hierarchy. Part of this challenge is to build a hierarchy from Wikipedia network of categories. As shown in Harrison [25], Wikipedia structure is a highly interconnected network. However, this digital library defines main categories, which underline a tree structure underneath that could resemble a Highly Branching Hierarchy. A HBH by definition is a complex structure due to the number of levels, children and asymmetrical distribution. A tree extraction method is required, but it needs to be kept as close as possible to its original shape. Likewise, hierarchical structures have as a characteristic to encapsulate in a parent node all the information from their direct children to the last leaves on their branch. This means that methods to aggregate and propagate the data to higher nodes are compulsory.

The Text Reuse Cases corpus is almost 600 GB with more than 100 million text reuse cases. Fetching all this data in a fast way is vital for a successful visualization. Besides, this corpus has low information density, showing one single matching case within the 100s of possible reuse cases in one category. Data pre-processing, compression or other methods might be necessary for efficient on and offline access.

The main challenges for visualizing the complex structure are dealing with high-dimensional data (each text reuse case has seven features and each feature is one dimension plus two heuristics) and multiple levels of details. The layout of such a large structure and its

2 https://wiki.polyfra.me

relationships need to be considering when designing a visualization. Riehmann et al. [22] and Castiglia [23] have worked on visualizations of text reuse cases. Yet, this thesis intents to not only visualize text reuse cases, but also their relationships within Wikipedia. This means that aside from the Text Reuse Cases within Wikipedia corpus, Wikipedia structure is essential.

In conclusion, this thesis has as main challenges: complex data structure, data retrieval and visualization of the complex data structure. The first challenge refers to combining the Text Reuse Cases dataset and Wikipedia Hierarchy. Part of this challenge is to overcome Wikipedia's highly interconnected network, and to create a new structure that would be easy to follow, this structure should be kept as close as possible to the original. The second challenge is the data retrieval and processing that supports fast access to the data, and diminishes latency. The last challenges is how to visualize such a complex structure with high dimensional data in way that considers the layout of the structure and its relationships.

# 3 Data

The data selected for this work is the dataset text reuse within Wikipedia [1]. The data for the visualization required an efficient real-time access, so the original layout was reorganized. The text reuse cases were aggregated based on their categories to visualize them in a Wikipedia hierarchical structure. The aggregation was done on a local and global level. This chapter will discuss on detail the method that was implemented for consolidation of the Wikipedia hierarchical structure, the actions taken to reorganize the text reuse within Wikipedia dataset [1], and it is fully described the process for text reuse aggregation.

## 3.1 Extraction of hierarchical structure from Wikipedia category network

A tree is a network with no loops, consisting of nodes that are connected through links. The nodes are associated in parent-child relationships, with each parent relating to more than one child (one-to-many relation), but each child relating only to one parent. A tree can be seen as hierarchical structure, where there is a single node with no parent called root, and it contains all other nodes [26].

Dawkins [27] defines a hierarchy as a set where one element is superior to all other elements in the set, and there are no circular relationships. Hierarchies may be classified into:

- A **Branching** hierarchy includes at least one element that contains more than one element
- **Linear** hierarchy is not branching.

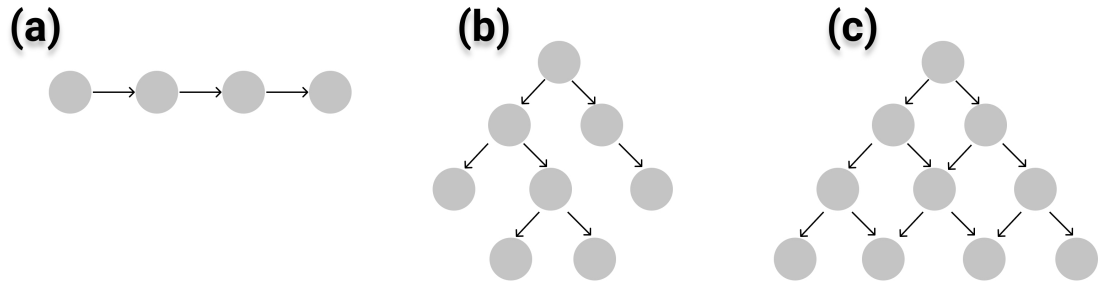


**Figure 3.1:** Hierarchical structure classification by Dawkins.(a) linear hierarchy. (b) branching hierarchy. (c) overlapping hierarchy

- **Overlapping** hierarchy includes an element that is children from more than one node.

Branching hierarchies can differ greatly from size and depth, as defined by Dawkins the only condition required is that at least one elements has more than one child. To focus on a specific type of branching hierarchy, in this work Highly Branching Hierarchy (HBH) is defined as branching hierarchy with high depth (at least 10 levels) and asymmetrical number of children per node. Good examples of HBH are digital libraries;for example, Digital public Library of America[1], Europeana[2], and taxonomic classification in biology.

Wikipedia is organized as a collection of articles, attributed to a network of categories. An article is classified into many categories, and a category is part of multiple categories, resembling a network of categories.

Alshomary et al. [28] extracted the following information of each category:

- Unique ID
- Title
- Articles belonging to this category
- Parent categories (It can have multiple parents)

1 https://dp.la
2 https://www.europeana.eu

Topologically, Wikipedia defines 27 most general categories (such as, e.g. "Academic Disciplines"). These categories accumulate distinct hierarchical clusters in their respective topological vicinity. In isolation from each other, these clusters form 27 hierarchical trees. This new resulting collection of 27 branching hierarchical structures represents a highly branching hierarchy, both in terms of depth and topological variance.

A hierarchical tree structure has a one-to-many relation. A method to generate 27 hierarchical trees from Wikipedia most general categories was implemented using the data from Alshomary et al. [28].These trees should have unique nodes to avoid loops, and each node should have only one parent.

The development of the hierarchical tree was done per level. The root node; that is called Wikipedia, is the starting point, where its children are the 27 most general categories. The next level contains the direct children of the 27 general categories. Due to the nature of Wikipedia structure, a category can have multiple parents. In order to shape the network into a hierarchical structure, only one parent was chosen. The chosen parent was the first to appear on the HBH structure. In other words; the parent closer to the root node was selected.

This HBH had a faster growth on width than depth. The tree had more than 20 levels on depth, capturing more than 800,000 categories. It had on average 2.3 children per category per level, with a variance of 87.82. The maximum number of children in a category was 2731.

## 3.2 Wikipedia Text Reuse Corpus

The result of Alshomary [2] was to consolidate a Corpus with around 100 million text reuse cases from around 380 thousand Wikipedia articles. This means that approximately 9% of the entire digital library has text reuse.

Each text reuse case had the following features:

- Unique case ID
- Articles ID
- Overlapping text reuse
- Articles title and text
- Jaccard similarity of N-grams (2,5,10 and 15)
- Named entities
- Jaccard similarity of named entities

## 3.3 Data Pre-processing

Wikipedia Text Reuse Corpus [1] is approximately 500 GB. It was stored in a Hadoop Distributed File System (HDFS) of Betaweb cluster of Web Information Systems Group and accessed through Apache Spark Engine. The data required an efficient real-time access to the text reuse data in response to user queries, which would be hindered by the highly redundant original layout.

The corpus was efficiently reorganized into three instances as shown in Figure 3.2. After the reorganization, it takes only 3.3% of the original space. Data features were fully preserved, except for "Named Entities" field, which was considered not useful for the purpose of this work. The data reorganization reduced redundancy, decreasing its size to only 20 GB.

The SQL diagram for the new structure is shown in Figure 3.3. One table stored all the features of the text reuse cases. There are instances that multiple text reuse cases exist

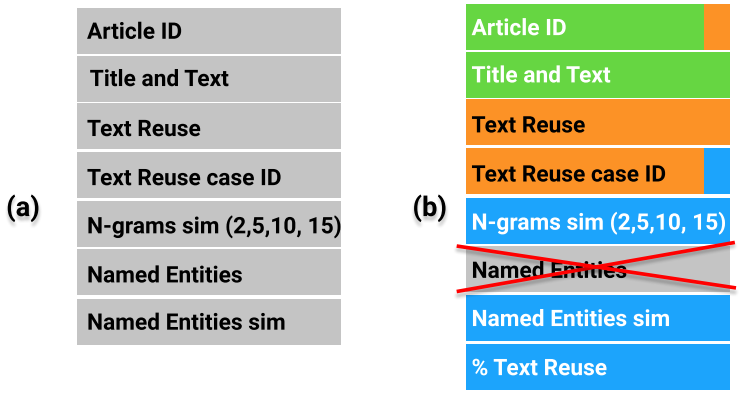


**Figure 3.2:** Wikipedia Text Reuse Corpus. (a) The original structure from A Pipeline for Scalable Text Reuse Analysis [2] (b) The new consolidated structure of the Corpus after reorganization and addition of the feature of text reuse percentage. The features were consolidated in different SQL tables. The colors represent a different SQL table, where the small rectangle to the right represents the feature used as key in the SQL table of that color. Figure 3.3 shows the SQL structure diagram of these tables.

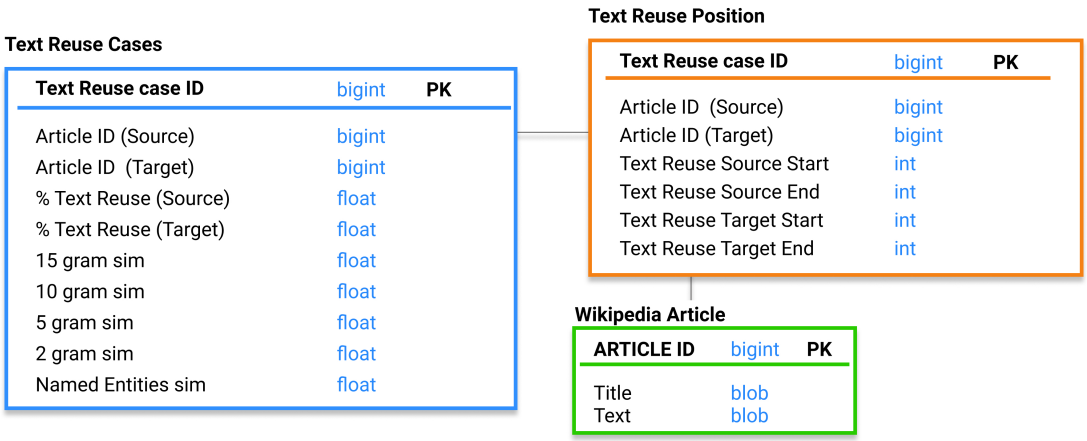


**Figure 3.3:** SQL diagram of Wikipedia Text Reuse dataset [1] after it was reorganized. The tables are color coded with the features from Figure 3.2

within the same two articles, which meant that previously the same articles were stored multiple times. Now, a separate table stored article id, text and title. The article id was used as foreign key to relate the text reuse case with their articles instead of storing whole text multiple times. Also, the overlapping text from both articles (the text reuse) was saved as the start and end positions in relation to the article, instead of saving the actual text.

## 3.4 Relationships between nodes

Conceptually, text reuse is applicable on the level of individual articles. The HBH of Wikipedia was composed of categories, as Wikipedia uses main categories to organize articles into various reference systems. In order to visualize the text reuse cases in Wikipedia's HBH structure, the articles were mapped to their classification categories. Therefore,individual text reuse cases were propagated to their categories.

An example is the text reuse case between the articles *Social Science* and *Anthropology*. Table 3.1 shows the categories of the two articles involving a text reuse case. The text reuse case is propagated using the categories instead its articles; i.e. A text reuse case is between "Social Science" category and "Humanities", and a text reuse case is between "Academic Disciplines" and "Humanities".

| Article | Categories |
|---|---|
| Social Science | • Social Sciences<br>• Academic Disciplines |
| Anthropology | • Humanities |

**Table 3.1:** Article-Category Table

### 3.4.1 Local View Structure

Text reuse cases (co-occurrences) happening in the same pair of categories were aggregated (figure 3.4). Even though, the HBH trees are the same Wikipedia structure, the relation between categories can happen in different levels of the trees.

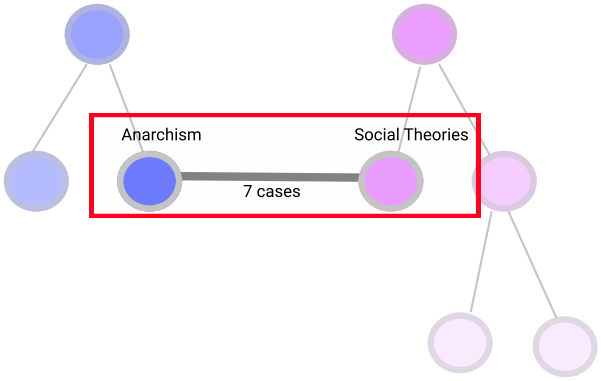


**Figure 3.4:** Local Aggregation of cases, where all text reuse cases found between the categories of "Anarchism" and "Social Theories" are aggregated. The purple node-link relation corresponds to one HBH tree, and the pink node-link relation corresponds to another HBH.

Each text reuse case was mapped to their corresponding categories, and the text reuse cases had two lists of n-categories (one list per article), that varied in length. Combinations of elements from two lists formed unique pairs, through this forming the co-occurrences between categories.

To describe the population of cases in a relationship, a normal distribution model was fit on per-feature basis (Jaccard similarity of N-grams (2,5,10 and 15), Named entities and Jaccard similarity of named entities). Classic formulas were used to describe the fit for a population with $N$ cases:

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \tag{3.1}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3.2}$$

where $\mu$ is the mean, $x_i$ is a feature value of an individual case $i$, and $\sigma$ is the standard deviation of the distribution. For example, Figure 3.4 shows the aggregation of 7 text reuse cases between the categories of "Anarchism" and "Social Science", where the mean and standard deviation were computed for each feature.

### 3.4.2 Global View Structure

A local view provides insight on the aggregation between two categories. In order to visualize this relationship, the user should know beforehand that there are text reuse cases between these categories. A global view of co-occurrences refers to the hierarchical aggregation (local level and all children categories) between two categories. This view provides overall overview of the relations happening through the hierarchies.

Given the large number of co-occurrences and categories, this task needed to be achieved in an offline fashion due to its significant complexity. The computational time required is 30 minutes using 1000 executors which each had 20 GB of memory and 2 cores. It would take around 21 days to compute the same amount of data with only one machine using 20 GB of memory and 2 cores.

Based on the already available local aggregation, the hierarchy was traversed up to the root level, while the information about local relationships was added to the upstream nodes. This process was done for both categories on the co-occurrences. Figure 3.5 shows the global aggregation process between "Social Science" and "Anarchism". "Anarchism" in this example is the last leaf on the HBH, while "Social Science" has 2 direct children and 4 in total. A Global View between "Anarchism" and "Social Science" is the aggregation of all the text reuse cases between "Anarchism" and "Social Science", and "Anarchism" and all the children from "Social Science".

A model of non-overlapping sub-population was used to describe joint distribution $X \cup Y$ as an aggregate between distributions $X$ and $Y$:

$$\mu X \cup Y = \frac{N_x \mu_x + N_y \mu_y}{N_x + N_y} \tag{3.3}$$

$$\sigma X \cup Y = \sqrt{\frac{N_x \sigma_x^2 + N_y \sigma_y^2}{N_x + N_y} + \frac{N_x N_y}{(N_x + N_y)^2}(\mu_x - \mu_y)^2} \tag{3.4}$$

where $Nx$ and $Ny$ are the counts of cases in distributions $X$ and $Y$, $\mu_x$ and $\mu_y$ are their respective means, $\sigma_x$ and $\sigma_y$ their standard deviations.

An adverse effect of this aggregation was occurrent duplication of cases. An article can correspond to multiple categories. These categories might share the same parent. The

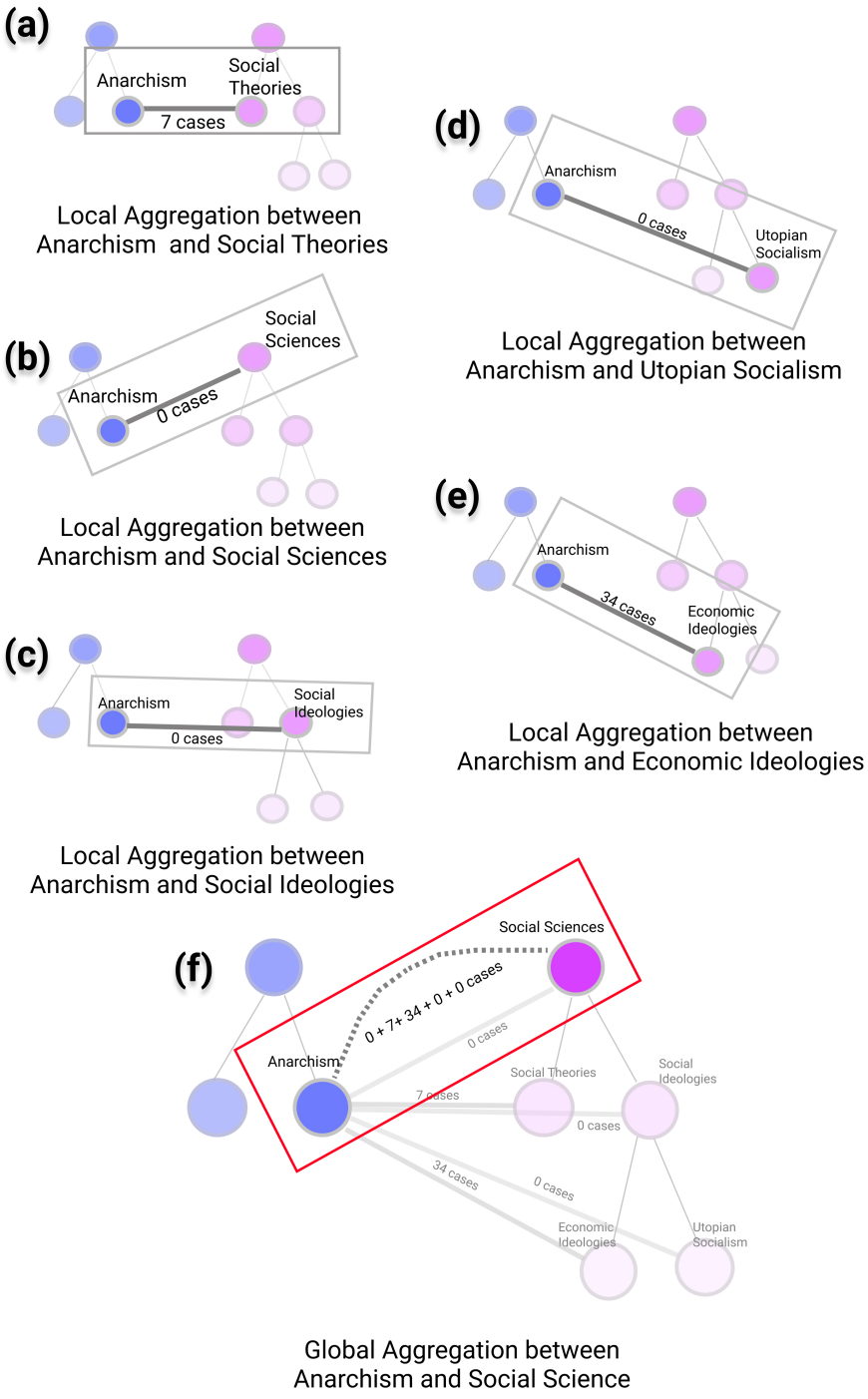


**Figure 3.5:** Example of Global Aggregation between "Anarchism" and "Social Sciences". (a) 7 text reuse cases between "Anarchism" and "Social Theories". (b), (c) and (d) show no cases between the categories. (e) presents the local aggregation of 34 text reuse cases between "Anarchism" and "Economic Ideologies". (f) The global aggregation for the relation between "Anarchism" and "Social Science" is the 7 cases between "Anarchism" and "Social Theories", the 34 cases between "Anarchism" and "Economic Ideologies"

same case could be counted multiple times, because it appeared in different children co-occurrences. When the parent aggregated cases, these were counted multiples times if the same case was shown in different children.

In order to solve this inconsistency, an index table was created (based on unique pairs of categories), where the text reuse cases that corresponded to those categories were added into a set. One of the characteristics of a set is that set elements are unique, duplicate elements are not allowed. All the cases were considered to be part of the same population; hence, the classic formulas for mean and standard deviation were used (3.1, 3.2).

## 3.5 Data summary

The text reuse Wikipedia corpus by Alshomary [1] was around 500 GB of data. The data was re-organized to efficiently access it on and offline. The organization took only 3.3% of the original space, decreasing it size to 20 GB. Only the Named Entities feature was not preserved, because it was considered not useful for this work.

A method to generate 27 hierarchical trees from Wikipedia most general categories was implemented, and they were consolidated under one root node. A highly branching hierarchy structure of more than 20 levels on depth was created, capturing more than 800,000 categories. It had on average 2.3 children per category, with a variance of 87.82.

Text reuse is applicable on the level of individual articles. Wikipedia's HBH was generated with categories, whereas the structure of Wikipedia is a network of categories. To visualize the text reuse cases in the HBH, it was need to map the individual articles to their correspondent categories. Then, the text reuse cases happening under two categories were aggregated, mean and standard deviation values were computed. The process of text reuse case aggregation was done on both local and global level. These values were pre-computed, due to its significant complexity (taking up to 21 days using only one machine).

# 4 Visualization Design

Visualization is considered to be a primary aid for data modeling and exploration. Computer aided visualization systems provide visual representations of datasets to help people perform their tasks more effectively [6]. Visualization usage can be analyzed in terms of: **what** data is shown, **why** the user needs it, and **how** the idiom is designed [6]. The same work proposes the Visualization Analysis Framework, where the designer shall answer these three core questions for each visualization. The visualization design for this work was partially based on this framework. This chapter answers these three core questions, while more information about data abstraction can be found on the previous chapter 3.

**Bough** system is our proposed tool to visualize text reuse cases. This system provides an overview of over 800, 000 categories of the Wikipedia digital library. The system lets the user explore the library, and access higher detail of information from the relationships between categories, which encapsulate the text reuse cases.

## 4.1 Views

The proposed text reuse cases visualization system has three Levels of Detail (LoD). The first LoD has two options to visualize data. (1) The HBH structures; which represent categories from Wikipedia digital library, are visualized both as a node-link diagrams with Bezier curves connecting the structures and as an adjacency matrix. This LoD provides an overview of the text reuse cases existing in Wikipedia by aggregation on cases per category. (2) The second LoD displays a parallel coordinates plot, where the cases inside a relationship are shown by features. (3) The last LoD shows the Wikipedia articles involved in a text reuse case and its reused text.

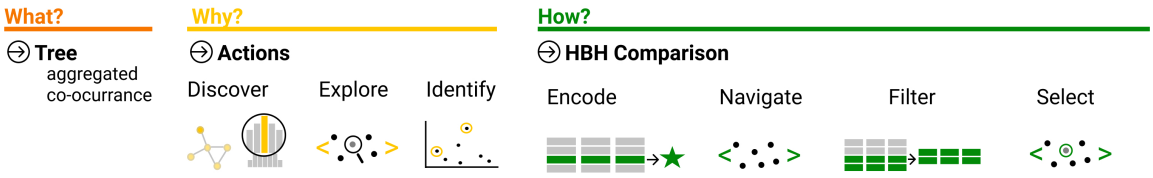


**Figure 4.1:** Analyzing what-why-how comparatively for the Level of Detail of Highly Branching Hierarchies

The HBH LoD is broken down following the "what-why-how" framework [6] in Figure 4.1. The data shown in this visualization is HBH structure of categories from Wikipedia and the aggregation of its text reuse cases. This aggregation is the collection of all text reuse cases that fall into two categories. Descriptive statistics are used to aggregate populations of values for each data feature.The aggregation is done both on the local and the global levels. In this LoD, both type of aggregation can be visualized.

The task abstraction ("Why is the task performed?") describes the user's goals. Munzner's framework [6] proposes three levels of actions, where higher level choices describe the data analysis as either for consumption or production. This visualization aims at the user's consumption of the presented data, where the goal is for the user to discover something new or analyze information. This visualization was designed for the user to discover patterns of text reuse between categories, or find rare cases of text reuse. With the help of the transfer function tool the user can generate new hypotheses for text reuse classification.

The mid-level choices are related to the type of search. In the case of HBH LoD the searches are towards an unknown location and target; hence, the search type is exploration. This type of search involves searching characteristics without regarding the location, and usually starts with an overview of everything. The first view of this visualization displays Wikipedia main categories as starting point and global aggregation for text reuse cases. This view aids the user to both discover the categories that are connected and the number of cases in each relation. This information helps the user to decide where they should continue exploring.

Once the search target has been found the next step is to query such target either by identifying, comparing or summarizing. The query for this visualization is identification [6]. The user identifies a co-occurrence that they are interested in, and it is sent to the next


**Figure 4.2:** Analyzing what-why-how comparatively for the LoD of Parallel Coordinates

LoD, where all the text reuse cases which form the aggregated collection are represented in a parallel coordinate plot.

The second LoD of the visualization is a parallel coordinate system that shows text reuse cases by their features (Jaccards n-grams, NE similarities and percentage of text reuse). This LoD displays only the cases corresponding to the chosen connection between two categories; i.e., it shows the aggregated cases that exist in a co-occurrence . Figure 4.2 broke down this view following the what-why-how approach. The data presented in this view is the feature vector per each text reuse case in the selected co-occurrence. This visualization provides the user with understanding on a per-case level of how the heuristic created on the transfer function tool describes the output shown in the HBH view. In the parallel coordinates view, the user can explore all the text reuse cases involved in the selected connection or locate a specific text reuse case. After a case has been identified, the user can click to see further details of that case. This action takes the user to next view.

The last LoD presents the text reuse in both articles. This LoD provides the user with the option to choose whether to examine the text reuse alone or inspect the text reuse in context of the full article. Figure 4.3 shows how this view follows the "what-why-how" framework. The data required for this view is the text reuse case and the articles involved in it. This view is created with the goal to lead the user to improve the heuristics for text reuse classification by studying specific cases. The sections of the articles that have been identified as text reuse are highlighted in both articles, so that the user can rapidly recognize the reuse section. The articles involved in the text reuse case are juxtaposed for easy examination. The text reuse view aggregates the data by displaying the counted words before and after the text reuse is found in the articles.


**Figure 4.3:** Analyzing what-why-how comparatively for the View of the Articles showing the Text Reuse

## 4.2 Representation

Solving specific problems means representing them in a way that searching for a solution is trivial [29]. Representation has to do with the structure, the relation between one or more items of data, as well as with the values [30].

There are several ways in which a value can be represented, so choosing the right channel to encode an element is not easy. If a wrong channel is chosen, the effectiveness is compromised. The design space of visual encoding can be described by graphical elements and visual channels to control their appearance. The same data attributes encoded with different channels offers different information to the user. The expressiveness principle advises that the information of the dataset attributes should be visually encoded. Simultaneously, the effectiveness principle recommends that the importance of the attribute should match the channel's weight [6].

### 4.2.1 HBH View Representation

Networks and trees, such as the highly branching hierarchical structures, are commonly visually encoded with node-link diagrams. The nodes are point marks and the links connecting them are lines. Another popular way to encode these structures is through adjacency matrix, where the nodes are laid out along the vertical and horizontal axis. The other usual way to encode tree structures is containment; like in treemaps. The hierarchical relationship is expressed in containment instead of lines [6].

Müller et al. [8] studied visualization commonly used to represent hierarchical data; treemaps, icicle plots and node-link representations. In their study they saw that the performance of node-link diagram was significantly better when using complex hierarchies. Hence, one of representation selected for the highly branching hierarchical data was node-link diagram. The other selected visualization for the HBH was an adjacency matrix. ProvenanceMatrix [13] shows that adjacency matrix can be an efficient way to visualize taxonomic classification. The same principles used on ProvenanceMatrix can be used to visualize Wikipedia's HBH. Each side of the matrix displays a HBH. Therefore, the second visualization for HBH was an adjacency matrix.

Hierarchical Edge Bundling (HEB) [21] represents compound graphs in a very efficient way. HEB shows the relations between items on top one hierarchical structure, and relations are view only between the leaves. One downside of this visualization is relation between inter-level is not possible. The aggregation by Wikipedia category of text reuse cases creates relation not only on the end on the HBH, but also in between levels. The best way to visualize this type of inter-level relations is to show the same structure twice. As shown in Figure 4.4 and Figure 4.5, the same Wikipedia HBH is shown twice, whether parallel to each other or along the vertical and horizontal axis.

Visualizing HBH is not an easy task, and all visualization for HBH have some drawbacks. Parallel node links might produce occlusion when the number of connection is very large, however it provides flow of a hierarchy, and it shows the links between two categories by creating a connection. Adjacency Matrix helps to reduce the occlusion, but it neither provides a flow of a hierarchy, nor does it create explicit links between categories. By having both visualizations, the user has more flexibility and better understanding of the hierarchy and the relations.

Data-ink is the non-erasable part of the design [31]. **Bough** system views were designed to be as clean as possible and to maintain the highest data-ink ratio possible. The data-ink ratio refers to the data-ink divided by the total ink used to print the graphic. No extra grids were added in any of the visualizations. On the HBH LoD on the node-link diagram, the bullet points next to the categories could have been removed, but they were kept for the interaction purposes. Beside the bullet point marker, there is no redundant data-ink on the visualization.

In the first visualization of the first LoD of the **Bough** system, two structures are shown, one on the left side and the other on the right. The two structures are connected by the

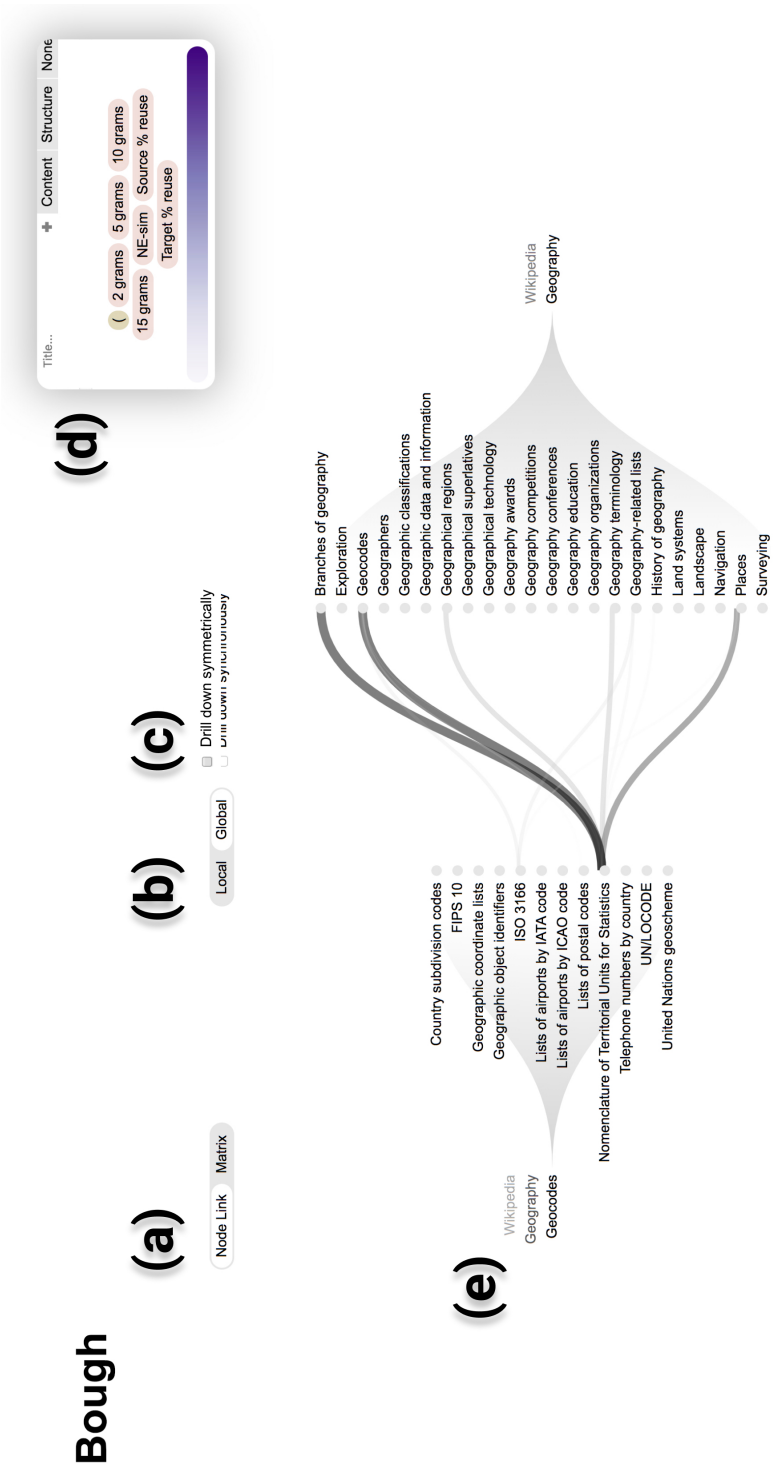


**Figure 4.4:** HBH view: (a) Toggle button to select the visualization. (b) Toggle button to select the type of aggregation on the data. (c) Checkbox to explore the HBH structures in symmetrical way. (d) Transfer function tool. (e) Visualization of the two HBH structures and the relations between categories, showing the number of the co-occurrences by the width of connections and the saturation on a gray hue.

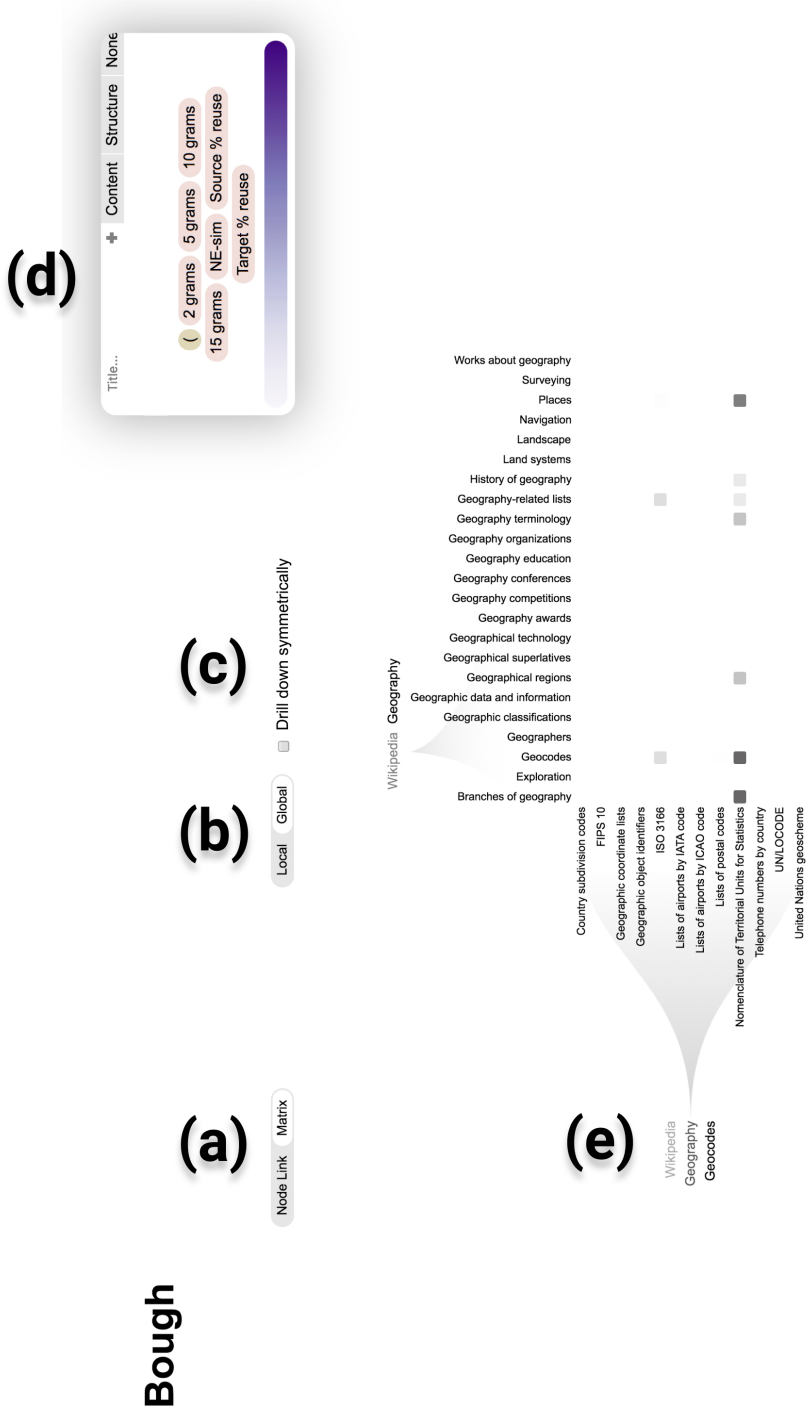


**Figure 4.5:** HBH view showing adjacency matrix visualization: (a) Toggle button to select the visualization. (b) Toggle button to select the type of aggregation on the data. (c) Checkbox to explore the HBH structures in symmetrical way. (d) Transfer function tool. (e) Adjacency matrix visualization of HBH structure and relationships between categories; the number of co-occurrences is shown as saturation of gray hue in the squares.

co-occurrences they share. For instance, the categories of "Geocodes" in HBH on the left are connected to the categories of "Geography" in HBH on the right, as shown in figure 4.4. The connection is a Bezier curve. The curve itself only expresses that there are text reuse cases between these two categories. However, the width of the curve represents the number of cases inside this co-occurrence. Width (Area 2D) is not the most effective channel to communicate, but it is the best available channel due to the arrangement of the two parallel structure facing each other.

The effectiveness of information encoding channels is quantified by accuracy. Visual channels are perceived with different accuracy, they are not equally recognizable. Humans are most likely to accurately quantify changes on length, while changes on brightness are harder to distinguish. The psychophysical power law of Stevens states that the apparent magnitude of sensory channels follows a power function in the stimulus intensity. This law shows that the length perception is entirely accurate, where area is sub-estimated, and saturation is exaggerated [6].

Color saturation was used on the connections to communicate the number of text reuse cases. It is encouraged to avoid redundancy when choosing features and channels. In this case, using only the width to express the number of cases was not enough. The reasons were that the order of magnitude on number of cases is best express on a logarithmic scale of base 10, and each category has a maximum number of pixels assigned to it without overlapping the neighboring (upper and lower) categories. Using another channel to express the same information made it easier to the user to identify the co-occurrences more accurately with a larger number of text reuse cases. Gray color was chosen because it withholds any positive or negative connotation, compared to using red or green.

Very similar to the node-link diagram, the second visualization of the HBH LoD shows the two structures. This visualization presents the structures along vertical and horizontal axis. The co-occurrences between categories are denoted by the squares drawn on coordinates where the categories intersect. For instance, the categories of "Geocodes" in HBH on the left are connected to the categories of "Geography" in HBH on the top, as shown in figure 4.5. The square expresses that there are text reuse cases between these two categories. The color saturation on this squares represents the number of cases inside this co-occurrence


**Figure 4.6:** Bread crumb from root node to "People from Lower Silesian Voivodeship". The color saturation changes as the user explores more the HBH structure.

As the user traverses the HBH structure new information is shown. The children of the selected category are displayed, while the siblings of the category are removed from the view. The traversal history is kept on the left or on the right of each HBH structure for the node-link diagram. The traversal history for the adjacency matrix is kept on the left vertical axis and on top of the matrix. The bread crumbs let the user rapidly identify where they are on the HBH structure. The color saturation on the bread crumbs lets the user know which parent on the bread crumb list is the more distant to its current position. The user drills down the tree, and as they explore the tree further, the farther the category lays from their current position the lighter it becomes. The lightest color is given to the "Wikipedia" node, and the more saturated color is given to the current selected category, as shown in figure 4.6.

A colormap denotes mapping between color and a value. Colormaps can be categorical or ordered. Ordered colormaps can be sequential or diverging. The output of the implemented transfer function is between "0" and "1"; thus, the appropriate colormap for the output of this function is an ordered sequential colormap. Usually, sequential schemes use darker colors to represent higher data values; mapping light-dark to low-high [32]. The same idea was followed when mapping the output of the transfer function to a colormap, the lower the value the lighter it is. The transfer function is extensively described later in this chapter.

### 4.2.2 Parallel Coordinate View Representation

Representing hypervariate data is an important and an omnipresent challenge, since high dimensionality is characteristic of real-world problems. One of the most common ways to represent multi-dimensional data is through parallel coordinate plots. Each dimension in the plot is a parallel axis. A single item is represented by a polyline that traverses all the axes, crossing each axis only once. Parallel coordinate plots support attribute visibility, which lets the user gain insight without a great detail of cognitive effort [30, 6]. For the second LoD in the **Bough** system a parallel coordinate plot to visualize seven text reuse case features (Jaccards ngrams, Jaccards NE similarities and Percentage of text reuse) was chosen.

On the parallel coordinate plot, all the text reuse cases that fall into the co-occurrence happening between the chosen categories are shown. If the user is exploring the HBH without any filter from the transfer tool, then the same color gray is used on the parallel coordinate plot. On the other hand, if a user is analyzing the HBH using the transfer tool, then each text reuse case is colored in a binary manner.

Figure 4.7 shows the heuristic for Structure classification defined by Alshomary in [2] and a parallel coordinate plot. The parallel coordinate shows the three cases involving "People from Wroclaw" and "People from Gmunden District". Each text reuse case is either green or red. The green color means that the text reuse case falls into the definition of the heuristic set in the transfer tool, in this case is for Structure classification. Red means that the text reuse doesn't fall into the heuristic.

## 4.3 Interaction Design

Information Visualization aims to assist the user to understand massive amounts of data that have no inherent physical, or spatial placement [26]. To visualize is to gain insight or acquire understanding, which complies with a mental model [30]. In order to create this visualization, it is necessary to relay on geometrical elements and visual channels. Just relaying on these elements and channels create static images, even though they have analytic and expressive value, their usefulness becomes limited as the data increases in size and number of variables. Therefore, it is necessary to include interaction in the

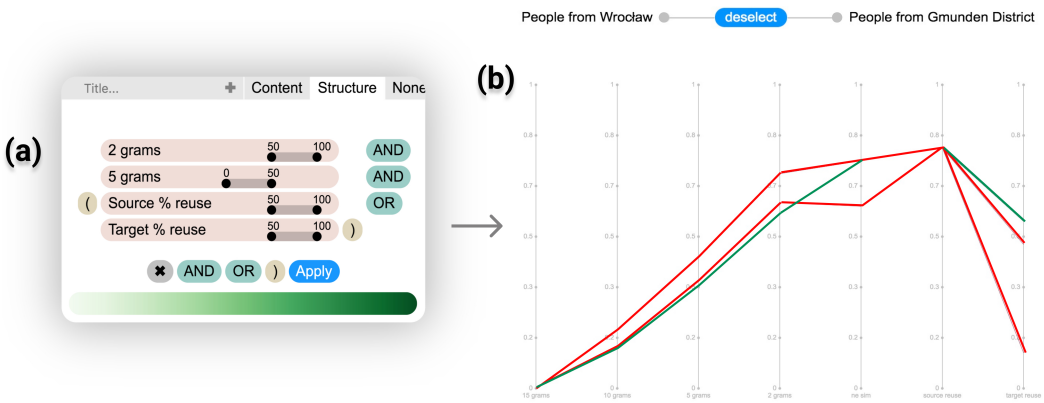


**Figure 4.7:** Parallel coordinate system: (a) Transfer function tool with the predefined structure reuse heuristic (b) three cases involving "People from Wroclaw" and "People from Gmunden District". Each text reuse case has been encoded either green or red. The green color means that the text reuse case falls into heuristic defined in the transfer tool. Red means that the text reuse doesn't fall into the heuristic.

visualization. Spence even suggests that a "passive interaction" assists the user by changing or complementing the created mental model [7].

In the work done by Yi et al.[7], interaction techniques in Information Visualization are seen as the features providing the ability to the user to manipulate and interpret representations. In their work, different visualization techniques were clustered categories based on the concept of "What a user wants to achieve" or the user's intent. This type of classification is effective because the low level interaction techniques are aggregated into more descriptive high-level categories. **Bough** system is analyzed within such framework [7].

### 4.3.1 Encode

Encoding techniques allow the user to alter visual representations of the data (color, size and shape), i.e., show different representations. These types of techniques are directly related to how the user understands both relationships and distribution of the data.

The first example in **Bough** System for encoding technique is the change between visualization on the HBH LoD. By default, the system shows the co-occurrences as Bezier curves

| Category | Interaction Technique |
|---|---|
| Encode | • Change color in co-occurrences after applying the transfer function<br>• Reset button on transfer function<br>• Change on views from Bezier curves to matrix view<br>• Content and Structure heuristics output comparison |
| Abstract/Elaborate | • Drill-down highly branching hierarchies<br>• Drill-down symmetrically<br>• Tooltip in HBH LoD<br>• Change of LoD by clicking on co-occurrences on HBH view or on text reuse cases in parallel coordinate plot. |
| Filter | • Click on the node next to a category of interest<br>• Transfer function tool<br>• Toggle button to choose Local/Global aggregation |
| Other techniques | • Undo action by clicking on Deselect<br>• Reset to view all co-occurrence in HBH view after selecting a node by using right click |

**Table 4.1:** Interaction techniques on **Bough** system.

in a parallel node-link representation. After the user clicks on the toggle of visualization types and choose Matrix, the co-occurrences changes from Bezier curves in the parallel node-link diagram to squares in an adjacency matrix.

The technique of encoding is also connected to the transfer function. The result of the transfer function is mapped to a color and an opacity value. The darker and more visible the color is, the better the co-occurrence matches the parameters chosen by the user in the transfer function. The color for structure reuse classification is green, while blues has been assigned to content reuse classification. The color purple has been selected for new heuristics defined by the user. Figure 4.8 is an example of encoding technique on the transfer function, the same HBH structure is shown in both visualizations. One visualization displays the result after applying the content reuse filter, while the other reveals the results for structure reuse filter.

The transfer function provides a separate tab, that removes all filters that were applied by the heuristics defined by the user. Figure 4.9 shows the layout of this tab. The HBH views (parallel node-link diagram and adjacency matrix) are reset to gray hue encoding the number of text reuse cases per co-occurrences.

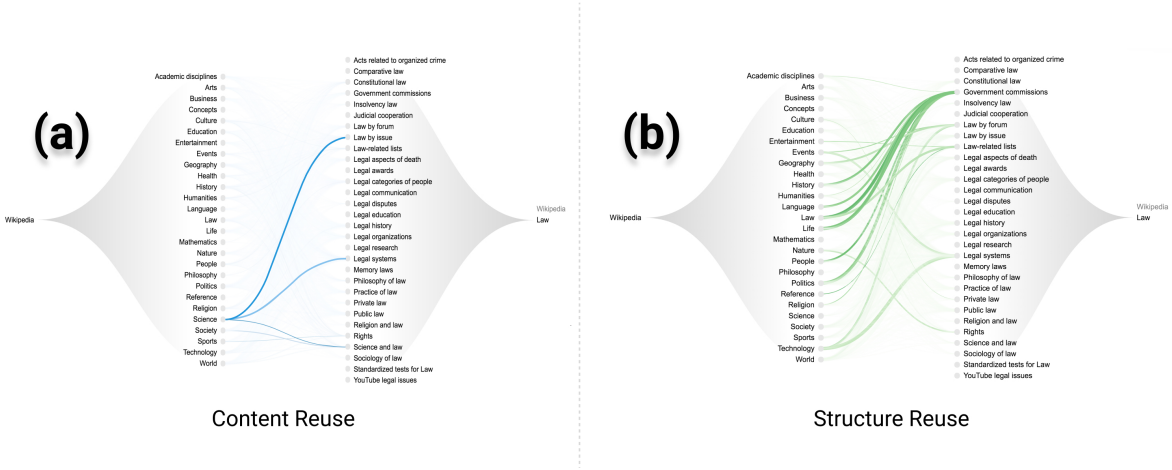


**Figure 4.8:** Examples of encoding techniques: (a) result from content reuse filter. (b) result of structure reuse filter.

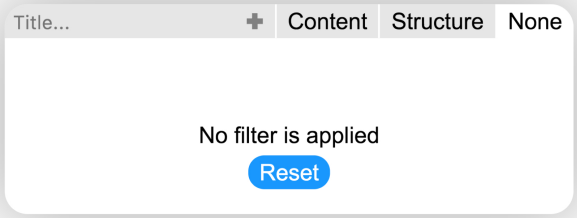


**Figure 4.9:** Tab in transfer function tool that applies no filter, reset the view to display the number of text reuse cases per co-occurrences.

The adjacency matrix has the option to compare Content and Structure heuristics in the same visualization. Instead of seeing squares where a co-occurrence is happening, two small rectangles are shown instead. This functionality is very practical when the user wants to find in which category both text reuse types are happening.

### 4.3.2 Abstract/Elaborate

The techniques under Abstract/Elaborate category provide the user with the ability to adjust the level of abstraction on the data; i.e., the user can adjust the level of detail shown.

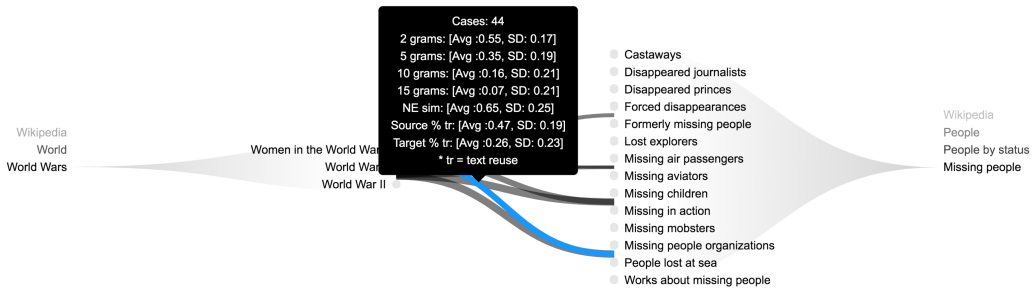


**Figure 4.10:** Tooltip is displayed on parallel node-link diagram, while highlighting in blue the corresponding co-occurrence

Drill-down operations are categorized as Abstract/Elaborate. Both of HBH appearing on **Bough** system use a drill-down operation to navigate to the children of a category. The user can navigate the HBH, while their path is kept visible as bread crumbs. The user can know right away where they are, where they can go next, or where they can return to. At the same time, while the user navigates the HBH, it allows the user to examine the co-occurrences more closely without losing the context of the structure.

The system also provides the functionality to symmetrically drill-down the tree. By enabling the checkbox on the top, the tree can be explored in a symmetric way. The user can click a category of interest, and the same category is shown as selected on the other tree. If the user wants to explore the tree asymmetrically, then they just need to uncheck the box.

A tooltip was added in HBH LoD in both views parallel node-link diagram and adjacency matrix, to the relations between the two hierarchical trees to provide extra information on the co-occurrences. The information shown is the number of cases per connection, the mean and average value computed on all the features (Jaccard n-grams, Jaccard NE similarities, and percentage of text reuse). Figure 4.10 presents a case where the tooltip is being displayed, and the corresponding co-occurrence changes its color to a bright blue.

The **Bough** system follows the Visual Information-Seeking Mantra: overview first, zoom and filter, then details on demand proposed by Shneiderman [33]. The first view in the project is an aggregated view of all text reuse cases, providing the user with an overview of the entire collection. If the user is interested in a specific category, they can zoom into it.


**Figure 4.11:** Local/ Global Toogle

The zoom provided in this project is a semantic zoom. If the user wants to access higher detail of information from the co-occurrence (Details-on-demand), then the user clicks on the bezier curve connecting the categories of interest in the parallel node-link diagram or the squares in the adjacency matrix. The result of the click is a new view, where each of the cases that were aggregated in the selected connection are shown in a parallel coordinates plot. The user can click on a specific text reuse case shown in the the parallel coordinates plot to examine it further. The actual text reuse appears under the plot, providing further details to the user.

### 4.3.3 Filter

Filter interaction technique allows the user to change the data based on a specific condition. Only the data that meets the criteria is presented. There are four interaction techniques implemented that fall in this category in **Bough** system.

The toggle button visualizes the local or global aggregation of text reuse cases. The user can decide whether to see the co-occurrences happening only within the categories shown on the screen, or to view co-occurrences happening between the categories and all their children.

The second filter interaction technique in the system is the selection of one category by clicking the bullet point next to it. The user clicks the point mark next to the desired category, and all the co-occurrences from this category are shown. The rest of the co-occurrences are removed from the view, as shown in Figure 4.12

The transfer function tool is a filter-like interaction. Figure 4.13 shows the layout of the tool. The user selects the range of interest for any of the data features, and the color below the features is used to encode the co-occurrence. The transfer function tool is further explained in the next section.

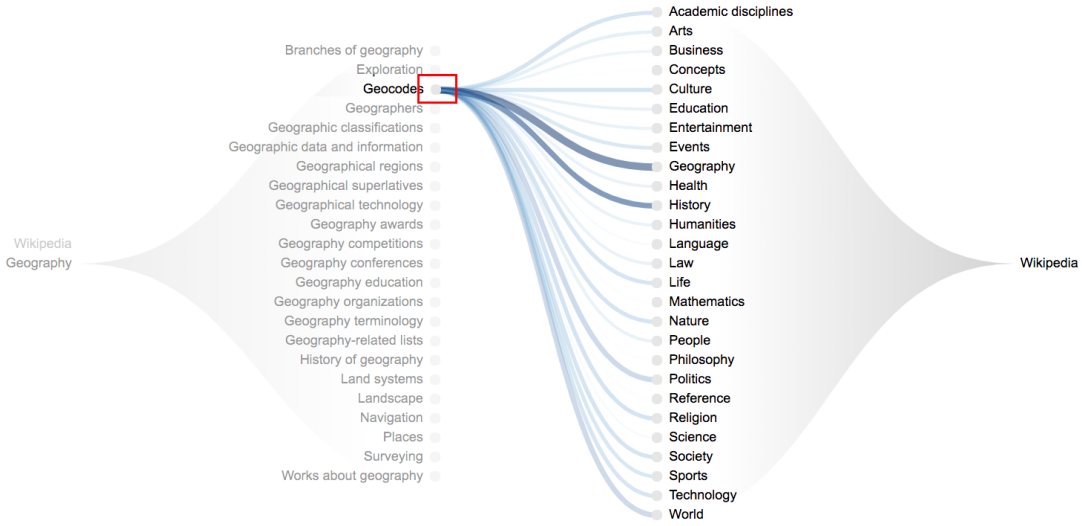


**Figure 4.12:** Filtering only category by clicking on the bullet point next to it.

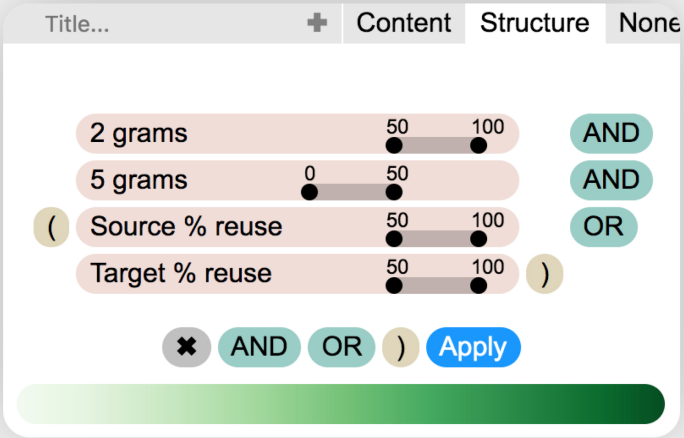


**Figure 4.13:** Structure Reuse tab in Transfer function tool. The sliders are set by default to the values proposed by Alshomary [2]. The green ramp at the bottom indicates the color of the output after the heuristic has been applied.

The last filter interaction is presented in the last LoD. The text reuse case that is presented in two Wikipedia articles can be brought into focus. The whole article can also be examined, while the text reuse section is highlighted in bright blue for easy identification. Figure 4.14 provides an example where only the text reuse case is displayed.

### 4.3.4 Other techniques

Not all interaction techniques can be categorized according to Yi et al. [7]. Some other interaction techniques available in **Bough** system are Undo (Deselect) and Reset operation.

The user selects one co-occurrence, and the parallel coordinate plot appears. The selected co-occurrence and the button "Deselect" is shown on top of the view. The user can press the button to go to the previous LoD. The same principle is implemented in the text reuse LoD. The 'Deselect' button appears on top of the compressed parallel coordinates plot. If a user wants to examine a new text reuse case in the text view, the user clicks on the button "Deselect". The button moves away from the parallel coordinates plot, and the plot is expanded.

In the HBH LoD parallel node-link visualization, filtering the co-occurrences by one category is done by clicking the bullet point next to the category of interest. Right click on any empty area of the visualization resets the node-link view, where all the co-occurrences from all the categories are shown.

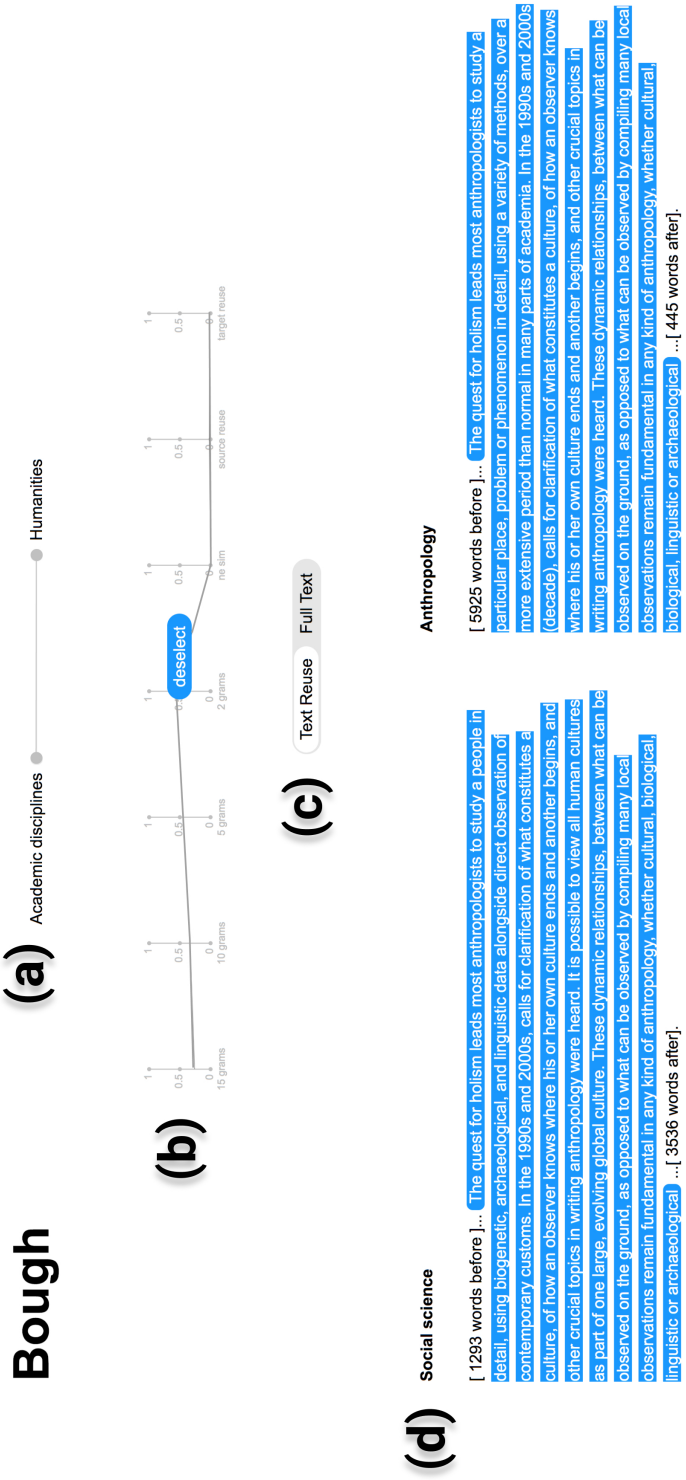


**Figure 4.14:** Text View is composed from the following elements: (a) selected co-occurrence from the HBH LoD is displayed; (b) compressed visualization of the parallel coordinates. A Button on top of the plot is available, in case the user wants to chose other cases inside the same co-occurrence; (c) toggle button allows the user to examine either only the text reuse or the full articles; (d) text is displayed, and the text reuse is highlighted in bright blue.

## 4.4 Transfer Function

Transfer function widgets are widely used in scientific visualization. These widgets allow, e.g., the user to visually query the volume rendered data, but are not limited just to volume rendering. Usually, transfer function relies on a 1D or a 2D histogram frequency plot of the data. Therefore, using a transfer function allows to alter the view and only display the data of interest [34, 35]. Transfer functions are not only used in scientific visualization, but also in Information Visualization. **Bough** allows the user to visually query co-occurrences.

In A Pipeline for Scalable Text Reuse Analysis [2] two heuristic rules for content and structure text reuse classification were proposed. Both rules had sub-optimal performance in the task of content reuse identification. Significant value can be extracted from the ability to explain these results, and to offer detailed view into both valid and invalid classifications.

**Structure Reuse** refers to two articles sharing the same text structure. For example, Polish Villages in Wikipedia, follow the same structure only changing the particular attributes of each village, e.g., name, actual location, number of inhabitants.

**Content Reuse** uses the same text. This type of reuse is usually found in articles that are part of another article; for example, *Tooth Eruption* is part of *Human Tooth Development*. Figure 2.2 provides an example of this.

There are two articles in each reuse case, and each article was mapped to its classified categories. An aggregation process was done on the text reuse cases, this process is extensively discussed in the previous chapter (3). **Bough** system applies the heuristics from Alshomary [2] on aggregated co-occurrences and not on individual cases. The heuristics are applied via a transfer function tool.

Table 2.1 shows the structure of these heuristics. These heuristics work on individual text reuse cases, and the HBH visualization displays categories and aggregation of cases.

Figure 4.15 illustrates in a simple way the idea behind the suggested transfer function. The input of the transfer function is Jaccard n-grams, Jaccard NE similarities, and percentage of text reuse as input. The output of the function is a floating-point number between "0"

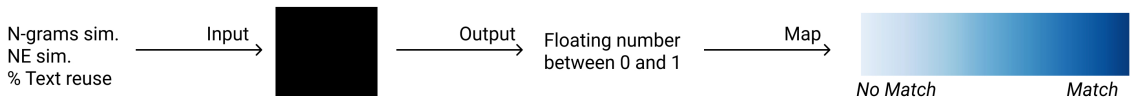


**Figure 4.15:** High level process of the proposed transfer function tool. The black box is our transfer function, where it takes as an input a combination of Jaccard Ngram, NE sim, and percentage of text reuse per articles. The output of the transfer function is mapped to a colormap.

and "1". This number is mapped to a colormap. The most saturated hue is equal to "1", and the least saturated is assigned to "0".

The user selects the features and ranges of their interest. A normal distribution is fitted per each feature. The probability of a random sample falling on the range of interest is estimated.

The classic formula was used to describe the probability density of Normal distribution:

$$f(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{4.1}$$

where $\sigma$ is the standard deviation, $\mu$ refers to the mean value of a feature and $x$ is the value of the random variable.

The terms of the heuristic are connected via Boolean operators (AND and OR). Since in our case, the result of each feature is a probability, general addition rule for not disjoint events was used.

$$P(E\ or\ F) = P(E) + P(F) - P(E\ and\ F) \tag{4.2}$$

For the connector AND the multiplication rule for independent events was followed:

$$P(E\ and\ F) = P(E) * P(F) \tag{4.3}$$

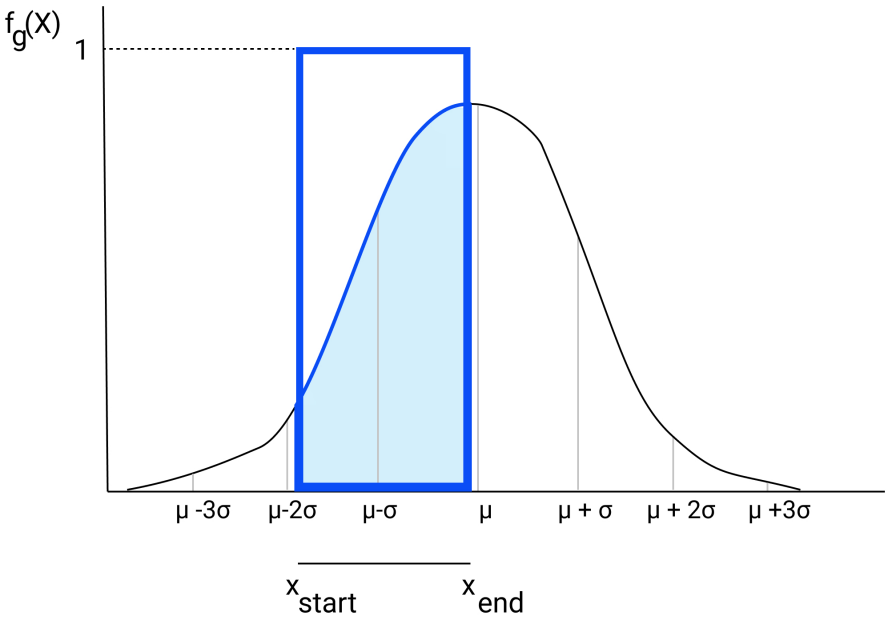


**Figure 4.16:** Example of normal distribution of one feature, and the blue square is the user's range of interest for that feature. There are two extreme cases using this computation, whether the output is "0" or "1".

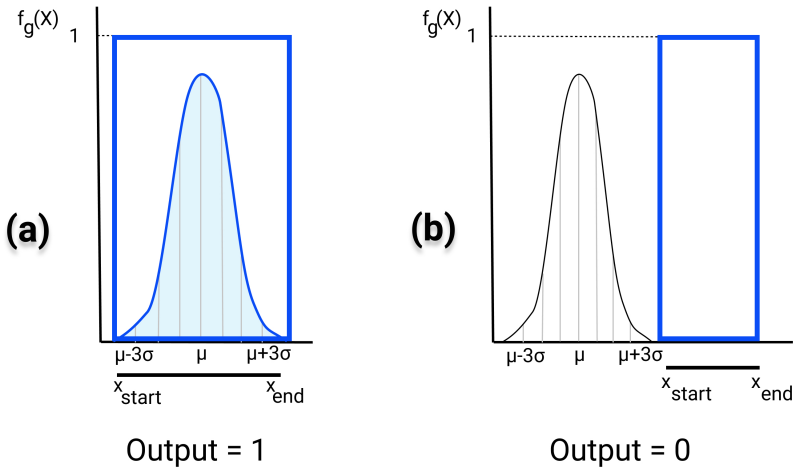


**Figure 4.17:** Output of the transfer function tool associated with extreme cases: (a) The transfer function tool outputs "1", when the range of interest captures the whole normal distribution of a feature. This case might happen when the user selects a very wide range of interest. (b) An output of "0" can be seen when the normal distribution is out of the range of interest.

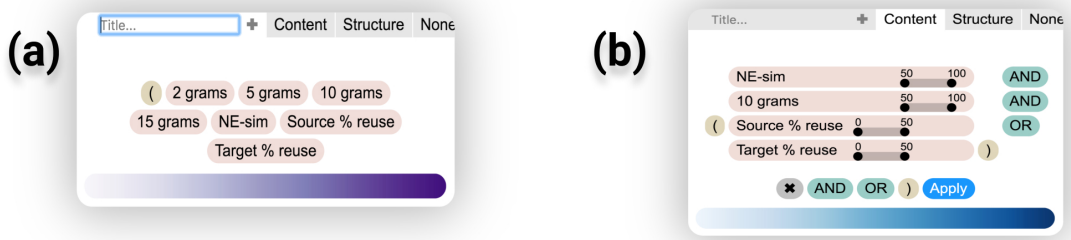


**Figure 4.18:** Transfer function tool tabs: (a) creation of user own heuristic. (b) predefined heuristic based on Alshomary [2].

### 4.4.1 Transfer Function Tool

By default the tool already has the two heuristics from Alshomary [2] and a tab to reset the HBH tree. The tool is organized as tabs, each heuristic has its own tab. When a tab is selected, all the attributes of the selected heuristic are shown. Each feature has a range slider next to it, and Boolean connectors are shown in a different color in between the features. Each of the options available on the tool is colored based on its category. The features inside the transfer function tool are colored light red, the connector are light teal, and the parenthesis are light ochre. The user can edit the predefined heuristics by changing the values on the range slider, as well as adding or removing features. The reset tab has only one button to reset the HBH view to its previous state before the transfer function was applied. The compare tab has one button to compare the outputs of structure and content heuristics in adjacency matrix visualization. Figure 4.18 shows the view of two of the tabs available in the transfer function tool.

If a user wants to create their own heuristics, it is also possible. The transfer function widget was designed to use a semi-predictive text. The first options available are parentheses and features (Jaccards ngrams, Jaccards NE similarities and Percentage of text reuse). After the user has chosen their first feature, Boolean connectors or a closing parenthesis is shown. Only when the last chosen option is a feature or a closing parenthesis is the **apply** button available. This ensures that the heuristic created by the user follows the proper structure. The user can name their heuristic and add it to the new tabs of the tool.

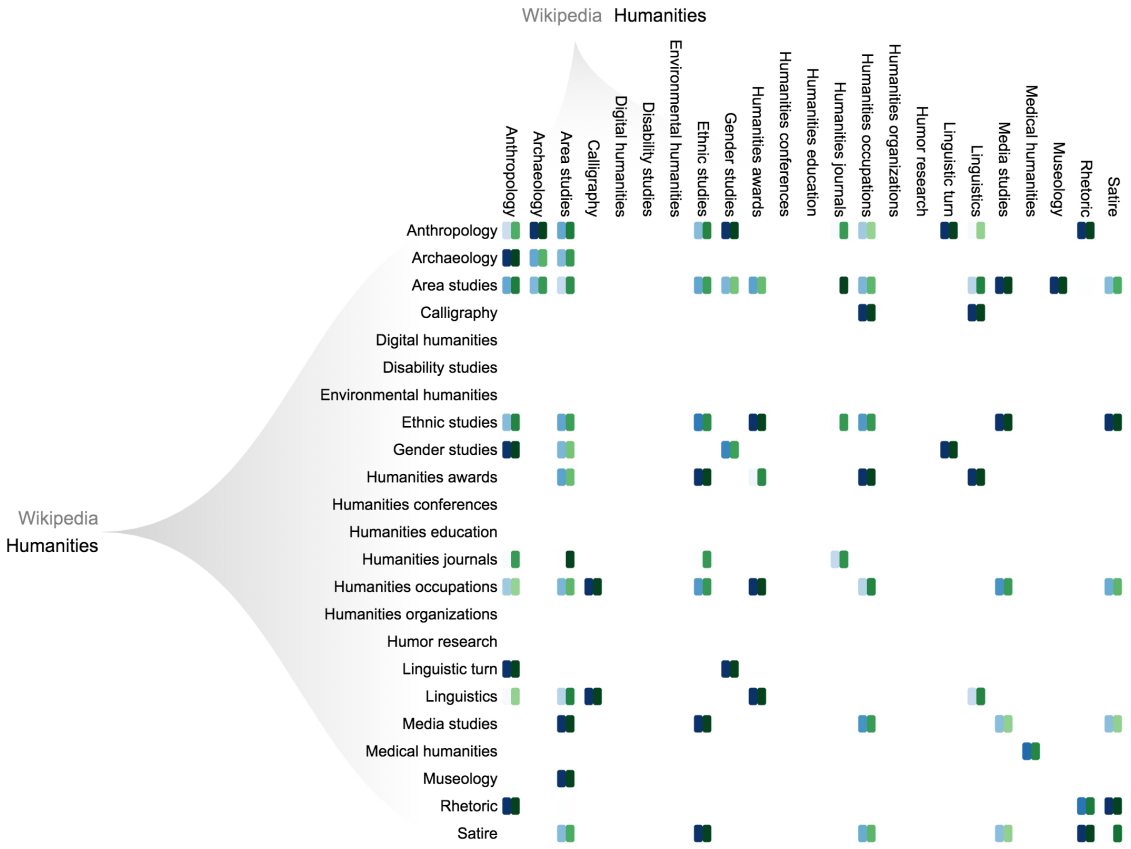


**Figure 4.19:** Comparison functionality in transfer function tool. It allows to compare content and structure text reuse heuristic tabs in one visualization.

The transfer function allows the user to apply one heuristic at a time, helping the user to focus on one type of classification. By only displaying one heuristic the user can focus on the discovery of new patterns on how text reuse is distributed inside Wikipedia structure. Currently, the two heuristics defined by Alshomary [2] exclude one another by the percentage of text reuse in articles. The user can also apply content and structure reuse heuristics at the same time by using the compare button on the transfer function, but only on the adjacency matrix. This functionality is suitable when the user wants to find which categories have both text reuse types.

## 4.5 Visualization design summary

Visualization provides a visual representation to a dataset to help humans with their tasks [6]. Visualization analysis was conducted in the framework of Munzner [6].

- ***What*** *data does a user see?* The user sees tree structures of categorical data (Wikipedia Categories) with links (co-occurrences). The links are associated to attributes with quantitative values(Jaccard similarity of N-grams (2,5,10 and 15), Jaccard similarity of named entities and percentage of text reuse ). The text reuse is a position inside the articles.
- ***Why*** *does a user intend to use a visualization tool?* The goal is to discover new insight of the data that is hard to find on other ways. The aim is to help the user to discover how text reuse is applied in Wikipedia and to refine text reuse classification.
- ***How*** *are the visual encoding and interaction idioms constructed in terms of design choices?* The visual encoding in the first LoD is based on node-link diagrams and adjacency matrix. The node-link diagram is duplicated, and placed in parallel position to create links between the two structures. In both visualization parallel node-link diagram and adjacency matrix, color was used to encode quantitative data. In the second LoD, a parallel coordinate plot was used to display the features of text reuse cases inside a co-occurrence. Color was used to encode whether a specific text reuse case falls into the applied heuristic. In the last LoD, the two articles from a selected text reuse case are juxtaposed and the text reuse is highlighted in the articles.

Interaction design was implemented with consideration of the user tasks. A user can explore the hierarchical structure by drill down; symmetrically or individually in each structure. Various types of filter functions were implemented. For example, it is possible to filter co-occurrences happening in one specific category, to filter based on the transfer function and view the co-occurrences on the global or the local level. Tooltip was added to get more details about the co-occurrence relationship; which include the number of text reuse cases per co-occurrence and the pre-computed values of mean and standard deviation.

The two main contributions of this work are the highly branching hierarchy visualization and the transfer function tool. In conjunction they support the text reuse exploration and visualization.

The HBH visualization provides a user with a good grasp of Wikipedia categories and how some categories are more prone to text reuse cases than others. At a glance a user is able to identify the co-occurrences with the largest number of text reuse cases.

The transfer function tool provides both better understanding of the corpus, and enhanced perception of text reuse classification.

The work by Alshomary [2] is a starting point to understand text reuse within Wikipedia. This digital library has a large number of text reuse cases; which constitute 9% of the entire library. This tool will help to answer questions like: (1) Is there content reuse in topics from different main classifications? (2) Is text reuse more common in parent-child relations than other? In other words, the transfer function tool allows a user to localize and qualitatively describe relationships with content reuse.

# 5 Implementation

The visualization of the Bough system was implemented in JavaScript as a client-server application, where the backend was developed using Python. The client-server communication was implemented in a web application socket protocol (WASP) environment, based on the Crossbar.io stack[1]. This chapter is divided into two section; Visualization and Data.

## 5.1 Visualization

In the beginning of the thesis, several programming tools were tested to find the most suitable one in terms of performance. D3.js was strongly contemplated for the visualizations, but higher flexibility for the development of the whole system was required, considering size of the data, and latency another framework was selected.

D3.js uses HTML, SVG, and CSS for data visualization. Following this idea, the visualization on Bough System was implemented using SVG elements. Instead of D3.js, ReactJS and Redux were adopted.

ReactJS is an open source JavaScript library, that allows to reuse UI components, which enables developers to create web applications that can change data without page reload, while being scalable, modular and simple. React community is sufficiently big, while documentation and technical support is easy to find.

Redux is an application state management framework, and a complementary library to React. Small applications with less complexity can work well without redux, but in this case it was vital to have a robust state machine to change states and update parameters as fast as possible. Figure 5.1 is a React-Redux diagram.

1 https://crossbar.io

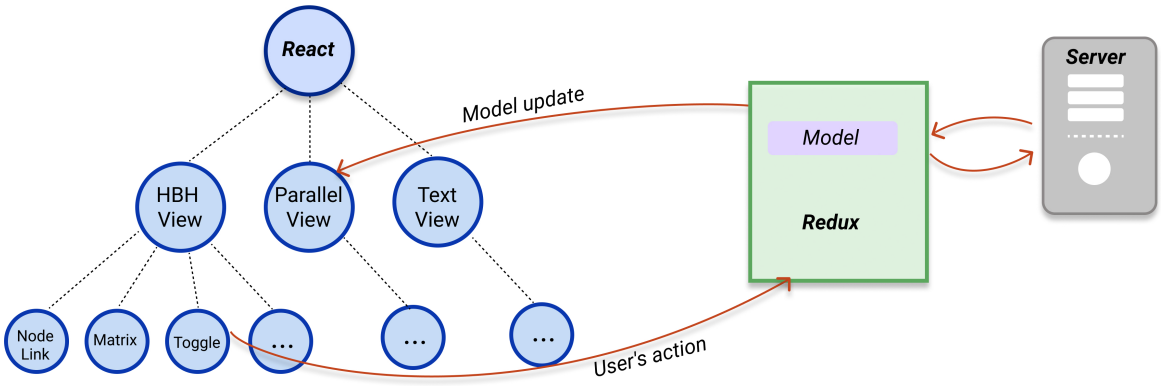


**Figure 5.1:** React-Redux diagram. React is responsible for rendering the views and all their correspondent components, and keeping the model and the view in sync. Redux keeps the states of the application in store.

## 5.2 Data

The "Text Reuse Case within Wikipedia" [1] dataset used on Bough system resided in a Hadoop Distributed File System (HDFS) of Betaweb cluster, and Wikipedia Category file stored in JSON format. The corpus was accessed through Apache Spark engine.

The data transformation; (explained in chapter 3), was done using Spark Python API (PySpark) due to the size of the data and significant complexity.

Part of the transformation of the data was to map articles to categories, and aggregate text reuse cases based on categories. The tables were saved as either Pickle files or Apache Parquet files [2]. Pickle files were used because the [1] dataset was already stored in this format, and it was also used for small files; e.g. Wikipedia name category dictionaries. Some of the advantages of using Parquet are that it stores binary data in a column-oriented way. The value of each column is organized in a way that they are all adjacent, enabling better compression.

PySQLite, a Python module, was chosen to be an SQLlite Database interface. The reasons to chose this DB were that the file format is stable, cross platform, easy to use and free to use for any purpose [3]. All heavy computations were done on the server side. For example,

2 https://parquet.apache.org/documentation/latest/
3 https://www.sqlite.org/index.html

the transfer function output values were computed in this fashion. Some of the external libraries that were used are: itertools, numpy, scipy, matplotlib and re(regular expression). The priorities in implementation choices were set to fastest task achievement and minimal latency time in UI transitions.

# 6 Conclusions

The focus of this thesis was the design of an interactive visualization tool that enables exploration of relationships between highly branching hierarchies. This tool was intended to support the task of identifying content and structure text reuse within "Text Reuse Cases Within Wikipedia" [1] dataset.

The Text Reuse corpus had a size of almost 600 GB and consolidated around 100 million text reuse cases. The corpus was fundamentally reorganized for efficient representation, real-time access and therefore lower latency. This process decreased the dataset size by 96.67% (down to 20 GB) through elimination of redundancy and irrelevant data features.

In addition to text reuse information, the category structure of Wikipedia was used. Wikipedia uses main topic categories to organize articles into diverse reference systems. A method was implemented to generate hierarchical structures from these main categories. All these tree structures were assigned as children of a root node named “Wikipedia” creating a highly branching hierarchy (HBH). This hierarchical branching structure had more than 20 levels of depth, capturing more than 800,000 categories.

To visualize text reuse cases in the HBH, individual articles were mapped to their correspondent categories. A pre-processing method for both local and global aggregation was developed. Distributions of feature vector values in text reuse cases were modelled as Normal distribution. One drawback of using averaging was the loss of fine detail; in other words, outliers were suppressed.

To support interactive visualization, "**Bough**" system was proposed and developed. This system follows the Visual Information-Seeking Mantra: overview first, zoom and filter, then details on demand [33].

The system contains three levels of detail (LoD). The first LoD has two views: a parallel node-link visualization, and an adjacency matrix. These views provide an overview of the distribution of text reuse cases between main categories by bundling text reuse cases, and assisting with visual cues for exploration and discovery. Both visualizations can be explored symmetrically and asymmetrically. When the structures are explored asymmetrically, the user can observe relationships between categories on different levels of the HBH.

After a co-occurrence has been selected in the first LoD, a semantic zoom is performed into the next LoD. All the text reuse cases which are comprised in the selected co-occurrence are shown in the parallel coordinate plot.

If a user requires more detail, they can click on a specific text reuse case in the parallel coordinate system. The two articles involved in the text reuse case are shown, along with the reused text itself. The text reuse is highlighted for rapid identification.

**Bough** system provides several interactive techniques to explore the text reuse corpus, discover new patterns and enhance the classification. The most important techniques are the Transfer Function Tool and semantic zoom through different views (Information-Seeking Mantra).

The Transfer Function Tool is a filter-like interaction. This tool allows the user to dynamically alter heuristics to filter through thousands of text reuse cases. The user selects ranges of interest for any features of the data. Based on these values, the tool returns a value between “0” and “1” that is mapped to a colormap. The colors used in this tool are predefined: blue correspond to content text reuse, green is structure text reuse, purple relates to new heuristics created by the user, and gray applies no filter returning the number of text reuse cases per co-occurrence. In addition, the Transfer Function Tool provides a comparison functionality between content and structure text reuse heuristics, which only works on the adjacency matrix visualization.

## 6.1 Future work

The next logical step for the system would be to do an expert review, and a controlled user study. An expert review will provide feedback on technical and design issues. The main focus of the controlled user study should be a comparison between parallel node-link diagram and adjacency matrix. Such study would allow to determine which visualization conveys information better, is easier to use and understand.

Future development of the **Bough** system could focus on implementing the comparison functionality in the parallel node-link visualization, in likeness to the one already available in the adjacency matrix.

Another possible direction would be to allow expanding multiple nodes with common parents as implemented on the CRE [12]. Similarly, in the adjacency matrix the function of ordering by clusters should be added.

One limitation of the current visualizations is the overwhelming quantity of text in the tooltip, which inherently represents trivially visualizable properties of feature vector value distribution (e.g., mean and standard deviation). Instead of showing the parameters of Normal distribution per feature, sliders or even small plots could be shown.

Furthermore, the text view could be enhanced by showing all the text reuse cases that happen between two specific articles. For example, the articles *Social Science* and *Anthropology* can have multiple occurrences of text reuse, and these are counted as different cases in this dataset. Also, it could be very helpful to add an extra window, where the user can see a list of articles that share text reuse with the articles of the selected text reuse case.

Enabling multi-user collaboration is another task that should be explored in future work. **Bough** system was targeting an application environment where a single user interacts with the system through a personal computer. Adapting this system for a touch-enabled display wall or other collaborative media would be beneficial for multi-user setting.

# Acknowledgements

A very special gratitude goes out to Dora Kiesel, Dr. Michael Völske, and Dr. Patrick Riehmann. Their patience, motivation, and immense knowledge made this work possible. I could not have imagined having better advisors and mentors for my thesis. Furthermore, I would like to thank Dora Kiesel for her hard effort in proofreading major parts of the thesis. I am very grateful for the frequent discussions and technical assistance from Dr. Michael Völske, which have been integral to my progress.

I would like to express my gratitude to Prof. Dr. Bernd Fröhlich and Jun-Prof. Dr. Martin Potthast for accepting my work under their supervision.

I also thank my friends that have become my family after I moved to Germany, for their continuous support and inspiration: Anton Frolov and Anton Brams. Also, I want to thank Karina Chapa that pushed me to get out of my comfort zone to chase my dreams.

My eternal cheerleaders; Fernando Garcés, Patricia Fernández and Ines Lara, thank you for encouraging me to become the better version of myself every day. I also want to thank my brother, who always provides me with the words of wisdom.

The only way to win a battle is to plan ahead and be prepared!

Thanks for all your encouragement!

# Bibliography


[1] M. Alshomary, M. Völske, T. Licht, H. Wachsmuth, B Stein, M. Hagen, and M. Potthast. Wikipedia text reuse corpus 2018, (webis-wikipedia-text-reuse-18) [data set].

[2] M. Alshomary. A pipeline for scalable text reuse analysis. Master's thesis, Bauhaus Universität Weimar, 2018.

[3] R. Spence. *Information Visualization*. Springer, 2014.

[4] G. Franzini, E. Franzini, and M. Büchler. Historical text reuse: What is it?, 2016.

[5] M. Potthast. *Technologies for reusing text from the web*. Phd thesis, Bauhaus Universität Weimar, 2011.

[6] T. Munzner. *Visualization Analysis and Design*. A K Peters visualization series. CRC Press, 2015.

[7] J.S. Yi, Y. Kang, Stasko J., and J. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE transactions on visualization and computer graphics*, page 1224– 123, November 2007.

[8] N. Müller, B. Lieblod, D. Pietschmann, P. Ohler, and P. Rosenthal. Hierarchy visualization designs and their impact on perception and problem solving strategies. *ThinkMind, ACHI 2017: The tenth International Conference on Advances in Computer-Human Interactions*, pages 93–101, March 2017.

[9] M. Burch, N. Konevtsova, J. Heinrich, M. Hoeferlin, and D. Weiskopf. Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *IEEE Trans. Vis. Comput. Graphics*, page 2440–2448, 2011.

[10] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. *IEEE Visualization*, page 284–291, 1991.

[11] J. B. Kruskal and J. M. Landwehr. Icicle plots: Better displays for hierarchical clustering. *The American Statisticians*, page 162–168, 1983.

[12] P. Craig and J. Kennedy. Concept relationship editor: a visual interface to support the assertion of synonymy relationships between taxonomic classifications. *Visualization and Data Analysis 2008, Proceedings of the SPI*, pages 680906–68091, 2008.

[13] T. Dang, N. Franz, B. Lüdascher, and A. Graeme Forbes. Provenancematrix: A visualization tool for multi-taxonomy alignments. *CEUR Workshop Proceedings*, pages 13–24, 2015.

[14] W. Wang, H. Wang, G. Dai, and H. Wang. Visualization of large hierarchical data by circle packing. *In Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 517–520, 2006.

[15] Nation D., D. Roberts, and S. Card. Browse hierarchical data with the degree of interest tree.

[16] G. Furnas. The fisheye view: A new look at structured files. *Information Visualization:Using vision to think*, pages 312–330, 1999.

[17] T. Munzner, F. Guimbretiere, S. Tasiran, L. Zhang, and Y. Zhou. Treejuxtaposer: Scalable tree comparison using focus+context with guaranteed visibility. *Proc. SIGGRAPH 2003*, pages 453–462, 2003.

[18] S. Bremm, T. von Landesberger, M. Heß, T. Schreck, P. Weil, and K. Hamacherk. Interactive visual comparison of multiple trees. *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, 2011.

[19] J. Zhao, F. Chevalier, C. Collins, and R. Balakrishnan. Facilitating discourse analysis with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics*, pages 2639–2648, 2012.

[20] M. Graham and J. Kennedy. Extending taxonomic visualisation to incorporate synonymy and structural markers. *Information Visualization (2005)*, page 206–223, 2005.

[21] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *Published in: IEEE Transactions on Visualization and Computer Graphics*, pages 741 – 748, 2006.

[22] P. Riehmann, M. Potthast, H. Gruendl, J. Kiesel, D. Jürges, G. Castiglia, B. Ter-Akopyan, and B. Froehlich. Visualizing article similarities in wikipedia. *Eurographics Proceeding. Eurographics Association.*, page 69–71, 2016.

[23] G. Castiglia. Visualisierung der kategorienstruktur der wikipedia und filterung der Ähnlichkeiten von artikeln auf basis von kategorien. bachelorarbeit, 2017.

[24] R.P. Biuk-Aghai and F.H. Hou Cheang. Wikipedia category visualization using radial layout. *The 7th International Symposium on Wikis and Open Collaboration. ACM.*, page 193–194, 2011.

[25] C. Harrison. Wikiviz: Visualizing wikipedia.

[26] A. Telea. *Data Visualization Principles and Practice*. CRC Press, 2015.

[27] R. Dawkins. *Hierarchical organisation: A candidate principle for ethology.* Growing points in ethology. Oxford, England: Cambridge U Press, 1976.

[28] M. Alshomary, M. Völske, T. Licht, H. Wachsmuth, B Stein, M. Hagen, and M. Potthast. Wikipedia text reuse: Within and without. *Advances in Information Retrieval.41st European Conference on IR Research (ECIR 2019)*, pages 747–754, 2019.

[29] H. Simon. *The Sciences of the Artificial*. MIT Press, 1996.

[30] R. Spence. *Information Visualization: Design for Interaction*. Prentice Hall, 2007.

[31] E. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 2001.

[32] C. Brewer. Color use guidelines for data representation. *Section on Statistical Graphics, American Statistical Association*, August 1999.

[33] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. *IEEE Symposium on Visual Languages*, pages 336–343, September 1996.

[34] R. Maciejewski, Y. Jang, I. Woo, H. Jänicke, K. P. Gaither, and D. S. Ebert. Abstracting attribute space for transfer function exploration and design. *IEEE TVCG 19*, page 94–107, January 2013.

[35] A. Sears and J. Jacko. *The Human-Computer Interaction Handbook*. CRC Press, 2003.

# Appendices

| Main category | # Categories | # Articles | Max. Branch | Max. Depth |
|---|---|---|---|---|
| Academic disciplines | 23183 | 418166 | 755 | 11 |
| Arts | 29852 | 531307 | 847 | 11 |
| Business | 8148 | 145461 | 198 | 10 |
| Concepts | 766 | 18899 | 22 | 8 |
| Culture | 48491 | 776165 | 494 | 13 |
| Education | 21820 | 409269 | 804 | 11 |
| Entertainment | 67336 | 947344 | 980 | 12 |
| Events | 1469 | 33301 | 45 | 8 |
| Geography | 45590 | 745429 | 234 | 14 |
| Health | 16130 | 227989 | 202 | 10 |
| History | 72636 | 792787 | 2731 | 13 |
| Humanities | 11036 | 255921 | 225 | 11 |
| Language | 2066 | 44638 | 53 | 12 |
| Law | 6552 | 99619 | 60 | 10 |
| Life | 1642 | 905393 | 23 | 10 |
| Mathematics | 1339 | 32298 | 36 | 8 |
| Nature | 36136 | 490948 | 533 | 22 |
| People | 4990 | 158316 | 238 | 9 |
| Philosophy | 285 | 5829 | 24 | 7 |
| Politics | 99450 | 1040264 | 374 | 13 |
| Reference | 1891 | 19015 | 209 | 10 |
| Religion | 18968 | 200623 | 181 | 10 |
| Science | 2513 | 56125 | 154 | 10 |
| Society | 34230 | 492117 | 226 | 12 |
| Sports | 60634 | 700825 | 553 | 12 |
| Technology | 22267 | 396004 | 528 | 12 |
| World | 204474 | 2046319 | 624 | 16 |

**Table .1:** Wikipedia's main categories statistics in the new Wikipedia Highly Branching Hierarchy structure.

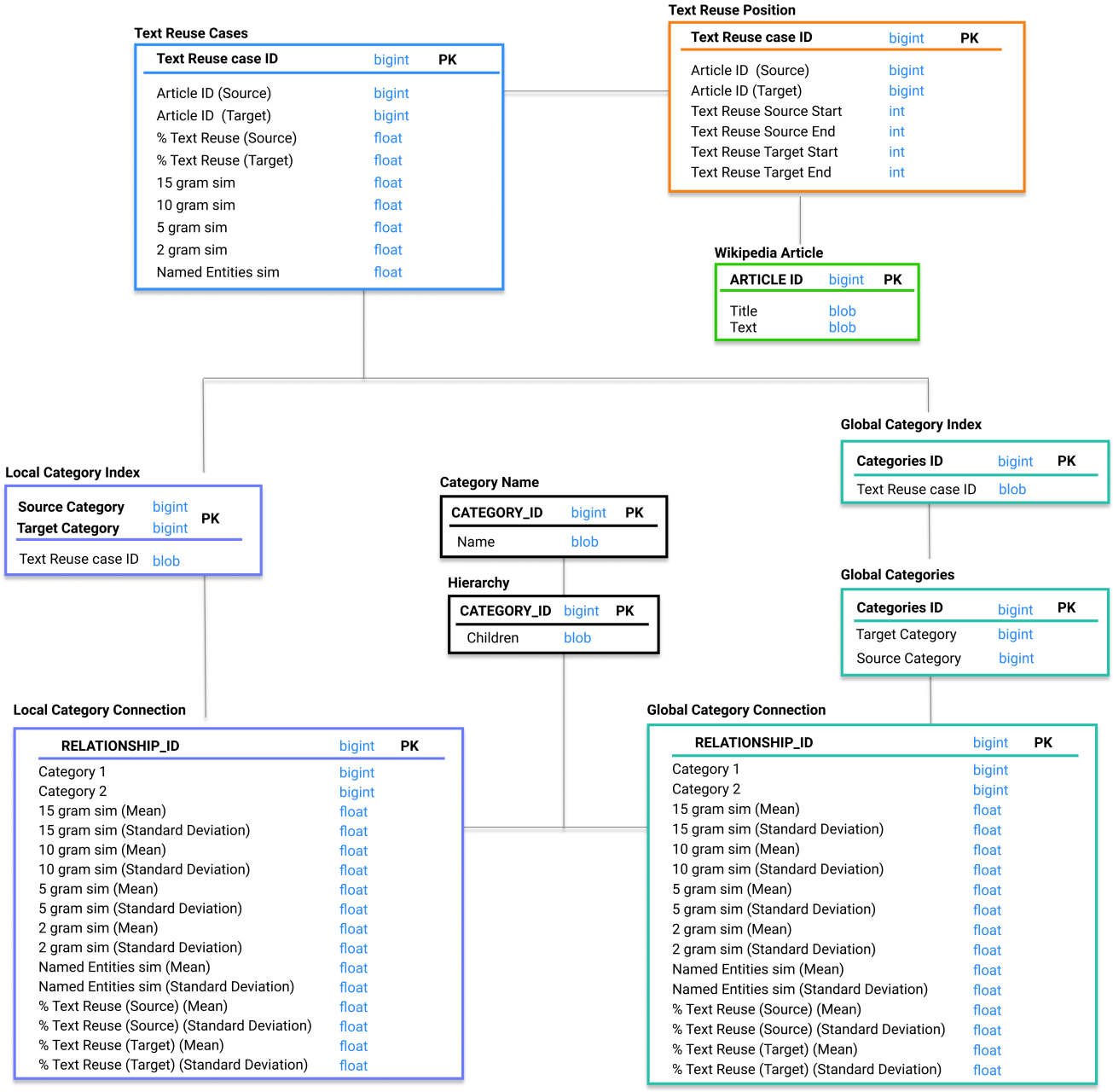


**Figure .1:** Bough SQL diagram, including Wikipedia Text Reuse Corpus, Local and Global aggregation of Text Reuse cases, Index Tables, Category Name table and Hierarchical Tree Table. The tables are color coded with the features from Figure 3.2. The local aggregation and its index is color coded in purple, while teal is used for global aggregation and its index.