Map-like Wikipedia Visualization

by

Pang Cheong Iao

Master of Science in Software Engineering

2011

Faculty of Science and Technology
University of Macau
Map-like Wikipedia Visualization

by

Pang Cheong Iao

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science in Software Engineering

Faculty of Science and Technology
University of Macau

2011

Approved by

Supervisor

Date
In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Macau, I agree that the Library and the Faculty of Science and Technology shall make its copies freely available for inspection. However, reproduction of this thesis for any purposes or by any means shall not be allowed without my written permission. Authorization is sought by contacting the author at

E-mail: inbox@patrickpang.net

Signature ______________________

Date _________________________
Wikipedia, the largest online collaboratively authored encyclopedia, allows anyone to easily contribute to its content. As it operates in a collaboratively authoring model, authors are not only responsible for writing up articles, but also assigning articles into categories in a way to classify by their topics. This makes the category system, which contains topic classifications under authors preferences is worth studying. However, given its large data volume and idiosyncrasies of its data structure, the analysis of Wikipedia’s category data is challenging. To facilitate ease and efficiency of understanding, we have designed a new visualization – a map-like visualization representing Wikipedia’s category data in a form similar to a geographic map. It allows readers to intuitively perceive large-scale patterns of categories and articles in the Wikipedia. In this thesis, we describe the algorithm of creating this novel type of visualization, and introduce our approaches for tackling the difficulties of handling the Wikipedia data.
TABLE OF CONTENTS

List of Figures ................................................................................................................. 3  
List of Tables .................................................................................................................. 5  
Glossary .......................................................................................................................... 6  
Acknowledgments........................................................................................................... 7  
Dedication ....................................................................................................................... 8  
Chapter 1: Introduction ................................................................................................... 9  
  1.1 Collaborative Authoring and Wikis .................................................................... 9  
  1.2 Wikipedia and MediaWiki ................................................................................ 11  
  1.3 Research Problems ............................................................................................ 12  
  1.4 Solution Approach ............................................................................................ 13  
  1.5 Outline of this Thesis ........................................................................................ 14  
Chapter 2: Literature Review ........................................................................................ 15  
  2.1 Information Visualization ................................................................................. 15  
  2.2 Wikipedia Data Processing ............................................................................... 19  
  2.3 Wikipedia Visualization .................................................................................... 20  
  2.4 Map-like Visualization ...................................................................................... 26  
  2.5 Force-directed Layout Algorithms .................................................................... 27  
  2.6 Overlap Removal Algorithms ........................................................................... 28  
Chapter 3: Map-like Visualization ................................................................................ 29  
  3.1 Overview of the Visualization .......................................................................... 29  
  3.2 Data Source Requirements ................................................................................ 30  
  3.3 Sorting Algorithm by Similarities ..................................................................... 31  
  3.4 Preliminary Layout ............................................................................................ 32  
  3.5 Hexagonal Visualization ..................................................................................... 35  
    3.5.1 The Hexagon Canvas ............................................................................... 35  
    3.5.2 Selection of Hexagons ............................................................................. 37  
    3.5.3 Clutter Reduction ..................................................................................... 38  
    3.5.4 Coloring Scheme ...................................................................................... 40
<table>
<thead>
<tr>
<th>Number</th>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1: Treemap visualization of a folder in a file system (reproduced from [10])</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Figure 2: A treemap with one million of items (reproduced from [11])</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Figure 3: Two representations of a tree: (a) the radial layout (b) the balloon layout (reproduced from [13])</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Figure 4: Connections added to: (a) a radial tree (b) a balloon tree; both visualizing function calls in a software (caller in green and callee in red) (excerpt from [14])</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>Figure 5: The screenshot of SpaceTree (reproduced from [17])</td>
<td>18</td>
<td></td>
</tr>
<tr>
<td>Figure 6: Haber and McNabb’s visualization model</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>Figure 7: Five levels deep, centered on Politics</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Figure 8: Close up of Chris Harrison’s WikiViz output</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Figure 9: Entity view with labeled nodes in WikiVis (reproduced from [23])</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Figure 10: WikiVis category view (reproduced from [23])</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Figure 11: Wikipedia category visualization (reproduced from [7])</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Figure 12: The Chromograms application: showing users’ activity in blocks (reproduced from [24])</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Figure 13: History flow visualization (reproduced from IBM Research)</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Figure 14: Author text analysis of a section of text with high depth of collaboration (reproduced from [26])</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Figure 15: Edit significance bars of revisions of an article (reproduced from [27])</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>Figure 16: Summary of different type of edit behaviors (reproduced from [27])</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Figure 17: Visualization of a 10-year period by cartographic means (reproduced from [28])</td>
<td>27</td>
<td></td>
</tr>
<tr>
<td>Figure 18: Comparison of overlap removal algorithms (reproduced from [32])</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>Figure 19: The bottom-up layout approach</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Figure 20: Approaches for radial placement of sorted entities</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Figure 21: Hexagons and their coordinates in the memory</td>
<td>35</td>
<td></td>
</tr>
</tbody>
</table>
Figure 22: Arithmetic of a hexagon ................................................................. 36
Figure 23: Example of hexagon selection ...................................................... 37
Figure 24: Before and after clutter reduction ................................................... 39
Figure 25: A topographic map uses colors for showing elevation .................. 40
Figure 26: Legend of a map-like visualization ................................................ 40
Figure 27: Rotation of text labels in a region ................................................... 41
Figure 28: Result of the text label placement .................................................. 42
Figure 29: Eliminating edges in the graph (edge indicates “parent-to-child”
    relationship) ............................................................................................. 50
Figure 30: Comparison of different experimental methods for aggregating
    similarity ..................................................................................................... 52
Figure 31: The current way to obtain category distribution information .......... 58
Figure 32: Highlight of category “Mathematics” in the Wikipedia ................. 58
Figure 33: Cluster of related categories “Environment”, “Life” and “Geography”
    placed in close proximity of one another ............................................... 59
Figure 34: Category “Science” in Wikipedia: (a) Danish (b) Swedish (c) Chinese 62
Figure 35: Highlight of Holloway’s Wikipedia visualization ......................... 63
Figure 36: A multi-level matrix visualization of category hierarchies ............. 64
Figure 37: A radial visualization of Wikipedia (reproduced from [37]) ....... 66
Figure 38: Extract of the radial visualization near the edge of the circle
    (reproduced from [37]) ........................................................................... 67
Figure 39: Overview map of the Simple English Wikipedia ......................... 78
Figure 40: Overview map of the Danish Wikipedia ...................................... 79
Figure 41: Overview map of the Chinese Wikipedia .................................... 79
Figure 42: Overview map of the Swedish Wikipedia ................................... 80
Figure 43: Overview map of the German Wikipedia .................................. 81
Figure 44: Overview map of the English Wikipedia ................................... 82
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Number</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1: Comparison of wikitext and HTML</td>
<td>11</td>
</tr>
<tr>
<td>Table 2: Examples of hierarchical data (number in the parentheses represents the size)</td>
<td>30</td>
</tr>
<tr>
<td>Table 3: Sorting algorithm with similarity pairs</td>
<td>32</td>
</tr>
<tr>
<td>Table 4: Algorithm for randomly selecting hexagons</td>
<td>38</td>
</tr>
<tr>
<td>Table 5: Namespaces in MediaWiki</td>
<td>44</td>
</tr>
<tr>
<td>Table 6: Definition of the “categorylinks” table</td>
<td>44</td>
</tr>
<tr>
<td>Table 7: Sizes of Wikipedias</td>
<td>45</td>
</tr>
<tr>
<td>Table 8: Top content categories in different Wikipedia language editions</td>
<td>46</td>
</tr>
<tr>
<td>Table 9: Comparison of name similarities under category “Aircraft 1950-1959” and “Computer science”</td>
<td>49</td>
</tr>
<tr>
<td>Table 10: Cosine similarity for categories in the English Wikipedia with data from different levels of sub-categories</td>
<td>51</td>
</tr>
<tr>
<td>Table 11: Standard deviations of similarity aggregation experimental expressions</td>
<td>53</td>
</tr>
<tr>
<td>Table 12: Aggregated similarity of category “Mathematics” in the Wikipedia</td>
<td>54</td>
</tr>
<tr>
<td>Table 13: Statistics of visualization output</td>
<td>56</td>
</tr>
<tr>
<td>Table 14: Statistics of the German and English Wikipedia</td>
<td>60</td>
</tr>
<tr>
<td>Table 15: Statistics of category “Science” in the Danish, Swedish and Chinese Wikipedia</td>
<td>62</td>
</tr>
<tr>
<td>Table 16: Responses of the user evaluation</td>
<td>68</td>
</tr>
<tr>
<td>Table 17: SQL commands for importing database dumps</td>
<td>83</td>
</tr>
</tbody>
</table>
GLOSSARY

**Article**: An encyclopedia article in the Wikipedia.

**Category**: A classification defined in the Wikipedia that groups articles with relevant information.

**Category Assignment**: An article is assigned to a category, and becomes part of the content of that category.

**Category Relationship**: A measurement of relatedness of categories. If contents within one category are more similar to the contents of another category, then we define they have a closer relationship or vice versa.

**Co-assignment of Categories**: If an article is assigned with multiple categories at the same time, the article is co-assigned with these categories.

**Cosine Similarity**: Cosine similarity is a measure of similarity between two vectors by measuring the cosine of the angle between them. It is widely used in measuring similarities of documents and strings.

**Directed Acyclic Graph**: A directed acyclic graph is a directed graph with no directed cycles. For example: for any starting point of vertex \( v \), there is no way to follow a sequence of edges that eventually loops back to \( v \) again.

**Namespace**: Namespaces are used in the Wikipedia to separate contents for different purposes, such as articles and categories are stored under Main and Category namespace respectively for public access, while Talk namespace consists of discussions that intended for the authoring community. In this way, different types of users are able to concentrate on the data which intended for their use.

**Page**: The basic element in the Wikipedia, which can be an encyclopedia article, a category or other object that differentiate by its namespace. A page is uniquely identified by its page ID.

**Sub-category**: A category that is assigned under an upper level (parent) category. Usually sub-categories are defined as more detailed topics of their parent’s topic.

**Top Categories**: The highest level of categories for classifying articles by their topics, such as Science, People, History, etc.
ACKNOWLEDGMENTS

I would like to thank my thesis supervisor, Dr. Robert P. Biuk-Aghai, who kept guiding and encouraging me along the way of this research. He inspired me with different new ideas, points to improve and possible solutions to problems. Without his leadership and experience, it is impossible for me to complete this thesis. Moreover, he paid so much effort in helping me improve the academic writing skills as well as publish publications. I truly appreciate his effort and help during these three years of supervision.

I am also grateful to my colleagues in the University of Macau, especially my unit head Steve Lai, team leaders Jack Wu and Christina Lou and my project manager Evi Ian, who helped me in my work and coordinated the workload so that I could successfully finish this degree.

I would also thank the former and current students of the Faculty of Science and Technology, such as Libby Tang, Henry Leong, Alex Leong, Kin Lei and Peter Fong, for sharing their experiences and knowledge related to this research.

I gratefully acknowledge the support for the publications of this research from the Macau Special Administrative Region – Science and Technology Development Fund (grant number 021/2011/A).

Last but not least, I will never forget the support and encouragement from my parents and all my friends on both my research and my life.
I would like to dedicate this thesis to my parents, who continuously support my studies and my life all the time.
CHAPTER 1: INTRODUCTION

Wikipedia\(^1\), a familiar word that surrounds in our lives recently, is known as a large and free encyclopedia online. Technically speaking, Wikipedia borrows the concept of a *wiki*. A wiki is a website which invites people to create and modify its content page, using simple markup language, and with an editing tool that is compatible with a plain web browser. Another characteristic for a wiki is inter-page hyperlinks, which display topic association between different pages [1]. In recent years wikis have witnessed increasing adoption in many application domains. The most well-known wiki is the free online encyclopedia Wikipedia, however there are also numerous other public as well as intra-organizational wikis [2]. Wikis are collaboration systems that are by design open and intended for collaborative editing. Usually any user is allowed to publish content in wiki articles. Authors can also create categories and assign articles to them in order to better organize content. All these editing and categorizing operations are performed manually by the wiki’s community of users. As a result a wiki is the product of large-scale human collaboration. The content and the structure of the outcome of this human collaboration attract many researchers to study. However, with the large quantity of data and the complex data structure, the progress of data analysis of Wikipedia faces many challenges. To address this issue, as well as provide a tool for wiki researchers to use with, we introduce a visualization method that produces a virtual “world map” of Wikipedia, providing an overview of its contents in the form of a geographical map. The proposed visualization provides an easily readable and understandable overview of the content in the entire Wikipedia.

1.1 COLLABORATIVE AUTHORING AND WIKIS

Collaborative authoring enables multiple authors to contribute written content to a common work through computer network. Over time people use various approaches to

\(^{1}\) [http://www.wikipedia.org/]
accomplish this purpose, for example, editing a file on a shared server and exchange multiple versions of a file via e-mail. Needless to say these methods are error-prone and inefficient. Collaborative software is introduced for improving this scenario and the first wiki system – WikiWikiWeb was developed by Cunningham [3]. With the emergence of Web 2.0, many business and organization started to use wikis as their internal collaboration tools. In the meantime, a number of wiki software was developed to support different technologies and platforms. Confluence (Java)², MediaWiki (PHP)³ and MoinMoin (Python)⁴ are some examples of popular wiki software.

The concept of wiki emphasizes the following differences when comparing with a traditional web site[1]:

- A wiki encourages users to create new pages or edit existing pages, with contract to traditional web sites that contents are provided by webmasters.
- Connections of concepts across pages are created via hyperlinks. Such a link directs users to another page if that page exists, otherwise leads users to create a new page.
- Any editing can be performed in a plain web browser without needs of installing plug-ins.
- A wiki tries to involve web site visitors in a continuous editing process.

To help users to easier be involved in editing, wiki uses a simple markup language known as wikitext to assist formatting and linking. Comparing with the common markup language for web sites – Hypertext Markup Language (HTML), its advantages include simpler to use, less complicated tags and unified formatting in output. Table 1 demonstrates an example of wikitext with comparison of its corresponding HTML and the output.

---
² http://atlassian.com/confluence
³ http://www.mediawiki.org/
⁴ http://moinmo.in/
Table 1: Comparison of wikitext and HTML

<table>
<thead>
<tr>
<th>Wikitext</th>
<th>HTML</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>The quick brown [[fox]] jumps over 'the lazy dog'. This is funny.</td>
<td>&lt;p&gt;The quick brown &lt;a href=&quot;fox&quot;&gt;fox&lt;/a&gt; jumps over &lt;em&gt;the lazy dog&lt;/em&gt;.&lt;/p&gt; &lt;p&gt;This is funny.&lt;/p&gt;</td>
<td>The quick brown fox jumps over the lazy dog. This is funny.</td>
</tr>
</tbody>
</table>

1.2 WIKIPEDIA AND MEDIAWIKI

Wikipedia is a project started by Jimmy Wales and Larry Sanger in January 2001. At the beginning it acted as a complementary project for Nupedia, a free online encyclopedia projects whose content were written by experts and examined through a peer review process [4]. Thanks to the nature of wikis, the openness of Wikipedia attracts volunteers worldwide to contribute its content, as a result produced the largest free online encyclopedia. Internationalization began soon after media coverage on the English version of Wikipedia which was the only language edition at the start point of the project. The first non-English edition of Wikipedia is the German Wikipedia which is created in March 2001. Other languages of Wikipedia (e.g. Chinese, Dutch, Esperanto, Hebrew, Italian, Japanese, Portuguese, Russian, Spanish and Swedish) were initiated soon afterwards [5]. Totally over 250 languages of Wikipedia are online nowadays. The human knowledge, regardless languages and cultures, has started to concentrate into Wikipedia since then.

Statistics display that the number of articles and authors have been growing rapidly in recent years. At the time of writing, there are over 3 million articles in the English Wikipedia, while the German and the French ones contain over 1 million articles. On the other hand, the number of users reflects the success of Wikipedia. For the English edition, there are over 14 million users registered in the system. While Wikipedia enjoys a rapid expansion over years, some people worry about the accuracy of the
content in Wikipedia. A research suggests that its content is generally in high quality, and the quality of some articles can compare to the Encyclopedia Britannica [6].

The core software running behind the large encyclopedia is MediaWiki. MediaWiki is an open source wiki software originally developed for use on Wikipedia. Now this software package is widely used in many wiki web sites as well as organizational wikis. It is written in PHP programming language and relies on a MySQL back-end database server to persist the content. General functionalities such as browsing, editing, page linking and revision history are provided by MediaWiki.

The Wikimedia Foundation is the non-profit organization that operates Wikipedia. They target to empower a global volunteer community to collect and develop the world's knowledge and to make it available to everyone for free and for any purpose. For this reason, they provide the entire Wikipedia database for anyone to access free of charge, and this makes many research works, including this one, feasible to be launched.

1.3 RESEARCH PROBLEMS

Considering Wikipedia as a result of large human collaboration, Wikipedia has a great value that has not yet been fully researched. Researches on Wikipedia have focused on both micro-level and macro-level of analysis. A micro-level of analysis typically focuses on a single article by examining its content or the related revision history. Our research is a macro-level of analysis, which studies the wiki as a whole, exploring relationships and the evolution of the entire content collection.

Wikipedia articles cover knowledge of different topics and contributed by worldwide authors. Authors assign categories to articles to identify the topics that are related to. Categories are further divided by authors into many sub-categories to represent more detail, precise sub-topics. In this way a category hierarchy, with main topics of human

5 http://wikimediafoundation.org/
knowledge (e.g. science) at the top, and more specific sub-topics (e.g. computer science) levels down, is created.

*Semantic coverage*, a term that represents the distribution and relationship among topics in Wikipedia, is firstly coined in the work of Holloway et al [7]. It is worth studying not only for analyzing the structure of a wiki in computer science field, but also for understanding the trend of authoring in social science field. With topic distributions of different language editions of Wikipedia, researchers can conclude characteristics of authors of using different languages as well. Therefore an analysis of semantic coverage can be seen as an entrance for further study on Wikipedia.

However many challenges are facing to obtain the semantic coverage of Wikipedia. First, the data structure of the category data structure in Wikipedia is on the whole very similar to a tree hierarchy but indeed it is a graph. However, because of the manual category creation, removal and assignment process, conditions such as multiple parents, loops and other anomalies exist. This makes the processing of category data different from existing methods designed for trees and graphs. Moreover, it is difficult to analyze the raw data since it involves a huge amount of data. The computation required for processing over 3 million articles and their mutual relationship with categories increases exponentially.

Although some data analysis and visualization tools for Wikipedia exist and these tools are capable with large amount of input data, the output produced by many of these tools is usually difficult to understand by untrained users. Considering not every user is familiar with computer scientific visualizations, these tools become an obstacle to study the hidden value of Wikipedia.

### 1.4 SOLUTION APPROACH

In the previous section we introduced some problems facing with Wikipedia data analysis. Here we will briefly describe our proposed solution to the problems. The first step is to clean up the category data structure by locating anomalies and remove them. After that, topic relationships are calculated with category connections, and
finally these values are used as input to a new visualization algorithm – map-like visualization. The goal of this visualization is to produce an overview visualization for Wikipedia that can produce a virtual map for semantic coverage with an appearance similar to a geographic map, as even untrained end-users can usually readily understand and relate to such maps.

1.5 OUTLINE OF THIS THESIS

In the following chapter, we review some researches and algorithms which are related to our work. Chapter 3 gives an overview of the whole process of generating the visualization. Details of data pre-processing algorithms, as well as methods of handling data processing challenges of the Wikipedia are described in chapter 4. Evaluations and applications of this research are discussed in chapter 5. Finally, we conclude our contribution in this research and state future work in chapter 6.
CHAPTER 2: LITERATURE REVIEW

Our study on Wikipedia covers different aspects of past researches, including graph operations, statistical analysis, data mining and visualization techniques. The section gives a brief overview of some related work.

2.1 INFORMATION VISUALIZATION

Figure 1: Treemap visualization of a folder in a file system (reproduced from [10])

Complex and huge datasets are difficult to understand by human being. Even computers are able to process large amount of data at a time, but human is irreplaceable for concluding features of the data. Information visualization is a technique to address these problems, by transforming data, information and knowledge into visual form, so that human’s visual capabilities can make use of [8].
In past decades, a number of information visualization algorithms are developed. These algorithms serve a purpose of reducing the number of dimensions of original data to either 2-dimensional or 3-dimensional output. Tree-map is a widely used example of information visualization which displays hierarchical information structures in a 2D plane (Figure 1) [9] [10]. When dealing with large amount of data, information visualization algorithms often faces challenges like decreasing performance and overlarge results. Researchers have been looking for solutions to address these problems. Taking the aforementioned Treemap as an example, the research of Fekete and Plasiant explores possibilities in various aspects, such as hardware accelerations, users’ perceptions, interpolation, query optimizations, etc., and achieved to render a visualization image with a level of million items (Figure 2) [11].

![A treemap with one million of items](reproduced_from_11)

Other kinds of hierarchical visualizations include the radial and the balloon view of tree visualization [12] [13]. A radial layout is derived from the simple tree graph with
the child nodes placed around the circle, which behaves well in general. A balloon view of the tree is obtained by projecting the tree onto the plane, where sibling sub-trees are included in circles attached to the parent node [13]. In addition to show the parent-child relationship in the visualization, Holten made a variant of the tree visualization to illustrate the inter-connections among independent items, as shown in Figure 4. A user study shows that the way of presenting relations in hierarchically organized systems is useful for quickly gaining insight of the connections [14].

Figure 3: Two representations of a tree: (a) the radial layout (b) the balloon layout (reproduced from [13])

Figure 4: Connections added to: (a) a radial tree (b) a balloon tree; both visualizing function calls in a software (caller in green and callee in red) (excerpt from [14])
Besides the visualization itself, users’ interaction with the visualization is another direction to study in past years. Since the size of an output device sometimes is not big enough to display all the data, in such case the algorithm needs to either allow the user to navigate to different parts of the output, or provide an overview to the user firstly and allow zooming in/out to view details. Some implementations and user experiments in this direction can be found in [15] [16]. SpaceTree is an implementation of interactive visualization which displays a tree structured data [17]. At first it shows an overview of the scaled tree that fits the size of the screen. A triangle icon in dark color summarizes the branches which cannot be expanded. Users are able to navigate the visualization and select the interested node to examine with more details. Figure 5 demonstrates the user interface of SpaceTree.

Figure 5: The screenshot of SpaceTree (reproduced from [17])

Haber and McNabb described a conceptual model for information visualization [18]. The whole visualization process involves three transformations of data and finally produces the output image as shown in Figure 6. The first step, data enrichment/enhancement, transforms the raw input data into derived data which used by later operations. In visualization mapping, abstract visualization objects (AVO)
are constructed from the derived data. AVO carries the attribute fields needed for the visualization. The final rendering procedure generates at least a displayable image using AVOs mentioned above. Most of the visualization methods, including the one introduced in this thesis, follow this model.

![Figure 6: Haber and McNabb’s visualization model](image)

### 2.2 WIKIPEDIA DATA PROCESSING

A number of elements, such as articles, categories, revision histories, user discussions, etc., make up the content of the free online encyclopedia – Wikipedia. Various relationships connect different kinds of these elements. For example: articles are classified with categories; categories are assigned to other categories as sub-categories; articles are linked to other articles by linked words. According to the Haber and McNabb’s visualization model, original data should be digested to obtain a more concrete dataset for the focus of the visualization. Besides, due to a complete manual manipulation of these data, errors in the data increase the difficulty of analyze and process the data. In this section I will briefly introduce some research that focus on data processing of Wikipedia.

Wikipedia has a category system that is similar to a tree structure. However, because the creation and maintenance of the category structure is a manual task performed by Wikipedia users, a small number of cases such as multiple parents, loops, and other anomalies exist. Indeed the category system of Wikipedia is a directed acyclic graph that multiple parents may exist for a single node and cycles may be found in the graph. However, trees are more preferable to use in many cases for their simplicity. Yu et al. remove multiple repeated parents in sub-categories using Dijkstra’s shortest path algorithm, by keeping the parent which is closest to the root and discarding the other
Zesch and Gurevych suggest another simple mechanism to solve the cycling problem. They use a depth-first search to traverse the category graph. Whenever cycles are detected among the nodes of the same level, they simply remove one of the links on that level to eliminate the cycles [20].

The raw data of Wikipedia is stored in relational database tables. Raw data includes but not limit to contents of articles, information about categories, edit histories and links between articles. These data are enough for the Wikipedia’s daily operations but not for academic research. Research generally needs condensed figures from the raw data. WikAPIdia⁶ and DBpedia⁷ are some examples of research outcomes for preprocess Wikipedia data and provide analyzed data for other purposes. WikAPIdia is a tool to connect concepts by any number of spatial reference systems (for instance geography, time and etc.), and across languages. It also provides some meta-data like number of edits and editors of an article [21]. DBpedia analyzes the user-written data and attempt to create a semantic ontology of Wikipedia. Therefore formal queries can be used to retrieve Wikipedia data [22].

2.3 WIKIPEDIA VISUALIZATION

Wikipedia contains large volume of data, not only contents and images of articles, but also evolution of contents (e.g. revisions) as well as users’ editing behavior. The analysis of this information can be carried out with visualization methods which utilize human visual capability to search for related data patterns. Some research focused on applying visualization techniques on Wikipedia and produced various visualization outcomes for different purposes. This section briefly introduces these visualization methods.

Chris Harrison developed a tool called WikiViz⁸ which generate an overview of Wikipedia categories and their connections with dots and lines. Sizes of dots and

---

⁶ http://collablab.northwestern.edu/wikapidia_api/Wikapidia/Home.html
⁷ http://dbpedia.org/
⁸ http://www.chrisharrison.net/index.php/Visualizations/WikiViz
thickness of lines are used to illustrate the number of connections in the Wikipedia graph of category links. Figure 7 is a WikiViz example with the category “Politics” centered, and includes Wikipedia data of five levels deep. Figure 8 displays an enlarged part of the visualization with different sizes of nodes and thicknesses of lines indicating the number of connections between nodes.

Figure 7: Five levels deep, centered on Politics

Figure 8: Close up of Chris Harrison’s WikiViz output
Biuk-Aghai introduces a visualization tool known as WikiVis for showing co-authorship in the Wikipedia [23]. This tool provides a collection of views, namely entity view, category view and search result view, for displaying the relationship of the content in the Wikipedia. Figure 9 is an instance of the entity view. The entity in concern is placed at the center of the image, while other related items are displayed around as nodes. Nodes with different shapes and colors represent different types of entities, such as the light blue cubes represent other articles and the red cones denote images. Shorter distance between nodes indicates a stronger degree of co-authorship, that is, proximity is used for expressing the relationship of the data.

Figure 9: Entity view with labeled nodes in WikiVis (reproduced from [23])

Figure 10: WikiVis category view (reproduced from [23])
The category view of WikiVis illustrates the relationship of categories and sub-categories which shown in yellow, as well as links to related articles which colored in white (Figure 10). Certain navigation tasks are available in the category view. Users can navigate by expanding a given category node, and explore deeper levels of sub-categories by repeating look into category nodes.

Figure 11: Wikipedia category visualization (reproduced from [7])
Another way of visualization is to produce an overview of Wikipedia. Holloway et al. render wiki categories as dots of different colors, representing the semantic coverage, which is formed by categories [7]. Figure 11 shows an example of this visualization – colored dots represent identified categories as shown in the legend, while grayed ones are the contents that cannot be classified into their pre-defined categories.

![Figure 11](image1.png)

Figure 11: An overview of Wikipedia categories.

Other types of visualization focus on analyzing editing activities and authorship. Wattenberg et al. created an application called Chromograms [24]. It displays various kinds of users’ operations on the content of Wikipedia, such as spell-checking, writing new contents, reverting changes, etc. Figure 12 demonstrates this application and derives visual patterns of users’ activities. Viégas et al. designed a visualization tool for representing authorship in an image [25]. This tool analyzes the revision history of an article to obtain authorship information. As shown in Figure 13, different colors show the different authors involved in the editing of the article, while gray and white represent the anonymous authors on the web.

![Figure 12](image2.png)

Figure 12: The Chromograms application: showing users’ activity in blocks (reproduced from [24])
Figure 13: History flow visualization (reproduced from IBM Research\textsuperscript{9})

On the opening day of Macau Fisherman Wharf, lots of people came to visit. When the clock stroke twelve on the New Year Eve, the volcano erupted, and one guy said very loudly that the volcano was too hot. He could not stand the heat and so he decided to leave the place. Actually, he didn’t want to leave, since he saw so many beautiful in Macau Fisherman Wharf. He wanted to get known to the beautiful girls first. Suddenly, he found a lovely girl who was in the ice-cream shop.

Figure 14: Author text analysis of a section of text with high depth of collaboration (reproduced from [26])

Figure 15: Edit significance bars of revisions of an article (reproduced from [27])

\textsuperscript{9} http://www.research.ibm.com/visual/projects/history_flow/results.htm
Some visualization methods show the collaboration behaviors of articles in the Wikipedia. Biuk-Aghai et al. describe a visualization in [26] to provide a “big picture” of collaboration of authors in a Wikipedia article. As denoted in Figure 14, the text inserted by different authors is highlighted with different colors. Through this kind of visualization we can easily identify the degree of participation of authors. Fong and Biuk-Aghai’s work summarizes the editing significance and the high level edit operations (such as adding content, spelling correction and deleting content) in a Wikipedia article [27]. A plug-in is added to the Wikipedia to visualize the edit significance of different revisions of an article in bar charts (Figure 15). Besides, another visualization graph displays articles that grouped with various edit behaviors (Figure 16). In this way, the administrators and authors are able to glance what’s happening in the Wikipedia.

2.4 MAP-LIKE VISUALIZATION

Map-like visualization is a novel way to visualize data which takes advantage of the intuitive understanding of geographical maps by general people. Elements such as mountains, valleys, land, rivers, sea, etc. are readily recognized by people even without special training. Therefore visualizing information in the form of a
geographic map enables people to relate to such representations more easily without requiring prior instruction. Skupin presents a method that produces a map-like visualization for a knowledge domain [28]. Figure 17 demonstrates a 10-year periodical library which categorized by keywords. Researchers, especially for those are not in the computer science field, make use of this kind of visualization to understand and discuss recent research trends in the discipline.

![Figure 17: Visualization of a 10-year period by cartographic means (reproduced from [28])](image)

### 2.5 FORCE-DIRECTED LAYOUT ALGORITHMS

Force-directed layout algorithm is a type of graph layout algorithm, which simulates a physical model of springs and electric particles to position nodes in a graph, while attempting to minimize the number of crossing edges. The algorithm treats nodes in the graph as electric particles and edges as springs. Springs act as pulling forces to pull nodes closer, while electrical particles give repelling forces to push them apart. The simulation is repeated iteratively until the system comes to an equilibrium state. Advantages of force-directed layout algorithm include easy to implement and producing good-quality results. However this type of algorithms is not scaling well with large graphs. Kamada and Kawai stated the basic implementation of this kind of algorithm in [29]. Fruchterman and Reingold made an enhancement of the layout
algorithm in [30] that near nodes are placed in the same vicinity and far nodes are placed far from each other.

2.6 OVERLAP REMOVAL ALGORITHMS

Overlap removal algorithms are layout algorithms which removing overlaps of rectangular items in a graph by arranging their positions. An early algorithm for overlap removal is the *Force-Scan Algorithm* (FSA) [31]. It is based on the concept of a *mental map*. A mental map maintains relative positions of items compared to those before movement, but distances between items may be changed. The force-scan algorithm performs its task by scanning overlapped items in both horizontal and vertical directions, finding values of forces which are able to push/pull items away in order to move them apart. Huang and Lai illustrated a new algorithm called *Force-Transfer Algorithm* (FTA) for overlap removal [32]. This new algorithm adds an enhancement to FSA in compacting the spaces between items in the output, while still maintaining the same effectiveness as FSA. Figure 18 shows adjustments of overlaps with both algorithms.

![Figure 18: Comparison of overlap removal algorithms (reproduced from [32])](image)
CHAPTER 3: MAP-LIKE VISUALIZATION

Starting from this chapter we focus on the design, implementation and application of a map-like visualization. Firstly, we explain the visualization algorithm in a general way, which is applicable to other hierarchical ontologies with acceptable changes in this chapter. The subsequent chapter then introduces a case study of the visualization on the Wikipedia, as an empirical application of this visualization.

3.1 OVERVIEW OF THE VISUALIZATION

A map-like visualization is a kind of information visualization which is different from the visualization that people usually use. A general problem of using a visualization is the difficulty of interpretation the figures which a visualization image would like to express. For this reason, the map-like visualization aims to be perceivable with minimum of training and adaption that needs for reading. In order to achieve this result, we set up these requirements for the map-like visualization algorithm as follows: (1) making shapes, lines and borders looking similar to the ones found in a geographic map, in contrast to the lines, points and their connections in scientific visualizations; (2) using suitable colors to illustrate the degree of attributes of interests so that users are able to understand as easy as reading a topological map; (3) labeling items in the visualization in a way that users can identify the depth in the hierarchy of items intuitively and efficiently.

The design of the map-like visualization algorithm can be summarized into two major steps: a preliminary layout procedure and a visualization creation based on hexagons. In the first step, the algorithm tries to generate a draft layout which is used in later steps. The remaining parts of the visualization involving the selection of hexagons using the mentioned draft layout, reducing clutters found in the visualization, coloring the visualization and performing the final text labeling work. The forthcoming sub-sections in this chapter will discuss these aspects in detail.
3.2 DATA SOURCE REQUIREMENTS

In a typical geographical map there are often several levels of political regions shown, for instance countries, provinces (or states), counties and so on. Since we are creating a visualization that looks similar to a geographic map, the similar hierarchical concepts of presenting information are borrowed to our work. In this way, the map-like visualization algorithm accepts hierarchical data source as input. Some examples of hierarchical data include category hierarchies and file systems (Table 2). Records in the dataset (i.e. a category or a file directory) are called entities in the context. Entities are displayed in regions which are in different sizes and shapes in the visualization, thus the dataset should include a size indicator, which is an integer for indicating the size of an entity (e.g. number of files in a directory) as well.

Table 2: Examples of hierarchical data (number in the parentheses represents the size)

<table>
<thead>
<tr>
<th>Category Hierarchy</th>
<th>File System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Science (10)</td>
<td>/ (0)</td>
</tr>
<tr>
<td>Chemistry (45)</td>
<td>/etc (125)</td>
</tr>
<tr>
<td>Physics (76)</td>
<td>/usr (5)</td>
</tr>
<tr>
<td>History (20)</td>
<td>/usr/bin (430)</td>
</tr>
<tr>
<td>Geography (8)</td>
<td>/var (10)</td>
</tr>
<tr>
<td>Continents (12)</td>
<td>/var/tmp (1021)</td>
</tr>
<tr>
<td>Asia (65)</td>
<td>/var/www (54)</td>
</tr>
</tbody>
</table>

In addition to the hierarchical data, we need a measurement to define the relationship between every entity within the hierarchy. This number is usually a similarity value of a pair of entity, which reflects the commonness of both. The value should be normalized to a floating-point number ranging from 0.0 to 1.0, whereas the bigger value means they are more common to each other. The visualization algorithm takes account of this factor to layout individual entity in the output. In principle the algorithm tries to position similar entities (i.e. the ones with higher the similarity values) closer.

In summary, the map-like visualization algorithm is able to operate with minimal two set of data: a hierarchical dataset (with size of individual entities) which the user
wishes to visualize, and a set of numbers specifying to relationship among entities in the dataset.

### 3.3 Sorting Algorithm by Similarities

A common problem faced in visualization is arranging items in order. General sorting algorithms accomplish this task by comparing a property (e.g. size, length or weight) associated with items, and the data type of the property usually is a number. However, in our case the source data consists of a pair of entities and a value indicating the similarity. As existing sort algorithms cannot be applied to the association of one number (the similarity value) to two entities, we thus designed a sorting algorithm to operate on the similarity of pairs.

The sorting algorithm is illustrated in Table 3. The input of this algorithm is a list of entity pairs with their similarities. The largest element, i.e. the entity pair with the highest mutual similarity, is placed at the center of the output list. Subsequent pairs are appended to either left or right hand side of the output. Initially the input list is sorted in descending order simply by their similarity values. Pairs other than the centered one in the output are processed one by one. The algorithm looks for the next entity that can be placed on either left or right hand side of the current output list. For a pair of entities, if anyone in the pair equals to the entity on the left or right tail, the other entity in the pair is appended to the end of the corresponding direction. Conversely, if no such pair can be found in the input list then the first entity pair is taken from the input list and split into left and right item which are placed on the left and right end of the output. These steps are repeated until all category pairs in the input list have been processed. In this way, the most related elements are placed next to each other, and this represents an order from highly related elements closer to the center to unrelated ones on the periphery.

This algorithm is designed for the preliminary layout during the visualization process, as we discuss in the following section. However, it can also be applied to other scenarios that need to sort elements with attributes involving pairs of items.
input.sort(similarity, DESCENDING);
result := input.first;
input.remove(input.first);
while not input.isEmpty do
    found := false;
    entry := input.first;
    while not found and not entry.isNull do
        if match(entry, result.left) then
            result.add(L, entry);
            input.remove(entry);
            found := true;
        else if match(entry, result.right) then
            result.add(R, entry);
            input.remove(entry);
            found := true;
        end if
        entry := input.next;
    end while
    if not found then
        result.add(L, input.first.item1);
        input.remove(input.first.item2);
        input.remove(input.first);
    end if
end while

3.4 PRELIMINARY LAYOUT

Figure 19: The bottom-up layout approach
Before creating a map-like visualization in accordance with the data provided, we first generate a rough layout from the data hierarchy which is called a preliminary layout. This layout contains approximate positions and estimated sizes of regions that represent individual entities. Our algorithm is designed to handle over ten thousands data records. In order to prevent performance bottleneck caused by total number of entities and the exponent increasing number of similarity pairs, we adopt a bottom-up approach to process a part of information at one time. The layout algorithm respects the data hierarchy that processes the layout level by level, starting from the lowest level to the top. This process is illustrated in Figure 19. A lower level entity (e.g. a category in a categorization system) settles the position of its children, as shown in Figure 19(a). After this, upper level ones then combine the layout from the lower levels and also adjust their positions as shown in Figure 19(b). This aggregation of subordinate layout continues in even higher levels as displayed in Figure 19(c) until the algorithm reaches the top.

A spring force-directed layout algorithm is applied at each level. Similarities are fed into the spring algorithm, acting as “forces” among entities. Entities are linked with each other with two forces: a pulling force which is proportional to the mutual similarity; and a repulsive force that prevents their placement too close. The layout algorithm adjusts the positions of entities until they are stable and forces are balanced.

The size of the region that represents a particular entity is computed with the size indicator obtained from the data source. We observe that the size indicators from the input data ranges varied. To avoid producing extreme small or big regions in the final visualization, we use the following log scale function to determine the size of entity regions. For a category C, the region size \( region(C) \) is the \( b \)-based logarithm of the size indicator of category C \( size_C \). In other words, the parameter \( b \) controls the increasing speed of the region size when the size indicator getting larger. Also, a multiplier \( m \) is introduced to magnify the final region size.

\[
region(C) = \log_b(size_C) \bullet m
\]
The initial placement of categories before applying the force-directed algorithm is crucial to the layout. Our empirical experiments showed that if related entities are initially placed in a shorter distance, the spring layout algorithm will have a better performance to achieve the balance state. For this purpose, we apply the sorting algorithm discussed in the previous section, and then arrange entities in the order of the sorted result on a circle. Instead of placing items evenly distributed around a circle, which would result in more overlaps (Figure 20(a)), we divide a circle into segments according to item sizes, and place the items within the corresponding segment (Figure 20(b)) to avoid overlaps.

As shown in Figure 20, a radial placement cannot remove all overlaps. It is possible that entities may overlap each other after the layout process. An overlap removal algorithm is subsequently applied to solve this issue. As mentioned in chapter 2, there exist many overlap removal algorithms, such as Force-Scan Algorithm (FSA). However, one drawback of FSA is that it results in large spaces between items after removal. This is not suitable for a map-like visualization, as it results in an unnatural appearance of the map. Therefore we chose to apply the Force-Transfer Algorithm (FTA) instead, which is a modified version of FSA but produces more compact graphs.

![Figure 20: Approaches for radial placement of sorted entities](image)

Eventually the preliminary layout consists of the approximate layout with the sizes and positions of all categories which are projected in a two-dimensional plane. In the
following section we discuss a method which uses this preliminary layout to generate the final visualization.

3.5 HEXAGONAL VISUALIZATION

Map-like visualization targets to make the visualization looked more natural in contrast to general scientific visualizations. Inspired by Dr. Skupin’s work, we adopt his approach to create the visualization with hexagons. This section introduces the core part of the map-like visualization – the way of using hexagons to make a natural output. These include the basic arithmetic and concepts of hexagons, and the algorithm of drawing and coloring hexagons according to the source data.

3.5.1 THE HEXAGON CANVAS

The hexagon canvas is the 2-dimensional plane filled with hexagons which are drawn in different colors according to the data. Hexagons are a good choice of shape because they are able to tile a surface flawlessly, and at the same time they outline a natural looking zigzag effect (similar to the seashore in real-life maps) with their borders. The
similar usage of hexagons in visualization can also be found in Dr. Skupin’s work. The canvas is stored in a normal 2-dimensional array in the memory. Nevertheless, odd number columns are displayed lower with the height of half of a hexagon, as shown in Figure 21, in order to ensure the flawless property of hexagons.

Figure 22: Arithmetic of a hexagon

Figure 22 illustrated the arithmetic of an individual hexagon. Parameter \( len \) is a pre-defined side length for each hexagon. Points are numbered from 0 to 5 starting from the one at the 1 o’clock clockwise direction. Given a coordinate \((x, y)\) for the center point, we can further derive the coordinates for other points as follows:

- Let \( m = len \cdot \cos(60^\circ) \)  \( n = len \cdot \sin(60^\circ) \)
- \( x_0 = x + \frac{1}{2} len \)  \( y_0 = y - n \)
- \( x_1 = x + m + \frac{1}{2} len \)  \( y_1 = y \)
- \( x_2 = x_0 \)  \( y_2 = y + n \)
- \( x_3 = x - \frac{1}{2} len \)  \( y_3 = y_2 \)
- \( x_4 = x - m - \frac{1}{2} len \)  \( y_4 = y_1 \)
- \( x_5 = x_3 \)  \( y_5 = y_0 \)

With these coordinates, the edges of a hexagon are simply drawn by connecting points to each other.
3.5.2 **Selection of Hexagons**

![Hexagon Diagram](image)

Figure 23: Example of hexagon selection

The pseudo-code of the algorithm is listed in Table 4. The essential operation of the map-like visualization algorithm is to assign hexagons with the result of the preliminary layout. In the preliminary layout result, each entity has a *pivot point* to represent its pivot in the output plane. A pivot point is initially the geometric center of the region of an entity. However, during the adjustment steps (e.g. overlap removal) in the preliminary layout, the pivot point remains in the same position, and thus the pivot is moved in relation to the whole region. Also, this pivot point is a balanced position of all the similarity factors applied on. For every entity item, the algorithm selects the hexagon whose coordinate corresponds to the pivot point of the entity. Starting from this hexagon, we randomly select the next hexagon among the neighboring ones. A data structure maintains a list of “territory” (i.e. occupied hexagons) of the current entity and randomly expands the selection to unused neighboring hexagons, as illustrated in Figure 23 (numbers correspond to the order of hexagon selection). The number of hexagons occupied by one entity relates to the size of that entity.

The reason of using random numbers in this part of algorithm is that we want to preserve an appearance which looks similar to those in a geographic map. Factors such as irregular shapes, various line length, colors used, etc. are important in
achieving a visualization that looks realistic and resembles a geographic map. Through experimentation we discovered that this form of random assignment produces regions with the preferred result. In order to create reproducible output, however, we control randomness throughout our program by using a fixed random seed (Hexadecimal number 8d14c220) to the pseudo-random number generator. Thus the same input data will always result in the same visualization, and make different visualizations comparable with each other. This seed is randomly selected during the early development of the project and remains for backward compatibility.

Table 4: Algorithm for randomly selecting hexagons

```plaintext
// input parameters
// pivot: pivot point; size: category size

init_random_seed(0x8d14c220);  // A fixed random seed.
count := 0;
territory := [pivot];
while count < size do
    picked := territory[rand()];
    target := find_available_neighbour(picked);
    if target != null then
        select(target);
        territory.put(target);
        count++;
    end if
end while
```

3.5.3 Clutter Reduction

Due to the nature of random selection of hexagons, in the result there exist “orphaned” hexagons, which are surrounded by all empty hexagons (i.e. water) or occupied hexagons of another entity. Examples of these problematic hexagons, which appear similar to small islands or regions/lakes in land, are identified with arrows in Figure 24(a). Consequently these hexagons cause the final visualization look distracting and cluttering. To address this problem, we introduce an enhancement to remove the clutter with these rules:
- If a hexagon is selected, and surrounded by empty hexagons (water) on all six sides, this hexagon will reset to empty state.
- If a hexagon is selected, and surrounded by hexagons of other entities, this hexagon will be merged into the surrounding ones.

Figure 24(b) demonstrates the effect of clutter reduction. The visualization result looks clearer as a result.
3.5.4 COLORING SCHEME

Figure 25: A topographic map uses colors for showing elevation

Color is one of the primary visual elements that a reader perceives in a visualization outcome. Topographic maps often employ various colors to represent elevation (Figure 25). We use it to represent the size of an entity. This allows users to quickly spot large and small items, and to perceive their distribution throughout the visualization. The color scheme we employ is similar to that of topographic maps: a darker colour represents a larger size (Figure 26).

Figure 26: Legend of a map-like visualization
3.5.5 TEXT LABELING

The purpose of placing text labels is to help users to quickly identify different data items conveniently. For a hierarchical data source, users would like to know the hierarchical levels at the first sight. Thus we can observe that in a real political map, authors make use of font sizes, colors, and font styles (e.g. normal/italic/underline) for distinguishing names of countries, states and cities, which belong to different levels in the hierarchy. We reference the similar concepts and apply to our work. As demonstrated in Figure 26, **TOP LEVEL ENTITIES** are highlighted in bold, underline and uppercase; **SECOND LEVEL** names are bold and upper; other names are remain normal font and first letter capital.

Real geographic maps often are created using manual placement of text label to avoid overlaps and produce an optimal appearance. However, given the data sizes of map-like visualization datasets this approach is clearly infeasible. In our work we choose to place text labels automatically. As shown in Figure 27, we draw a rectangle for every region which can contain all its area. The text is initially placed at the center point of the rectangle center-aligned. Then we draw a cross line from the left-most point and the right-most point. The angle $\theta$ between the cross line and the horizontal line is measured. The text label is rotated according to the angle $\theta$. However, the text is only rotated either in 0, 45, 90 or 315 degrees to keep the map tidy.

![Figure 27: Rotation of text labels in a region](image)

- 41 -
Furthermore, where necessary, we reduce font sizes of label text to enable labels to remain inside the area of their respective entity, with a minimum allowed font size to ensure readability. As displayed in Figure 28, the algorithm tries to scale the text so that it can be drawn within a region. At the same time the result is looked tidier thanks to the constraint on the rotation angles.

Figure 28: Result of the text label placement
CHAPTER 4: WIKIPEDIA DATA ANALYSIS

The ultimate goal of this research is to produce a visualization of the Wikipedia’s overview. Producing a visualization of the Wikipedia is faced with complexity because of the volume of its data, as well as idiosyncrasies contained in it. Firstly, the number of articles and categories in the Wikipedia can be large, depends on the language. The English Wikipedia, the largest one among more than 200 Wikipedia language editions, occupies over terabytes storage including related meta-data. If we only count the data rows of pages and categories without their contents, the dataset can be still over 15GB. Data volume affects the running time of our algorithms exponentially. Secondly, as a result of the human editing process, the Wikipedia category structure is an undirected graph, rather than a tree hierarchy common in many other classification systems. The category graph contains duplicate entries, loops and non-content categories and consequently requires cleaning and pre-processing prior to visualization. This chapter describes the techniques to transform the Wikipedia data into a hierarchical tree structure, which the map-like visualization algorithm accepts. Although our work is designed for Wikipedia and its backend system MediaWiki, however, concepts and operations of the method should be generic enough to be applicable on other wiki systems with only minor modification.

4.1 DATABASE SCHEMA OF MEDIAWIKI

MediaWiki is the wiki system behind the Wikipedia. It is created by the Wikimedia Foundation for the use in the Wikipedia, and then its usage is extended to other projects in the foundation. The software is written in PHP language, and operated on a MySQL database which stores all articles and contents. The complete description of all database tables can be found on the MediaWiki’s wiki\(^\text{10}\). Among the tables in the

\(^{10}\) http://www.mediawiki.org/wiki/Manual:Database_layout
database schema, our method focuses on only two tables, namely “categorylinks” and “pages”. The “pages” table stores the title, namespace and other meta-data. Indeed elements in MediaWiki including articles, categories, discussions (i.e. talk pages) are considered to be “pages” and correspond to a record in the “pages” table. Three attributes, namely page ID, page title and namespace are essential for one page record. Page ID and title are unique identifiers in both numerical and descriptive ways, while the value of the namespace field tells the type of the page. Full definition of namespaces can be found on the MediaWiki website\textsuperscript{11}. Table 5 gives some examples of them.

<table>
<thead>
<tr>
<th>Numerical Value</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Main namespace (articles)</td>
</tr>
<tr>
<td>1</td>
<td>Talk pages</td>
</tr>
<tr>
<td>6</td>
<td>Uploaded files</td>
</tr>
<tr>
<td>14</td>
<td>Categories</td>
</tr>
</tbody>
</table>

Table 6: Definition of the “categorylinks” table

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Nullable?</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>cl_from</td>
<td>int unsigned</td>
<td>No</td>
<td>Primary</td>
</tr>
<tr>
<td>cl_to</td>
<td>varchar(255)</td>
<td>No</td>
<td>Primary</td>
</tr>
<tr>
<td>cl_sortkey</td>
<td>varchar(255)</td>
<td>No</td>
<td>--</td>
</tr>
<tr>
<td>cl_timestamp</td>
<td>timestamp</td>
<td>Yes</td>
<td>--</td>
</tr>
</tbody>
</table>

The “categorylinks” table manages links across pages. Assignment of one category to another is realized by a link record between two categories. As a result we can obtain the category hierarchy with this table. Table 6 displayed the data structure of the “categorylinks” table\textsuperscript{12}. In case of a category assignment, the page ID of the sub-category (\texttt{cl\_from}) will be connected to the page title of the parent (\texttt{cl\_to}), and this record is saved in the table. Sort keys (\texttt{cl\_sortkey}) are used for arranging

\textsuperscript{11} http://meta.wikimedia.org/wiki/Help:Namespace

\textsuperscript{12} http://www.mediawiki.org/wiki/Manual:Categorylinks_table
sub-categories in order whenever multiple children are found. Timestamps (cl_timestamp) are time references of the establishment of connections among categories.

4.2 OBTAINING WIKIPEDIA DATA

The Wikimedia Foundation, which is the organization operates the Wikipedia, opens the access the Wikipedia data to the public. The data is available in all language editions through database dumps and can be downloaded at MediaWiki’s database dump website\(^\text{13}\). A full database dump provides data in XML format. However, we concentrate on few tables as described in the previous section, and as a result we choose to download SQL database dumps of individual tables.

<table>
<thead>
<tr>
<th>Language</th>
<th>Number of Articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Danish</td>
<td>154,387</td>
</tr>
<tr>
<td>Chinese</td>
<td>371,772</td>
</tr>
<tr>
<td>Swedish</td>
<td>407,803</td>
</tr>
<tr>
<td>German</td>
<td>1,278,718</td>
</tr>
<tr>
<td>English</td>
<td>3,721,656</td>
</tr>
</tbody>
</table>

We processed the data from database dumps of different language editions: Danish, Chinese, Swedish, German and English, all from January 2011. The approximate numbers of articles in these language editions are shown in Table 7. We selected these database dumps for several reasons: firstly because my supervisor and I are conversant in some of these languages (which is needed for interpreting the visualized result). Moreover, this selection gives us a selection of very large (English), medium-sized (Chinese, Swedish) and small (Danish) Wikipedias; moreover, the English version is about three times the size of the German one, which in turn is about three times the size of both the Chinese and the Swedish versions. This enables us to

\(^\text{13}\) http://dumps.wikimedia.org/
compare wikis of a different scale as well as pairs of wikis of a similar relative size difference.

4.3 TRANSFORMING CATEGORY GRAPH

As discussed in the previous chapter, our map-like visualization algorithm processes hierarchical data, and transforms the data into a graphical output. Unfortunately, the data structure of categories in Wikipedia is a graph with idiosyncrasies. This section introduces a methodology to convert the graph into a hierarchical format. In the meantime, idiosyncrasies such as looping edges and multiple parents are removed.

4.3.1 CHOOSING SEMANTIC ROOT

Table 8: Top content categories in different Wikipedia language editions

<table>
<thead>
<tr>
<th>Swedish</th>
<th>German</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Topp</strong></td>
<td>!Hauptkategorie</td>
<td><strong>Main topic classifications</strong></td>
</tr>
<tr>
<td>Geografi</td>
<td><strong>Sachsystematik</strong></td>
<td>Contents</td>
</tr>
<tr>
<td>Historia</td>
<td>Geschichte</td>
<td>Articles</td>
</tr>
<tr>
<td>Kultur</td>
<td>Kultur</td>
<td>Culture</td>
</tr>
<tr>
<td>Personer</td>
<td>Personen</td>
<td>People</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

The direct acyclic data graph of Wikipedia begins at the root node. But in each Wikipedia language edition, it has its own category organizations with no standard node designated as the root node, nor any standard on where under the root node content-related categories are placed. In this case, we have to decide the semantic root, which represents the starting point of the category graph in the context. For instance, as shown in Table 8, content categories are created at the level directly under the root node in the Swedish Wikipedia, two levels below the root in the German Wikipedia, and three levels below the root in the English Wikipedia. Therefore we identify a semantic root which constitutes the parent node of the top-most content category nodes. On the other hand, categories are named in their language. Automatically
determining the semantic root becomes difficult due to these reasons. Thus the semantic root node needs to be manually identified (shown in boldface in Table 8).

4.3.2 Removing Non-Content Categories

The Wikipedia category structure contains non-content categories which are not useful for our analysis, and indeed would adversely affect the calculation of similarity and the visualization in the later steps. This mainly includes three types of categories: (1) Wikipedia administrative categories, (2) stub categories and (3) list categories. These types of category nodes need to be removed, each of which requires a different approach.

The first two types of non-content categories are relatively easy to be eliminated. Wikipedia administrative categories are normally located in the “Wikipedia” namespace, and we can simply drop the categories in this namespace. On the other hand, stub categories contain articles which have not enough content and need expansion. Names of these stub categories usually contain the term “stub”. In this way we can easily identify and remove them by looking for a particular substring in category names.

List categories, for example “1879 births”, “1976 deaths”, “History of China by period” and others are convenient for readers to look up articles, but are not useful to include in the final visualization as they are large in number and do not actually contain article content themselves. These categories usually repeat certain keywords in their names, such as “births”, “deaths” and “History of”. We can record the occurrence of such words in category names. Words that appear frequently in sibling nodes (i.e. under the same parent node in the category graph) are assumed to be part of such list categories, so these nodes are removed.

One characteristic of these list categories is that they share similar category names in sibling categories under the same parent category, for instance, list category “Mammals of Norway”, “Mammals of Latvia” and “Mammals of Germany” are placed under the parent “Mammals by country”. Given this knowledge we can
develop a method to identify list categories by computing similarities of category names.

The method for removing list categories works as Equation 1. The calculation places the characters of a pair of category names into a vector, and then applies the cosine similarity formula to measure the degree of how much they have in common. This character-basis algorithm has an advantage over other methods in that it is applicable to Asian languages in which the basic linguistic component is not a word but a character, such as in Chinese, Japanese and Korean. In the formula below, \( \text{name}_1 \) and \( \text{name}_2 \) are the names of a pair of categories. \( CO_1 \) and \( CO_2 \) are the numbers of common characters occurring in both names. \( C_1 \) is the total number of characters in \( \text{name}_1 \), and \( C_2 \) is the total number of characters in \( \text{name}_2 \). Finally, \( \text{sim}(\text{name}_1, \text{name}_2) \) is the similarity value of the two names.

\[
\text{sim}(\text{name}_1, \text{name}_2) = \frac{\sum (CO_1 \cdot CO_2)}{\sqrt{\sum (C_1)^2 \cdot \sum (C_2)^2}}
\]

For every sibling category under the same parent category, we compute the similarity values of a pair one by one. Finally we compute the average value as a threshold value \( \theta \) from these similarities as shown below:

\[
\theta = \text{avg}(\text{sim}(\text{name}_1, \text{name}_2), \text{sim}(\text{name}_2, \text{name}_1), \ldots, \text{sim}(\text{name}_{n-1}, \text{name}_n))
\]

When the threshold value is greater than a certain pre-defined value (we have experimentally determined that 0.8 is a good value), all these categories are classified as list categories because their names are highly similar to each other.

Table 9 illustrates the comparison of similarity values of two parent categories. The first parent category “Aircraft 1950-1959” contains list categories which classify aircraft articles by year, whereas the latter one consists of general computer science topics. It is obvious that the list categories produce high similarity values, and thus result in a high average value. Therefore all these categories are identified for removal.
On the other hand, “normal” categories are not affected by the algorithm. We can see that topics like “Computer graphics” and “Computer security” also give high similarity between their names, however, other computer science categories contribute to make the average value lower than the threshold. As a result these categories are preserved.

Table 9: Comparison of name similarities under category “Aircraft 1950-1959” and “Computer science”

<table>
<thead>
<tr>
<th>Aircraft 1950-1959</th>
<th>Similarity</th>
<th>Computer science</th>
<th>Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Civil aircraft 1950-1959</td>
<td>0.932</td>
<td>Algorithms</td>
<td>0.529</td>
</tr>
<tr>
<td>Italian aircraft 1950-1959</td>
<td>0.876</td>
<td>Data structures</td>
<td>0.679</td>
</tr>
<tr>
<td>Dutch aircraft 1950-1959</td>
<td>0.899</td>
<td>Operating systems</td>
<td>0.682</td>
</tr>
<tr>
<td>Soviet aircraft 1950-1959</td>
<td>0.912</td>
<td>Computer graphics</td>
<td>0.842</td>
</tr>
<tr>
<td>Military aircraft 1950-1959</td>
<td></td>
<td>Computer security</td>
<td></td>
</tr>
<tr>
<td>Average:</td>
<td>0.905</td>
<td></td>
<td>0.683</td>
</tr>
</tbody>
</table>

### 4.3.3 Building a Category Tree

At this stage, we start to build a category tree structure according to the data source, after choosing the semantic root and removing non-content categories as mentioned in the above sections. In order to create such a tree, first we apply a breadth-first search that starts from the root. We address two problems existing in the graph in this algorithm: looping edges and multiple parent nodes. The algorithm traverses every node encountered, keeping a list of visited nodes. Loops in the tree are removed by simply eliminating the edge that causes the loop (see Figure 29(a)).

For case of repeated parent relationships in multiple-parent nodes (see Figure 29(b)), we use similarity values to determine the most appropriate category for the child. For every parent-child pair of categories, a similarity value is computed with their number of articles which appears in the both categories. The larger value means a stronger
relationship between categories, and thus the pair with the highest similarity value should remain in the category tree. The details of computing similarity values are explained in the next section.

Figure 29: Eliminating edges in the graph (edge indicates “parent-to-child” relationship)

4.4 CATEGORY RELATIONSHIPS

The relationships among categories are heavily referred in our different stages of this research. Category relationships are measured by the cosine similarity values among them. Cosine similarity has been long used for computing similarity between articles related by identical keywords [33] [34] [35], however, using numbers of category assignments between wiki articles is a novel application of this method as pointed out in [7].

Usually editors assign an article to multiple categories when the topics of these categories are related to the article’s content. We can therefore assume that a pair of categories is more similar to each other if they share many common articles assigned to them. The number of common articles is referred to as co-occurrence of category
assignments between a pair of categories. A greater number of co-occurrences implies a stronger similarity and vice versa.

The cosine similarity can be calculated with the following formula:

$$\cos_{i,j} = \frac{\sum_{k=1}^{n} A_k C_{ij}}{\sqrt{\sum_{k=1}^{n} A_k C_i \sum_{k=1}^{n} A_k C_j}}$$

(3)

In Equation 3, $\cos_{i,j}$ represents the cosine similarity of categories $C_i$ and $C_j$. $A_k C_i$ is the assignment of article $A_k$ to category $C_i$, and similarly for $C_j$. $A_k C_{ij}$ is the co-occurrence of article $A_k$ in categories $C_i$ and $C_j$.

However, this above similarity formula focuses on the assigned articles of a pair of categories to compute their similarity, which we term *direct similarity*, without considering the effect on similarity by the content in their sub-categories. Intuitively the articles of their sub-categories should also contribute to a certain degree to the relationship of their parents. As illustrated in Table 10, merely considering only the direct similarity ($\cos_{i,j}$) is insufficient for most cases. The similarities of lower levels of sub-categories indicate a slight relationship between category pairs, which cannot be shown in the direct similarity.

Table 10: Cosine similarity for categories in the English Wikipedia with data from different levels of sub-categories

<table>
<thead>
<tr>
<th>Pair of Categories</th>
<th>$\cos_{i,j}$</th>
<th>1 Level</th>
<th>2 Levels</th>
<th>3 Levels</th>
<th>4 Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>History – Geography</td>
<td>0.017293</td>
<td>0.001006</td>
<td>0.000081</td>
<td>0.000020</td>
<td>0.000008</td>
</tr>
<tr>
<td>History – Culture</td>
<td>0.021874</td>
<td>0.000692</td>
<td>0.000073</td>
<td>0.000022</td>
<td>0.000010</td>
</tr>
<tr>
<td>History – Agriculture</td>
<td>0.000000</td>
<td>0.000595</td>
<td>0.000038</td>
<td>0.000009</td>
<td>0.000005</td>
</tr>
<tr>
<td>History – Politics</td>
<td>0.024456</td>
<td>0.000715</td>
<td>0.000046</td>
<td>0.000018</td>
<td>0.000010</td>
</tr>
<tr>
<td>History – Nature</td>
<td>0.000000</td>
<td>0.001001</td>
<td>0.000045</td>
<td>0.000012</td>
<td>0.000006</td>
</tr>
<tr>
<td>History – Technology</td>
<td>0.000000</td>
<td>0.000538</td>
<td>0.000043</td>
<td>0.000009</td>
<td>0.000004</td>
</tr>
<tr>
<td>History – Education</td>
<td>0.000000</td>
<td>0.000618</td>
<td>0.000039</td>
<td>0.000012</td>
<td>0.000006</td>
</tr>
<tr>
<td>History – Applied Sciences</td>
<td>0.000000</td>
<td>0.000705</td>
<td>0.000032</td>
<td>0.000010</td>
<td>0.000008</td>
</tr>
</tbody>
</table>
Therefore, we introduce the aggregate cosine similarity, which combines both the direct similarity and the similarity from subcategories, as an averaged value according to following aggregated similarity formula:

\[ ac_{i,j,n} = \text{avg}(\cos_{i,j,1}, \cos_{i,j,2}, ..., \cos_{i,j,n}) \] (4)

In this formula \( \cos_{i,j,n} \) is the average value of similarities of \( n \) levels of sub-category below a given category level. The overall aggregated similarity is the average of the similarities from \( n \) levels. The value of \( n \) is determined by the need of the scenario. Including more levels makes the final result more accurate, but at the same time it severely degrades the performance.

Figure 30: Comparison of different experimental methods for aggregating similarity

We used an empirical experiment to choose the way of aggregating similarity values. The following expressions are the three ways we tested to summarize similarities of top categories in the Wikipedia:
Figure 30 shows the results of experiments on the English Wikipedia. Firstly, the values of the direct similarity $\cos(i, j)$ and Expression C are not desirable because they magnify the differences between categories with zero and high values. We applied the same experiments on the Wikipedias in other languages. Standard deviations of values obtained with different expressions are displayed in Table 11. Expression B has the lowest standard deviation, which means it can minimize the effect of extreme values across categories, while it retains the characteristics of the values as shown in the figure.

Table 11: Standard deviations of similarity aggregation experimental expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Danish</th>
<th>Chinese</th>
<th>Swedish</th>
<th>German</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\cos(i, j)$</td>
<td>0.015995</td>
<td>0.004116</td>
<td>0.013223</td>
<td>0.011817</td>
<td>0.010498</td>
</tr>
<tr>
<td>Expression A</td>
<td>0.006372</td>
<td>0.001412</td>
<td>0.004425</td>
<td>0.006040</td>
<td>0.003513</td>
</tr>
<tr>
<td>Expression B</td>
<td>0.003932</td>
<td>0.000840</td>
<td>0.002659</td>
<td>0.002083</td>
<td>0.001761</td>
</tr>
<tr>
<td>Expression C</td>
<td>0.016071</td>
<td>0.004066</td>
<td>0.013147</td>
<td>0.013397</td>
<td>0.010286</td>
</tr>
</tbody>
</table>

Table 12 gives an example of aggregated similarity values of top category “Mathematics” of the English Wikipedia in the descending order of the similarities. From the table we can observe some conclusions: Mathematics and Science have the strongest relationship which seems valid by the common sense; Mathematics has weak relationships to People, Law and Politics categories; Mathematics is also related some other fields like Business, Computers, Humanities, etc.
Table 12: Aggregated similarity of category “Mathematics” in the Wikipedia

<table>
<thead>
<tr>
<th>Names of Categories</th>
<th>Similarity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mathematics</td>
<td>Science</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Business</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Life</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Computers</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Humanities</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Belief</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Arts</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Language</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Applied sciences</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Nature</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Chronology</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Environment</td>
</tr>
<tr>
<td>Mathematics</td>
<td>History</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Education</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Geography</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Agriculture</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Health</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Technology</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Society</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Culture</td>
</tr>
<tr>
<td>Mathematics</td>
<td>People</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Law</td>
</tr>
<tr>
<td>Mathematics</td>
<td>Politics</td>
</tr>
</tbody>
</table>

This chapter explained the category relationship as well as the method to obtain the relationship in figures. The map-like visualization depends on the result to generate a virtual overview map of Wikipedia. The forthcoming chapter will discuss its outcome and their applications in the maintenance tasks of the Wikipedia.
CHAPTER 5: CASE STUDY ON WIKIPEDIA

In order to evaluate the map-like visualization of Wikipedia, we have applied the algorithm to five Wikipedia language editions, namely Danish, Chinese, Swedish, German and English with database dumps in January 2011. All these editions went through the process of data pre-processing, building category tree, similarity calculation and data visualization. This chapter describes the performance of the visualization, and briefly discusses the results and their usages.

5.1 PERFORMANCE

The performance of the entire data analysis and visualization process consists of two major parts: the data manipulation and the visualization. The former operation takes a longer time to digest raw data from the Wikipedia database dump, and saves valuable information as plain text files in the file system. In this way, the visualization module can utilize the same data to generate results with different parameters without the need of running the data analysis step again.

The cosine similarity calculation consumes the most of time during the data analysis step. The calculation is based on the number of articles in a pair of categories. Hence, a straightforward way to implement the calculation (that is used in the first version of the map-like visualization) is to request the number of articles for every pair of categories, which turns out total \(n^2/2\) queries (\(n\) is the number of categories in the Wikipedia). This kind of \(O(n^2)\) algorithm performs badly when the data amount increases. For instance, the English Wikipedia contains approximately 600,000 categories, with every query runs for 0.02 seconds, the whole algorithm would spend 228 years for merely obtaining the number of articles.

In some later version of the algorithm, we optimized the way the algorithm operates. The new version creates counters in the memory to store the number of articles co-assigned by category pairs, and examines each article for increasing counters.
which belong to the assigned categories. This approach makes its running time to $O(n)$ in terms of number of articles. Since not every pair of categories has co-assigned article, we can save memory space with these unused counters and accomplish an acceptable memory usage.

Table 13: Statistics of visualization output

<table>
<thead>
<tr>
<th>Language</th>
<th>Resolution</th>
<th>Width (px)</th>
<th>Height (px)</th>
<th>No. of Categories</th>
<th>Processing Time</th>
<th>Drawing Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese</td>
<td>235 Mpx</td>
<td>15,167</td>
<td>15,556</td>
<td>82,639</td>
<td>75 mins</td>
<td>5 mins</td>
</tr>
<tr>
<td>Danish</td>
<td>198 Mpx</td>
<td>18,065</td>
<td>11,004</td>
<td>19,193</td>
<td>15 mins</td>
<td>2 mins</td>
</tr>
<tr>
<td>English</td>
<td>198 Mpx</td>
<td>14,582</td>
<td>14,756</td>
<td>600,684</td>
<td>300 mins</td>
<td>15 mins</td>
</tr>
<tr>
<td>German</td>
<td>190 Mpx</td>
<td>15,176</td>
<td>12,542</td>
<td>68,677</td>
<td>50 mins</td>
<td>4 mins</td>
</tr>
<tr>
<td>Swedish</td>
<td>119 Mpx</td>
<td>12,881</td>
<td>9,217</td>
<td>82,039</td>
<td>75 mins</td>
<td>5 mins</td>
</tr>
</tbody>
</table>

We used a computer with an Intel Core 2 Duo 3.2GHz CPU and 4GB memory installed. Our program runs on a 64-bit Java J2SE 6.0 Virtual Machine which installed on Windows 7 and uses the included Java AWT library to create its output image. Use the English version of Wikipedia which is the largest one in size as an example, the data manipulation part took approximately 5 hours to process records for over 600,000 categories. The visualization outcome is a 198 mega-pixel PNG image which was created in 15 minutes. 75% of the time is spent by the Java AWT library on writing and compressing to PNG format. Table 13 gives the statistics of the visualization output of different Wikipedia language editions. Despite of the most number of categories found in the English Wikipedia, its output resolution is not the largest among all visualizations. This is because the size of the output depends on the distances among categories instead of the number of categories.

5.2 DISCUSSION ON APPLICATIONS

The following sub-sections highlights parts of the visualization result in different Wikipedias, and illustrates their applications on various scenarios. For the complete result of the visualization, please refer to Appendix A.
5.2.1 UNDERSTANDING CATEGORY COMPOSITION

Contents in the Wikipedia are classified into categories. Understand the category composition helps to recognize the current topic distribution in the Wikipedia, which represents the topics of focuses over existing articles. There is no straightforward method to find out topic distribution currently. Doing so manually would require inspecting many categories and sub-categories one by one. As illustrated in Figure 31, a user needs to firstly browse the main topic category list, then looks up sub-categories in the subsequent screen, and transverses to deeper sub-categories. This approach still does not provide a good overall picture of the categories and content in the Wikipedia.

Our map-like visualization provides a much more easily perceivable category composition. For example Figure 32 shows a highlight of category “Mathematics” in the English Wikipedia. It shows both sub-categories such as “LEMMAS” and “HISTORY OF CALCULUS”, as well as sub-sub-categories such as “Algebra” and “Geometry”. Relative sizes can be easily seen, which reflects different popularity of different topic areas among authors in different colors. For instance, category “Geometry” is shown with two color levels darker than nearby category “Arithmetic”, indicating its larger number of articles. Overall about half of all categories are in the darkest area color, indicating a high level of content maturity, but sub-sub-categories colored in the lightest color can be seen in almost all sub-categories, indicating a need for further content development. This type of overview allows editors to quickly perceive the current content collection of the Wikipedia, such as when identifying large topic areas for potential division into sub-categories, or for identifying relatively under-represented topics that deserve more attention.
Figure 31: The current way to obtain category distribution information

Figure 32: Highlight of category “Mathematics” in the Wikipedia
5.2.2 Displaying Category Relationship

Figure 33: Cluster of related categories “Environment”, “Life” and “Geography” placed in close proximity of one another

Category relationship measures the extent to which different categories in the Wikipedia are related by content. Cosine similarity can be used to numerically denote the relationship. Interesting questions such as “Which categories are more related to Belief?”, or “Are categories Society and History closely related?” can be answered by similarities, but in their numeric form it is difficult to perceive multiple mutual relations, and this becomes more difficult as the number of categories involved increases.

However, when viewing the visualization, proximities clearly express the relationships between categories. An example of this is shown in Figure 33. It shows
the three neighboring topic categories “Environment”, “Life” and “Geography”. The content of these categories is closely related to each other (higher mutual similarity values), which is expressed in the visualization by placing them adjacent to each other. On the other hand, other much less related top-level categories such as “Arts” and “Belief” are placed much further away (not pictured in the figure). In this way, the visualization summarizes the overall relationship of these categories into visual proximity. This form of visualization provides a straightforward way to discover relations among topic areas in the Wikipedia.

5.2.3 OVERVIEW OF MULTIPLE WIKIPEDIAS

A map-like visualization effectively serves as a virtual “world map” of the Wikipedia. It provides a first impression of the overall situation of the Wikipedia, similar to how a real world map gives an overview of the distribution of land and sea, the relative sizes and positions of continents, etc. Visualizations of multiple Wikipedia language editions make comparisons possible, just like using several political maps helps compare aspects of different physical countries.

Table 14: Statistics of the German and English Wikipedia

<table>
<thead>
<tr>
<th></th>
<th>German</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of top-level categories</td>
<td>28</td>
<td>24</td>
</tr>
<tr>
<td>No. of all categories</td>
<td>84,161</td>
<td>602,141</td>
</tr>
<tr>
<td>No. of all articles</td>
<td>1,055,243</td>
<td>3,411,491</td>
</tr>
<tr>
<td>Avg. articles per category</td>
<td>12.5</td>
<td>5.7</td>
</tr>
</tbody>
</table>

Figure 43 and Figure 44 (in Appendix A) give an overview of the German and English Wikipedia editions respectively. The first impression we get from these visualizations is a difference in color: the German Wikipedia is more strongly dominated by dark orange colors than the English Wikipedia and this indicates that the categories contain more content. Given the knowledge of the sizes of these Wikipedias (the English Wikipedia having about three times as many articles as the
German one), this indicates that the category hierarchy in the German Wikipedia is not as finely divided as the English one.

Figures in Table 14 also reflect a similar phenomenon. Although the number of top level categories in the German and English Wikipedias does not differ strongly from one another, the total number of categories varies greatly. In the English Wikipedia there are about 7 times as many categories as in the German one, but only about 3 times as many articles. Correspondingly the average number of articles per category in the German Wikipedia is more than twice that of the English Wikipedia. Clearly the English Wikipedia’s user community favors dividing their content into finer topic sub-divisions, with the effect that each category has relatively fewer articles than in other language Wikipedias, hence the relative lack of the darker orange colors is standing for categories with more article content.

5.2.4 COMPARISON OF TOPIC AREAS IN MULTIPLE WIKIPEDIAS

Besides comparing the overview of the Wikipedia, a map-like visualization can be used to compare the same or similar topic areas across multiple languages. For example, we can compare characteristics of a topic area in different Wikipedia language editions to discover the relative maturity of the same topic across these encyclopedias. A sample comparison with the “Science” category in the Danish, Swedish and Chinese Wikipedia is performed with the figures in Table 15.

The visualization for the comparison is shown in Figure 34 (same scaling ratio applied for comparability). These visualizations firstly display the size of content among the different wikis. The overall area occupied by this category is the largest in the Swedish Wikipedia, followed by the Chinese and Danish ones. This matches the figures shown in the table. Besides, the average number of articles in the Chinese Wikipedia is actually the lowest, although it has more sub-categories under the “Science” category than either of the other two wikis. The colors in the visualization, related to the density of articles in categories, accurately reflect this fact: many areas
in the Danish and the Swedish visualizations are displayed in darker colors, whereas most Chinese sub-category regions are displayed in lighter colors.

Table 15: Statistics of category “Science” in the Danish, Swedish and Chinese Wikipedia

<table>
<thead>
<tr>
<th></th>
<th>Danish</th>
<th>Swedish</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of sub-categories</td>
<td>21</td>
<td>30</td>
<td>34</td>
</tr>
<tr>
<td>No. of sub-sub-categories</td>
<td>65</td>
<td>186</td>
<td>105</td>
</tr>
<tr>
<td>Total no. of sub-categories</td>
<td>86</td>
<td>216</td>
<td>139</td>
</tr>
<tr>
<td>Total no. of articles</td>
<td>2382</td>
<td>7867</td>
<td>1563</td>
</tr>
<tr>
<td>Avg. articles per category</td>
<td>27.7</td>
<td>36.4</td>
<td>11.3</td>
</tr>
</tbody>
</table>

Figure 34: Category “Science” in Wikipedia: (a) Danish (b) Swedish (c) Chinese

5.3 COMPARE WITH OTHER WIKIPEDIA VISUALIZATIONS

Visualization of Wikipedia becomes a new direction of research, thanks to the rapid development of the information visualization field. Many researchers paid their effort
to create novel visualization to allow better understand different aspects in the Wikipedia, ranging from revisions of articles to evolution of the entire Wikipedia. Each of these visualization algorithms has diverse focuses on the information displayed. We compared several existing Wikipedia visualizations to our work. This section will discuss the similarity as well as uniqueness of our work in contrast to other visualizations.

5.3.1 COMPARISON WITH SEMANTIC COVERAGE VISUALIZATION

Holloway et al. present a way to measure semantic coverage of topics through a visualization image as shown in Figure 35 [7]. Dots represent categories in the Wikipedia while different colors showing the topics to which categories related. The color pattern suggests the topic coverage among the categories of Wikipedia.

Figure 35: Highlight of Holloway’s Wikipedia visualization
Categories are positioned by similarities as well, in a way that more related category dots are placed near to each other.

To compare with the map-like visualization, there are a few differences between two visualizations. Firstly, their work renders all categories as dots in the output, unlike ours that the category hierarchy still remains in the visualization. We also use color to represent the sizes of topics which is not easy to measure in their visualization because of the spread out color dots. Finally, large amount of gray dots (i.e. categories without a specific topic associated with) occupy many spaces in the image but distracting the focus to color dots.

5.3.2 **COMPARISON WITH MATRIX VISUALIZATION**

![Figure 36: A multi-level matrix visualization of category hierarchies](image)

A matrix visualization displays mutual relations among the members of a set of items by placing items on the horizontal and vertical axes of a matrix, and then plotting attributes of the relation of a pair of items in a suitable visual representation at the corresponding horizontal/vertical intersection within the matrix. A variation of matrix visualization is applied on Wikipedia in [36]. An approach to combine multiple
matrix visualizations, one for the top level topic categories and another for their sub-categories, is employed to display more information on one screen. Figure 36 shows an example of this visualization. Colored circles serve the purpose to denote the sizes of categories. The larger ones in lighter colors are main topic categories, while smaller in darker colors are sub-categories. Besides, category (or sub-categories) orders are arranged in terms of similarity, in other words, more related categories are placed next to each other.

This visualization type assists data analysis of categories because it is possible to simultaneously explore associations of large numbers of categories and sub-categories without the need for reducing the number of data dimensions. The matrix visualization also provides a general overview of numbers of categories and their sub-categories. It also reveals clusters of category pairs as related ones are adjacent placed.

The map-like visualization presents the information shown in the multi-level matrix visualization as well. We map the category hierarchy to multiple level political regions, such as countries, provinces and counties, in contrast to only two levels in the matrix one, consequently showing more levels at once. Our work also has advantage on illustrating category relationship with proximity in all directions which provides richer bandwidth for showing one-to-many relationships; while the matrix visualization only shows the relationship along the axes, that is limited to put the most related two items adjacent to a category.

5.3.3 COMPARISON WITH RADIAL VISUALIZATION

A radial visualization of Wikipedia takes two levels of categories (i.e. main topic categories and their sub-categories) and places them around a circle. In addition, articles are represented as colored dots, and filled the circle according to their strength of their classification to categories, which locate on the edge of the circle [37]. Besides, each sub-category consists of a gauge that shows the total number of articles within that sub-category. An example of the radial visualization is displayed in Figure 37.
Figure 37: A radial visualization of Wikipedia (reproduced from [37])

From the figure we are able to observe dotted lines with colors are connecting categories across the circle. Figure 38 gives a zoomed-in vision near the edge of the circle. One dot denotes an article in the Wikipedia. Dots may stack up in the same position because of the similar closeness to categories. The author chooses to use different colors to indicate the degree of stacking: the more sparsely spaced green discs represent a single article, the yellow ones up to four overlapping articles, and the red ones in this extract up to 256 articles.
The radial visualization of Wikipedia aims to provide an overview of article distribution as well as basic information such as names and sizes of categories. On the other hand, map-like visualization targets for an overview of semantic coverage in terms of category relationships. Both works have a diverse focus on generating an overview picture for Wikipedia, but they are common in visualizing connections of categories. Map-like visualization puts related categories closer to each other, meaning that the proximities indicate the strength of category similarities. Comparing with the radial visualization, article dots are clustering into lines for connecting categories. This approach clearly shows the bonded category pairs based on articles instead of similarity values.

5.3.4 USER EVALUATION

To evaluate the applications that map-like visualization is suited, as well as receive feedback from users, we carried out a preliminary evaluation as follows. We recruited 28 participants: 19 male, 9 female; 19 students, 9 non-students; having mixed
computer expertise but most self-reporting basic to intermediate knowledge; age range 19-30 with the median age 22. Participants were briefed about the basic knowledge about the Wikipedia category system, then shown and explained three types of visualization for the Simple English Wikipedia: map-like, multi-level matrix (refer to section 5.3.2) and radial (refer to section 5.3.3). After all participants were clear about the meaning of the visualizations, they were presented with seven information tasks that required them to use the visualizations to find answers, whereas these tasks are further classified as three groups: (A) determining the size of a category’s content; (B) discovering relationships among categories; (C) analyzing sub-category distribution inside categories. Exact statements of these seven tasks are listed as follows:

- Task 1: List three categories that contain the most content.
- Task 2: Find the smallest category.
- Task 3: In the smallest category you found in Task 2, find the largest sub-category.
- Task 4: List three categories that are most related to the contents of “Science”.
- Task 5: Find the category that contains the largest number of sub-categories.
- Task 6: Find the sub-category in the “History” category that contains the most content.
- Task 7: Considering content assigned to pairs of categories (a) “Geography” and “People”, and (b) “Literature” and “People”, which of these pairs has the most content?

Table 16: Responses of the user evaluation

<table>
<thead>
<tr>
<th>Group</th>
<th>Tasks</th>
<th>Map-like</th>
<th>Multi-Level Matrix</th>
<th>Radial</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Number</td>
<td>%</td>
<td>Number</td>
</tr>
<tr>
<td>A</td>
<td>Task 1</td>
<td>5</td>
<td>17.9</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Task 2</td>
<td>5</td>
<td>17.9</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Task 3</td>
<td>4</td>
<td>14.3</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>Task 4</td>
<td>12</td>
<td>42.9</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>Task 7</td>
<td>3</td>
<td>10.7</td>
<td>12</td>
</tr>
<tr>
<td>C</td>
<td>Task 5</td>
<td>3</td>
<td>10.7</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Task 6</td>
<td>4</td>
<td>14.3</td>
<td>4</td>
</tr>
</tbody>
</table>
Each visualization method has its strength and weak points in different scenarios. Answers of the respondents therefore reflect the preferences of choosing the visualization in certain tasks. Table 16 shows the evaluation results for the above mentioned tasks for looking up information of Wikipedia categories. From the figure we discover that the map-like visualization was preferred over the other two visualizations with approximately 43% user’s support in Task 4, which is related to spotting related categories. Alternatively, users reported to choose radial visualization than others for group A and C tasks, suggesting that this type of visualization are relatively more suitable for these information tasks. This suggests that the map-like visualization has an advantage over the other two visualizations in the scenario of identifying the related categories.

Besides, respondents were also asked to provide general comments on the visualizations, and many commented that the visualizations were useful for looking up information about the Wikipedia. One of them commented “I like the map-like [visualization]”, and another one remarked “Matrix: easiest to understand; Map: I like the style of showing information in this way; Radial: seems difficult to understand”. These responses clearly show that personal preferences play an important role in the choice of visualizations.
CHAPTER 6: CONCLUSION

This thesis presented a novel type of visualization – map-like visualization for the Wikipedia, which mainly aims to provide a virtual overview map for understanding the semantic coverage in the Wikipedia. Before this research was carried out, there was no straightforward method to easily obtain such information. In addition, community-written articles and categories are linked without a uniform standard, thus making the data structure full of idiosyncrasies and complicated to analyze. We proposed solutions to these problems such as transforming the category system into a simple tree structure and cleaning up the category tree. We also adapted the cosine similarity for measuring the relationship of categories in terms of number of articles they co-assigned with.

In the literature review we studied basic concepts of information visualization, including hierarchical data visualizations, force-directed graph layout algorithms and overlap removal methods. Then we reviewed past researches of Wikipedia visualization and their approaches to process Wikipedia data. We further referred some map-like visualization related researches and improved the algorithm for the use in the Wikipedia’s dataset, as well as applications of the cosine similarity formula on categories and articles.

We applied our experimental methodology to the Simple English Wikipedia for evaluating the outcome. Meanwhile, a group of users were surveyed to get feedback on the map-like visualization. Finally, the map-like visualization algorithm ran with the data from the Danish, Chinese, Swedish, German, and English Wikipedia. The corresponding results have many applications and comparability Wikipedias across various languages, as mentioned in the chapter 5. Visualization of this type has the potential to reveal in a readily perceivable way much information contained within other wiki collections, even other hierarchical data sources.
6.1 CONTRIBUTION

Our research contributes knowledge to different areas, such as data analysis and information visualization. Brief descriptions of contributions related to this work are listed as follows:

- Introduced an algorithm for transforming Wikipedia category graph into a tree.
- Created a map-like visualization that can help understanding the Wikipedia.
- Proposed a new way of representing information in a map-like approach.

In details, we firstly proposed an algorithm to transform a Wikipedia category graph into a simple tree structure, especially using cosine similarity to determine the most appropriate parent to keep in case of multiple parents exist. Cosine similarity is also applied to measure the category relationship, which is based on the number of articles assigned with categories in concern at the same time. Moreover, we have incorporated concepts of the map-like visualization in our work, and modify them to be usable for Wikipedia’s data. Such visualization helps people to understand abstract and complicated aspects of Wikipedia, in practical like viewing topic distribution and performing administrative tasks. On the other hand, map-like visualization on Wikipedia suggests a new direction to present hierarchical data in a 2-dimensional output. A preliminary evaluation shows that this kind of visualization has a potential to assist users to explore relationships of hierarchy information in an easy understandable approach.

6.2 FUTURE WORK

This research has still many rooms for improving in several directions. First, performance is of critical importance when processing large amount of data, thus we are looking for ways to optimize the running time and memory usage of the algorithm. Moreover, currently we are mapping categories and articles into political regions in the virtual Wikipedia map. We are considering adding more elements in to the map-like visualization, such as roads between cities representing significant linkages between the corresponding articles. This will enrich our visualization by increasing
the information communicated by it. Besides, an interactive version of the map-like visualization can be another enhancement to the project.

The current visualization is outputted as a static image, which may be difficult to navigate and explore the information. An interactive version is designed to display the overview initially and responds to user’s command like zooming and panning to show the part of information requested. Finally, the presentation of the visualization is another concentration in the future. Based on the users’ feedback, we may re-design the shapes, colors, lines or other elements in the result in order to increase the efficiency and the readability of the visualization, ultimately helping users to better understanding the composition of the Wikipedia.
ASSOCIATED PUBLICATIONS


REFERENCES


APPENDIX A: RESULTS OF WIKIPEDIA

MAP-LIKE VISUALIZATION

Our work has been applied to six different languages of Wikipedia, which are Simple English, Danish, Chinese, Swedish, German, and English (in ascending order of their sizes) respectively. In this appendix, we present static snapshots of the map-like visualization of these editions of the Wikipedia.

Figure 39: Overview map of the Simple English Wikipedia
Figure 40: Overview map of the Danish Wikipedia

Figure 41: Overview map of the Chinese Wikipedia
Figure 42: Overview map of the Swedish Wikipedia
Figure 43: Overview map of the German Wikipedia
Figure 44: Overview map of the English Wikipedia
This appendix is to explain the steps for importing the Wikipedia database dumps into a database. Downloaded SQL database dumps are compressed with the gzip format, and gunzip is used to expand it. Each language edition requires an entire blank database instance to import. We start the MySQL console and execute the statements in Table 17 to initiate the import. In the example, enwiki represents the name of the newly created database; and it is recommend using the language to name it. page.sql and categorylinks.sql are the files extracted from the compressed database dump.

Table 17: SQL commands for importing database dumps

| CREATE DATABASE enwiki;  
| USE enwiki;  
| SOURCE page.sql;  
| SOURCE categorylinks.sql; |
VITA

PANG Cheong Iao (Patrick)

University of Macau

2011

I graduated in the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau in 2007 and obtained a Bachelor of Science in Software Engineering degree. My research interest includes information visualization, data analysis and collaborative software systems. I am currently working as an Information Technology Technician in the University of Macau. The main job duties consist of information system development, system design and analysis, writing system documentations and liaison with users.