Portland State University

PDXScholar

Dissertations and Theses

Dissertations and Theses

3-26-2020

Multiple Diagram Navigation

Hisham Benotman

Portland State University

Follow this and additional works at: https://pdxscholar.library.pdx.edu/open_access_etds

Part of the [Theory and Algorithms Commons](#)

Let us know how access to this document benefits you.

Recommended Citation

Benotman, Hisham, "Multiple Diagram Navigation" (2020). *Dissertations and Theses*. Paper 5426.

<https://doi.org/10.15760/etd.7299>

This Dissertation is brought to you for free and open access. It has been accepted for inclusion in Dissertations and Theses by an authorized administrator of PDXScholar. Please contact us if we can make this document more accessible: pdxscholar@pdx.edu.

Multiple Diagram Navigation

by

Hisham Benotman

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

Doctor of Philosophy
in
Computer Science

Dissertation Committee:

David Maier, Chair

Lois Delcambre

Jeffrey Ovall

Kristin Tufte

Portland State University
2020

© 2020 Hisham Benotman

Abstract

Domain novices learning about a new subject can struggle to find their way in large collections. Typical searching and browsing tools are better utilized if users know what to search for or browse to. In this dissertation, we present Multiple Diagram Navigation (MDN) to assist domain novices by providing multiple overviews of the content matter using multiple diagrams. Rather than relying on specific types of visualizations, MDN superimposes any type of diagram or map over a collection of documents, allowing content providers to reveal interesting perspectives of their content. Domain novices can navigate through the content in an exploratory way using three types of queries (navigation): diagram to content (D2C), diagram to diagram (D2D), and content to diagram (C2D).

To evaluate the MDN user interface, we conducted a user study, which showed that users found MDN useful and easy to use in exploratory-navigation scenarios. Encouraged by these positive results, we extended the functionality of MDN to provide a ranking of collection documents for D2C queries (expressed by a selected diagram concept). We studied different elements of the ranking process. As a case study, we targeted our research towards the Wikipedia collection. With the goal of studying ranking in different types of diagrams, we introduced two diagram models: the Items-and-Attributes model and the Universal model. We also studied two ranking algorithms: Personalized PageRank (PPR), an algorithm used in similar applications;

and Greedy Energy Spreading (GES), an algorithm that we designed. We also studied different approaches to computing rankings for C2D queries.

Our results show encouraging performance on the ranking of D2C and C2D queries. For example, in an experiment targeting diagrams conforming to the Items-and-Attributes model, results showed reasonably high similarity between a diagram concept selected by the user and the top-ten-ranked pages. We also found that diagrams had a strong influence on D2C ranking, which yielded a ranking reflecting the aspect presented by the diagram. In C2D-query ranking, GES was able to rank the most related concept in the diagram to a Wikipedia page selected by the user in the top five or six positions on average (in diagrams with 50 elements).

In our tuning for the two studied algorithms, we observed that GES performs slightly better than PPR in some aspects of D2C and C2D ranking. We also noticed differences between MDN and similar applications on the optimal settings for Personalized PageRank. Our configuration of the Wikipedia graph revealed that including (or excluding) Wikipedia categories and article-template hyperlinks have a high influence on the tested algorithms. Also, compared to related work that included only reciprocal links (where a pair of pages link to each other) in the graph, we report better performance when also including low-weight non-reciprocal links.

Acknowledgements

I would like to express my sincere gratitude to my advisor Professor David Maier for his continuous support during my Ph.D. study. I have learned a lot from his passion for research and his attention to details. His constructive feedback has always motivated me to become a better researcher.

I would also like to thank my previous advisor Professor Lois Delcambre for her support. Professor Delcambre provided valuable teaching and guidance, which were much needed in the beginning of my Ph.D. journey. I also appreciate the valuable feedback she provided for my thesis proposal.

I am also grateful to the other members of my committee: Professor Kristin Tufte and Professor Jeffery Ovall. I would like to thank them for their understanding and feedback.

My sincere thanks go to my colleagues at the Portland State University datalab, including Abdussalam Alawini (alumnus), Scott Brittel, Basem Elazzabi, Christopher Giossi, Patrick Leyshock (alumnus), and Veronica Megler (alumuna) for their feedback and support. I would also like to thank Melody Bandlamudi and Angel Martinez Villalta for volunteering in my research project.

My research was supported by the Libyan Ministry of Higher Education and Scientific Research through the Canadian Bureau for International Education (CBIE). I would like to acknowledge support for my work from the Intel Science and Technology

Center for Big Data, through a gift from the University Industry Research Corporation. I would also like to thank the Department of Computer Science at Portland State University for their support through the Richard Kieburtz Memorial Graduate Fellowship. I also acknowledge travel support during the early stages of this research from NSF grant number 1250340.

Table of Contents

Abstract	i
Acknowledgements	iii
List of Tables	ix
List of Figures	xii
1 Introduction	1
1.1 MDN Visual Query Types	6
1.2 Basic versus Extended MDN	8
1.3 Scope of This Research	9
1.4 Contributions	9
1.5 Dissertation Organization	12
2 Basic MDN	13
2.1 BMDN Components	13
2.2 BMDN Visual Query Processing	15
2.3 BMDN Implementation	16
2.4 BMDN Evaluation	17
2.4.1 Methods	18
2.4.2 Results and Discussion	19
2.5 Conclusions	28
3 Extended MDN	30
3.1 The Wikipedia Collection	34

3.2	The Items and Attributes (I&A) Diagram Model	36
3.2.1	Model Guidelines and Graph Construction	39
3.2.2	I&A-Model Support for D2C Queries	45
3.3	The Universal Diagram Model	46
3.3.1	Directed Unweighted Graph for Universal-Model Diagrams and Wikipedia Pages	53
3.3.2	Undirected Weighted Graph for Universal-Model Diagrams and Wikipedia Pages	54
3.3.3	Universal-Model Support for D2C Queries.....	55
4	Existing Algorithms: Personalized PageRank (PPR)	57
4.1	The PageRank Algorithm	57
4.2	The Personalized PageRank Algorithm	61
4.3	Using PPR for MDN D2C Queries	63
4.4	Monte Carlo Method for PageRank	66
5	The Greedy Energy Spreading Algorithm	69
5.1	Energy Spreading	69
5.2	The Greedy Energy Spreading (GES) Algorithm	70
5.2.1	GES-Diagram: Propagating Energy in the Diagram	70
5.2.2	GES-Wikipedia: Propagating Energy in Wikipedia	72
6	Evaluation with I&A-Model Diagrams	80
6.1	Methods.....	82
6.1.1	Tuning the PPR Algorithm.....	86
6.1.2	Tuning the GES Algorithm.....	87
6.2	Results and Discussion.....	89
6.3	Discussion Summary	95
7	Evaluation with Universal-Model Diagrams	99
7.1	Methods	101
7.1.1	Preparation of the Undirected Weighted Graphs	103
7.1.2	Tuning the PPR and GES Algorithms	107
7.2	PPR Tuning Results	110

7.3	GES Tuning Results	113
7.4	Comparison of GES and PPR Algorithms.....	119
7.5	Discussion Summary	126
8	Evaluating Approaches for C2D-Query Computation	130
8.1	Approaches for C2D-Query Computation	131
8.2	Research Questions	133
8.3	Methods	134
8.4	Tuning PPR-C2D	138
8.5	A Comparison of Five Approaches to Compute C2D Queries	139
8.6	Discussion Summary	144
9	Related Work	146
9.1	Exploratory Search Tools	146
9.2	Induced Visualizations	148
9.3	Entity Ranking and Recommendation	149
9.4	Exploratory Search Tools For Wikipedia	151
9.5	Word Relatedness and Named Entity Disambiguation	155
9.6	Superimposed Information	158
9.7	The Personalized PageRank Algorithm	160
9.8	The Energy-Spreading Algorithm	164
9.9	Summary.....	166
10	Conclusions, General Observations, and Future Work	167
10.1	Conclusions	167
10.2	General Observations.....	171
10.3	Future Work.....	174
10.3.1	Extending Diagram Navigation to Collections Other than Wikipedia	174
10.3.2	Comparing Diagram Navigation with Other Exploratory Search Tools.	175
10.3.3	Automating the Task of Connecting Diagrams to Wikipedia.	175
10.3.4	Multiple Selected Elements or Pages in D2C and C2D Queries....	176
10.3.5	Studying C2D Queries in More Detail.....	176
10.3.6	Measuring the Quality of Diagrams for MDN Ranking	177

10.3.7 Computing the Ranking of D2D Queries	177
Bibliography	179
Appendices	188
Appendix A: Exploratory Tasks	188
Appendix B: Labeling Criteria	189
Appendix C: Diagrams Used in the Experiments	194

List of Tables

2.1	The subjects' level of experience before and after the user study.	20
2.2	Questions (from TAM [36]) about usefulness and usability of MDN with mean and standard deviation for the ten subjects.....	21
2.3	Questions about MDN visual queries: Diagram to Content (D2C), Content to Diagram (C2D), and Diagram to Diagram (D2D) with mean and standard deviation for the ten subjects	22
2.4	Averages of subjects' hovering on different groups of visual elements (color scale is separate for the rightmost column). Clicks are shown as asterisks. Low-experience subjects (the top two rows) tended to spend more time on the diagrams (hover time) than high-experience subjects (the bottom two rows). Also, subjects focused their hovering and clicking on different groups of concepts or different sides of the diagram.	25
2.5	Types of visual queries used to find keyword elements. While high and medium experienced subjects used more C2D queries, low-experience subjects depended on hovering on the diagrams	26
6.1	The 20 diagrams we used in the I&A-model experiment. The diagram column indicates the diagram's attributes. We show the size (number of item pages) of the Target Universe (TU) Collection. For the best setting in both the PPR and the GES algorithms, we show the average rating (Rtng) of the top ten pages (1 is best and 3 is worst) in all queries in each diagram. Lighter background colors highlight the best Rtngs. In general, GES was slightly better than PPR in all diagrams with Rtng difference > 0.15 in 7 diagrams. The rightmost column: given the top ten pages $p_1, p_2, \dots, p_{10}$ when selecting an element e in diagram D_1 shown on the second column, we show the average position of the pages p_1 to p_{10} when selecting e on diagram D_2 , where D_1 and D_2 present different aspects of the same domain. The high averages indicate that two diagrams showing two aspects of the same domain produce different rankings.....	83

6.2	Tuning results for 45 PPR settings based on 120 D2C queries issued from 12 diagrams. We show the average similarity rating (Rtng) for the top ten pages in all queries, where $1 \leq \text{Rtng} \leq 3$ (lower values are better). The performance of the PPR algorithm was better with a high damping factor ($d \geq 0.45$), indicating the importance of increasing the effect of diagrams on the ranking process. Similar to GES, PPR achieved the best performance with keeping nonreciprocal links but giving reciprocal links more importance (nRcp values: 0.1, 0.06, and 0.03).	89
6.3	Tuning results for 36 GES settings based on 120 D2C queries issued from 12 diagrams. We show and sort by average similarity rating (Rtng) for the top ten pages in all queries, where $1 \leq \text{Rtng} \leq 3$. The GES parameters are series (Srs), non-reciprocal-link conductance (nRcp), and greediness factor (F). A moderate decrease of nRcp (0.7 or 0.4) had the best performance (Rtng close to 1) compared to removing nonreciprocal links (nRcp = 0) or giving the same weight to reciprocal and noreciprocal links (nRcp = 1)	90
6.4	The number of processed pages (pages allowed to send energy) in L2 in four different settings including the best two settings in Table 6.3. Comparing (2, 0.7, 0.9) and (2, 0.7, 0.75) as well as (3, 0.4, 0.75) and (3, 0.4, 0.9) shows that switching from greediness factor 0.75 to 0.9 reduced the number of processed pages to about 60% without harming the average rating.	92
6.5	For the best settings in the PPR and GES algorithms, we show the average similarity rating for the top ten pages in all queries issued from three data sets: the first 12 diagrams in table 6.1, the last 8 diagrams in the same table, and all 20 diagrams. Overall, the GES algorithm (2, 0.7, 0.9) is slightly better than the PPR algorithm. Using PPR instead of GES would decrease similarity (increase rating by 1) for about one and a half page in the top-ten pages.	92
6.6	When selecting the ‘Buffalo Mozzarella’ cheese in the cheese-by-country diagram, we show the top ten pages returned by each algorithm (GES on the left and PPR on the right). Compared to GES, PPR had greater tendency to elevate the rank of high-degree pages. Those pages are usually important concepts in the domain such as ‘Milk’ and ‘Cheese’ or pages that were part of the noise in the TU collection such as page # 9.	93
6.7	Top ten Wikipedia pages with similarity ratings when selecting the ‘Dinka’ language on the language forest (left) vs. selecting the same language on the language map (right). As with all diagram pairs, pages were quite similar to the selected element (‘Dinka’) based on the used diagram; languages in the same language family (left) vs. languages spoken in the same country (right).	94
7.1	The 20 tables we used for Universal-Model-Diagrams Investigation.	102

7.2	Tested parameters for the PPR algorithm. We show best values in bold font.	108
7.3	Tested parameters for the GES algorithm. We show best values in bold font.	109
7.4	A comparison between the best greediness setting (row 1) and two alternatives. For each setting, we show the (mean, median) number of nodes allowed to propagate energy in each stage.	118
7.5	Ranked lists returned from PPR and GES when ‘Cappuccino’ is selected in the coffee diagram. L1 pages (pages connected to the diagram) are in bold italic font. Pages less relevant to the selected element are in shaded background.....	121
7.6	In a D2C-query’s top 100 ranked pages, we compute the percentage of pages that are L1 pages in each 10-position interval. We show the average percentages of all 200 D2C queries. PPR has more tendency to rank L1 pages in the top compared to GES.	122
7.7	Performance of GES and PPR in each diagram (diagrams used for tuning have shaded background). GES and PPR were robust to all graph classes in the Universal Model. Bad performance in some diagrams (e.g., WW2 Timeline and Data Science) was likely due to low relatedness between a D2C selected element and elements in its neighborhood (i.e., close elements acted as noise in the ranking process).	125

List of Figures

1.1	The Attack Graph (AG) [28], which organizes threats (in blue) and goals (in red) in different attack scenarios, presents a process or how-to overview for the subject of information-security threats.	3
1.2	An edited excerpt of the threats tree diagram [59]. Compared to the attack graph in Figure 1.1, the threats tree (TT) presents a different perspective by organizing information security threats in a hierarchical structure. Orange background is used for each pair of threats that are subtypes of each other (i.e., there are a pair of arrows in each direction).	4
1.3	An MDN prototype including the Attack Graph, the Threat Tree (only a small part is shown at the bottom left), and a Wikipedia page on the right. This screenshot of MDN shows a content to diagram (C2D) query where a user visiting the Wikipedia page of ‘Phishing’ results in MDN highlighting the phishing element (with yellow background) in the Attack Graph.	5
1.4	Part A: a Diagram-to-Diagram (D2D) visual query. A user selects Ransomware in the Attack Graph; MDN highlights Trojan Horse, a related element in the Threat Tree. Part B: a diagram to content (D2C) visual query. A user selects elements in the diagrams to get a list of related Wikipedia pages on the right.	7
2.1	An overview of the components of a Basic MDN navigation system	14
2.2	Preferred Number and Choice of Diagrams. Part A: Average response for ten subjects. Part B: All responses grouped to: two subjects on the left side favoring the AG over two diagrams, five subjects with equal ratings, and three subjects favoring two diagrams over only the AG. ...	23
2.3	Part A: Heat map of total hovers for all subjects in session one (darker elements received more hovers) (numbers point to keyword elements mentioned in the browsing tasks). Part B: total hovers for all subjects in session two. Part C: total clicks for all subjects in session one. For top ten clicked elements, a pair shows # of clicks and # of users who clicked on the element. Blue/Red elements were not clicked.	24

3.1	A: a Diagram-to-Content (D2C) query in Extended MDN (EMDN). A user clicks ‘Dinka’ language in the World Languages Forest Diagram [45]; MDN displays a ranked list of related Wikipedia pages. B: Previously used connections between diagrams and Wikipedia in BMDN in dashed lines. Additional internal diagram and Wikipedia connections used in EMDN in solid lines.	31
3.2	An example Wikipedia article about the Cartoon Network TV Channel. The information box, displayed in the top right, contains important information and hyperlinks. Multiple templates and categories reflecting the multiple aspects of the article are displayed in the bottom. Templates usually have a multitude of hyperlinks.	35
3.3	Representing the I&A Model using the notation of the entity-relationship model.	37
3.4	An excerpt of a subway map for data platforms [7]. We model data platforms as items and features (some shown at the bottom right) as attributes. An attribute subway line connects all items that have that attribute.	38
3.5	An excerpt of a ring-like diagram for candy bars [68]. We model candy bars as items and ingredients as attributes. Items and related attributes are connected with lines.	39
3.6	Part of the graph of the language forest diagram. We model items and attributes diagrams as a bipartite graph where an item can connect to multiple attributes.	40
3.7	A partial graph for a single D2C query including the selected element ‘Dinka’ and its related attributes ‘Eastern-Sudanic’ and ‘Nilo-Saharan’. The target universe node TU connects the selected element to all Wikipedia pages (items) harvested from related categories and List_of pages.	43
3.8	Part of the graph resulting from representing the language forest as a universal model diagram. We make sure that all nodes in the graph are connected by adding edges to a central node <i>C</i>	47
3.9	Part A: An excerpt of World World 2 timeline [88]. The diagram is a sequence of concepts where close concepts are more closely related in time than the far ones. Part B: An excerpt of a diagram [91] showing all bridges in Portland, Oregon. The diagram has a similar organization to the timeline, arranging concepts in a sequence, but based on geographic location.	48
3.10	A part of the path graph that represents the elements in the World War II timeline shown in Figure 3.9.A.	49
3.11	An excerpt of the JavaScript libraries diagram [107] that we represent using a tree graph. The diagram is a mind map. Mind maps usually arrange concepts in a tree.	50

3.12	An excerpt of the coffee diagram [3]. The diagram contains distinct groups of related concepts, such as types of coffee beans, types of coffee drinks, and types of coffee pots and machines. Therefore, we modeled the diagram as a tree graph as shown in Figure 3.13 where each group has a separate parent.	50
3.13	A part of a tree graph that represents the coffee diagram where each group of related concepts has a separate parent.	51
3.14	An excerpt of the programming languages diagram [97]. This diagram is represented using a generic graph.	51
3.15	An excerpt of the TV networks diagram [30]. This diagram is represented using a generic graph. The AMC network part is not connected to the rest of the diagram. Similar to Figure 3.8, we make the graph connected by adding a central node C that has a 2-edge path to every node in the graph.	52
3.16	Part A: The modeling of a universal-model diagram and Wikipedia pages using a directed unweighted graph. Part B: The modeling of a universal-model diagram and a subset of Wikipedia pages (the set R) using an undirected weighted graph. Note that we include only edges where both pages connected by the edge are in R (e.g., edge connecting pages X and Y is not included in the graph since Y is not in the set R).	54
4.1	Part A: An example graph. Part B: Starting with equal scores, we compute PageRank algorithm for 14 iterations. Page scores converge in the last few iterations.	59
4.2	The first line shows a transition matrix M of a random surfer on the graph in Figure 4.1 using damping factor $d = 0.15$. M combines a probability $(d - 1)$ of following one of the edges on the graph (matrix A) and a probability d of jumping to a random page (matrix B). The second line shows one iteration of PageRank where we multiply M by the initial PageRank vector P_0 to get the PageRank vector P_1 which shows the probability of the random surfer to be at each page after one step of surfing the graph.	60
4.3	Part A: An example diagram with three elements connected to four pages in Wikipedia. We use the term <i>node</i> to denote both elements and pages. We show a ranked list of diagram elements and Wikipedia pages when using PPR with a damping factor $d = 0.15$. Part B: Another example where a page with a high degree is ranked higher than pages connected directly to the diagram when setting $d = 0.15$. However, increasing d to 0.5 boosts Pages 1 and 2, which are closer to the diagram. Part C: Showing that a graph node can be also affected by the degree of nodes it connects to. Page 1 is ranked higher than Pages 2 and 3 even though each one of the three pages has a similar path to the selected element in the diagram.	64

5.1	Different distributions of energy for diagram elements based on their distance (or shortest path) from a selected element in a diagram-to-content (D2C) query. Dist1 has a sharp decay in energy as we travel further from the selected element compared to the gradual decay in Dist2 and Dist3. Dist4 starts with a gradual decay in the beginning, and then decreases sharply. Dist5 assigns the same energy to all diagram elements regardless of their distance.	72
5.2	Different series used to compute and tune Series Adjusted Summation (SAS).	74
5.3	An example of spreading energy using GES-Diagram and GES-WIKIPEDIA. We show energies (scores) next to nodes for diagram elements and <i>L1</i> pages. For <i>L2</i> and <i>L3</i> Wikipedia pages, we show how their received energies (REs) and scores change in each stage of energy spreading.	78
6.1	A graph of a single D2C query including the selected element ‘Dinka’ and its related attributes. The target universe node TU connects the selected element to all Wikipedia pages (items) harvested from related categories and List_of pages.	84
7.1	Part A: The modeling of a universal-model diagram and Wikipedia pages using a directed unweighted graph. Part B: The modeling of a universal-model diagram (the set D) and a subset of Wikipedia pages (the set R) using an undirected weighted graph. Note that we include only edges where both nodes connected by the edge are in $D \cup R$ (e.g., edge connecting pages X and Y is not included in the graph since Y is not in the set $D \cup R$).	104
7.2	Best two settings of PPR and GES on the directed unweighted graph using a data set of 120 D2C queries. We show the mean rank of the target page at different k values.	106
7.3	For the best setting in each of the four graph configurations listed in Table 7.2, we show the mean and median of target-page ranks at different k values. In the mean’s top left figure, we show a zoomed-in view for k values up to 107. The best setting (0.65, TT, 0.5) performs close to (0.75, FT, 0.5) and (0.45, FF, 0.1), but (0.65, TT, 0.5) is considered better given that it works with a larger graph that includes both articles and categories.	111
7.4	Mean and median rank of target pages at different k values for the best setting for each non-reciprocal link weight listed in Table 7.2. PPR achieves the best performance with keeping low-weight non-reciprocal links as the setting (0.65, TT, 0.5) achieves best mean rank @ $k = 108$. ..	112
7.5	Mean and median rank of target pages at different k values for the best PPR setting (0.65, TT, 0.5) and all settings that used 0.15 as a damping factor (similar to related work).	113
7.6	Testing GES with four graph configurations. The algorithm performs the best with the TF graph which includes Wikipedia articles and categories and excludes hyperlinks found in article templates.	114

7.7	Testing GES with three weights for non-reciprocal links in the TF graph. GES achieved best performance by giving non-reciprocal links about half the weight (0.4) of reciprocal links.	115
7.8	Among the tested six series, S_6 was the best choice with a mean rank equal to 83 @ $k = 108$ and overall median equal to 17 @ $k = 116$	116
7.9	The performance of four energy distribution arrays (used to propagate energy among diagram elements). Dist1 (1 for short) achieves the best performance by giving more weight to diagram elements close to the selected element (see the sharp decline of Dist1 in Figure 5.1)	117
7.10	Comparing different greediness-factor combinations for (F_1, F_2, F_3) . High greediness in the ‘2–2’ and ‘2–3’ stages ($F_2 = 0.9$) is better than both lower greediness ($F_2 = 0.75$) and extremely high greediness ($F_2 = 1.1$), which is equivalent to canceling energy propagation in L2 and L3.	118
7.11	Mean and median rank of target pages at different k values for the best settings of GES and PPR.	120
7.12	After removing L1 pages from D2C ranked lists, we show mean and median rank of target pages at different k values for the best PPR and GES settings. GES and PPR had similar performance when L1 pages are not considered in the evaluation.	123
7.13	Distribution of 200 target-page ranks (L1 pages are removed) for PPR and GES best settings. GES and PPR had similar performance.	124
8.1	An example graph with the page xp selected in a C2D query.	132
8.2	An example graph with the page ap selected in a C2D query. After disconnecting ap from its related diagram element a , we can evaluate a C2D ranking with the help of the structure provided by the diagram (e.g., we expect a to be ranked in the top and f in the bottom)	134
8.3	The best and worst setting for C2D-PPR (i.e., running PPR from a selected page). The low damping factor value 0.15 performed better from the high one 0.85 (opposite behaviour from the one we saw in D2C queries).	139
8.4	Performance of five approaches of computing C2D queries. GES-D2C-Rank was very close to C2D-PPR indicating that computing C2D queries using D2C-query ranked lists is reasonable.	140
8.5	Performance of five approaches of computing C2D queries in ranking the most related element in the diagram. GES-D2C-Rank clearly outperforms C2D-PPR. GES-D2C-Rank is also better than PPR-D2C-Rank (difference is about 5.65 positions when considering MNAPC and 4.36 positions in MDAPC).	141
8.6	An example C2D query where ‘CBS News’ is the selected page. We highlight the top elements with yellow ovals and red outlines. We also show the rank of each element. Even though ‘ABC News’, which is not related to the selected page, was ranked in the top five, the user can easily spot this ranking error and realize that most related pages are close to the CBS group.	142

9.1	Related entities to ‘Portland State University’ (displayed in the bottom right) ranked by the Google search engine.	150
9.2	The Wikipedia Faceted Search [55].	152
9.3	Facetedpedia user interface after searching for ‘computer scientist’ articles [6].	154
9.4	A visualization of Wikipedia categories [23]	155

1 Introduction

Navigating collections of information resources such as website pages or digital-library documents has been widely supported with traditional information-retrieval tools such as keyword searching and hyperlink and menu browsing. However, with a vague information need and little domain knowledge, a domain novice may struggle to find the right search keywords or click the appropriate hyperlink(s) with such tools. For example, a domain novice may struggle to expand their knowledge about information security given a basic concept such as “computer viruses” or “phishing attacks”. Large collections of information resources about information security can include diverse content targeting different audiences, such as ordinary users, developers, network and systems professionals, and digital-forensics practitioners. Domain novices may spend a long time iteratively searching and browsing to find content that is suitable for their needs. To tackle this information-seeking scenario, known as *exploratory search* [75], the Information Retrieval (IR) and Computer Human Interaction (CHI) research communities have joined efforts to provide interactive IR solutions.

As opposed to basic lookup searches, Marchionini [75] characterized exploratory search as pertaining to learning and investigating processes. Lookup searches are suitable for fact-finding information needs such as “What is the weather today?”, “Who is John von Neumann?”, or “Where is Crater Lake located?” Learning and investigative searches are more complex and require analysis and comparison of multiple

information resources [75]. Compared to lookup searches, learning and investigative searches typically target broader topics or subjects, take longer periods of time, and require a higher level of cognitive processing [75]. Examples of learning and investigative searches include learning about a new subject such as database management systems, information security, or machine learning. Shopping or exploring unknown commodities or services can also show characteristics of learning and investigative searches. Consider, for example, a tourist looking for attractions to visit at a destination that she has never visited before, or a novice computer user selecting a security software product for his personal computer. Both users need to learn about a new domain and evaluate multiple options.

To support exploratory searches, researchers have used techniques from IR and CHI to provide interactive IR user interfaces. Marchionini [75] listed some examples such as hierarchical menus, query by example (QBE), and faceted metadata interfaces. Recommending content in large repositories such as YouTube and Amazon is also a popular approach. Contrary to typing keywords when doing lookup searches, these examples show the effectiveness of allowing users to select from given options [75]. Other interfaces show overviews of the content to help users discover existing concepts or topics and the relationships among them [75]. A major body of research (discussed in Chapter 9) has used this approach with collections of documents to provide graphical overviews such as word clouds, timelines, and concept maps. These overviews can help users navigate to desired or interesting parts of the collection. Such overviews are visualizations dynamically induced from the underlying documents and document metadata.

Induced visualizations are restricted by the available types of visualization tools and the nature of the collection and the data that can be extracted from it. In this

Figure 1.1: The Attack Graph (AG) [28], which organizes threats (in blue) and goals (in red) in different attack scenarios, presents a process or how-to overview for the subject of information-security threats.

work, in contrast, we superimpose existing diagrams over a collection of documents or web pages. Our approach aims to broaden the choices of graphical overviews by including any type of diagram or map (i.e., any visual structure saved in any image format). This flexibility can be helpful in revealing interesting and multiple perspectives for a collection. For example, the Attack Graph (AG) shown in Figure 1.1 [28], illustrating different scenarios for attacking a computer system, provides a process or a “how-to” aspect for the domain of information-security threats. The Threat Tree (TT), shown in Figure 1.2, presents a different perspective for the domain: a hierarchical classification of computer-security threats. We expect that using such diagrams to provide interactive navigation to related content can be helpful to domain novices.

Diagrams can show aspects of a domain that no content-visualization tools can

Figure 1.2: An edited excerpt of the threats tree diagram [59]. Compared to the attack graph in Figure 1.1, the threats tree (TT) presents a different perspective by organizing information security threats in a hierarchical structure. Orange background is used for each pair of threats that are subtypes of each other (i.e., there are a pair of arrows in each direction).

create. For example, for an information-security collection, the available visualization tools (that we know of) that can provide an overview of the content are word clouds and map-based visualizations (an example map-based visualization is shown in the Chapter 9). While these visualizations can be useful to discover the popular keywords or topics in the collection, they cannot show aspects of the domain such as the ones presented by the AG and the TT diagrams. Diagrams can also show the point of view of a domain expert that can guide domain novices in their exploration. While visualizations rely on the collection content and structure to create an overview, domain experts may organize concepts using a different structure or layout than the one presented by the collection.

We propose Multiple Diagram Navigation (MDN), a visual navigation system that allows content providers to superimpose multiple diagrams over a collection of information resources. MDN relies on relatively few manual connections between concepts in diagrams and related documents in the collection. Also, MDN uses both

Figure 1.3: An MDN prototype including the Attack Graph, the Threat Tree (only a small part is shown at the bottom left), and a Wikipedia page on the right. This screenshot of MDN shows a content to diagram (C2D) query where a user visiting the Wikipedia page of 'Phishing' results in MDN highlighting the phishing element (with yellow background) in the Attack Graph.

connections between concepts in the diagram and connections or relationships between documents in the collection. Using all these connections together, MDN can access all documents in the collection. By coordinating multiple diagrams, MDN provides three types of *visual queries*: *diagram to content* (D2C), *diagram to diagram* (D2D), and *content to diagram* (C2D). These capabilities can facilitate domain exploration by allowing users to smoothly switch among seeing an overview by looking at a diagram, seeing a different point of view by changing to a different diagram, and reading content after browsing to a page or a document. Figure 1.3 shows an MDN

prototype that includes the AG, TT, and related Wikipedia pages.

1.1 MDN Visual Query Types

We explain a key feature of MDN, the MDN queries (or MDN visual queries). We use a motivational example to show how a domain novice might use the three types of MDN queries. Consider Tom, a student with minimal knowledge of information security. With the increase in university warnings about information security threats such as phishing, keyloggers, and identity theft, Tom wants to explore the subject of information security but he is confronted with a large number of topics, courses, and websites targeting different types of users. Tom starts by searching for ‘Phishing’ and reaches the ‘Phishing’ page in an MDN-supported Wikipedia interface where both the AG and the TT diagrams are displayed. MDN considers the current page, ‘Phishing’, as the current interesting concept to the user and highlights a related element (rectangle) in the AG diagram (with yellow background and red outline) as shown in Figure 1.3. This highlighting is what a C2D query does where MDN considers the current page as the query parameter and highlights related elements in the diagrams. Using this query, Tom can switch from the detailed content in the ‘Phishing’ page to a concise overview. He can also see the location of the current concept in the overall overview and start exploring nearby concepts. As multiple diagrams can show different perspectives, Tom can start exploration using a diagram that fits his interest or information need.

When Tom selects ‘Ransomware’ in the AG, a D2D query in MDN highlights related elements in the TT as shown in Figure 1.4.A. A D2D query helps Tom see concepts of interest from different perspectives (diagrams) and explore different sets of related concepts.

The third type of visual query is the D2C, which allows end users to select elements

Figure 1.4: Part A: a Diagram-to-Diagram (D2D) visual query. A user selects Ransomware in the Attack Graph; MDN highlights Trojan Horse, a related element in the Threat Tree. Part B: a diagram to content (D2C) visual query. A user selects elements in the diagrams to get a list of related Wikipedia pages on the right.

in the diagrams and retrieve related pages. In Figure 1.4.B, Tom selects ‘Sell Bot Services’ in the AG and ‘Backdoor’ in the TT to retrieve a list of related pages from Wikipedia.

MDN queries do not require prior knowledge of the domain as they are issued by selections made on the diagram elements or collection documents. Similar to hierarchical menus, in D2C and D2D, the available options (clickable concepts in the diagram) are organized in a meaningful structure (i.e., the relationships among the concepts). However, in MDN, the structure can be of any type and does not have to be hierarchical. Another difference from menus and browsing in general is that in MDN queries can access related collection documents not connected directly to the diagrams (as we describe in more detail in the next section). For example, in the D2C query in Figure 1.4.B, the query can retrieve and rank all related pages in Wikipedia. Therefore, the MDN D2C query in this case is more similar to keyword search than browsing. It is important to note that MDN queries only supplement, and do not

replace collection searching and browsing features.

1.2 Basic versus Extended MDN

In this dissertation, we present two versions of MDN. Basic MDN (or BMDN) is a simplified version that can only access collection documents directly connected to the diagrams. Extended MDN (or EMDN) is an extended version that can access all collection documents given that those documents are connected in some way. For example, in Figure 1.4.B, all pages listed in the D2C query result (‘Internet Bot’, ‘Zombie Computer’, and ‘Backdoor’) are directly connected to the ‘Sell Bot Service’ and ‘Backdoor’ in the diagrams. Content authors or other users have to manually create those direct connections between diagrams and a collection. BMDN relies only on those connections to provide limited navigation to a small subset of the collection. Based on our experience in connecting over 30 diagrams to Wikipedia pages (in our experiments), we note that direct connections between diagrams and a collection can be relatively few. In the case of the Wikipedia collection, each diagram concept generally corresponds to one related page in Wikipedia (e.g., there is one page about ‘Zombie Computer’ and one page about ‘Backdoor’). Since each diagram contained between 10 and 190 concepts, diagram-collection connections ranged between few tens to few hundreds connections.

EMDN extends the functionality of BMDN by exploiting internal connections in both diagrams and collections. For example, in the same D2C query in Figure 1.4, EMDN can benefit from hyperlinks among articles and categories in Wikipedia as well as connections between rectangles and ovals in the diagrams. Therefore, EMDN can reach other related pages such as ‘Denial of Service Attack’, ‘Email Spam’, and ‘Computer Network Security’. EMDN also ranks those pages based on their similarity to the concepts in the diagrams. Chapter 2 focuses on the BMDN system while the

rest of the chapters focus on EMDN.

1.3 Scope of This Research

We target Wikipedia as an example collection throughout this research. We select Wikipedia for its diverse content and high likelihood of being a destination for exploratory searchers. Wikipedia is also rich in hyperlinks, which we use in EMDN to retrieve and rank related content. Some of our findings are orthogonal to Wikipedia and can be useful when using MDN with any type of collection. Other findings are influenced by the nature of the Wikipedia structure. However, those findings can guide future research targeting different types of collections.

We also focus on D2C queries with a single selected diagram element in most of the chapters. Those are queries where a user clicks a single concept in a diagram to retrieve related documents in the collection. D2C queries are the most challenging in EMDN given that we have to find (and rank) related pages in a large collection (Wikipedia). On the other hand, C2D and D2D queries rank only concepts in a diagram. Also, by studying the ranking of Wikipedia pages in D2C queries, we benefit from a large body of research that focused on the same task. In Chapter 8, we switch to C2D queries and study different approaches to compute C2D queries with a single Wikipedia page. We do not aim to study the computation of D2D queries in this research. In Chapter 10, we will briefly discuss how it is possible to compute D2D queries using a combination of D2C and C2D queries.

1.4 Contributions

1. We present Multiple Diagram Navigation (MDN) as an exploration system that can use any type of diagram or map as an overview to explore a collection of documents. MDN provides three novel navigational query types. MDN requires

relatively few manual connections in the diagrams and between diagrams and collection documents. MDN also benefits from the abundance of internal connections in collections, such as hyperlinks in Wikipedia and citations in digital libraries. Using all of these types of connections, MDN can retrieve and rank content related to the diagrams.

2. In a user study, we evaluate the usability and usefulness of MDN, including all three types of MDN queries. We show that MDN can be useful in exploratory-search scenarios.
3. We identify two diagram models: the *Universal* model and the *Items and Attributes* (I&A) model. We apply these models with a variety of existing diagrams such as geographic maps, historic timelines, classification trees, and subway maps. We collected those diagrams from various domains.
 - The universal diagram model is a generic model that represents a diagram as a graph. We use different diagrams represented with different classes of graphs – such as a tree graph or a path graph – and report on the effect of those classes on retrieving and ranking related content.
 - The I&A diagram model can be applied to diagrams that exhibit a common pattern in Wikipedia (and possibly in other collections) of connecting entities – such as products, languages, and historic events – to their attributes – such as manufacturers, categories, and locations. We show that this model is useful in retrieving and ranking related content. We also show that I&A-Model diagrams can easily present multiple aspects of a domain. Furthermore, we show that ranking in D2C queries is influenced by the aspect of the diagram used. Therefore, issuing two D2C queries by

selecting the same concept in two diagrams, showing different aspects of a domain, yields different ranked lists of Wikipedia pages.

4. We show that both I&A-Model and Universal-Model diagrams positively bias the ranking process to retrieve pages related to the selected element in a D2C query. To study diagram influence on the ranking process we test two algorithms:

- *Personalized PageRank* (PPR), which is considered as the state-of-the-art algorithm in similar applications using the Wikipedia collection. PPR is influenced by both the diagram and Wikipedia content and structure.
- We introduce the *Greedy Energy Spreading* (GES) algorithm that spreads energy (activation) from diagrams to a collection. We designed GES to emphasize the influence of the content and structure of only the diagram used.

We test different settings of both algorithms. We show that even though PPR can be tuned to boost diagram influence, GES performs slightly better in some aspects of D2C and C2D ranking.

5. We test different graph structures by assigning different weights to reciprocal and non-reciprocal links. In the undirected graph we use in our experiment, reciprocal links are undirected edges created between two Wikipedia pages that link to each other (there is a hyperlink in each direction in Wikipedia). Non-reciprocal links, on the other hand, are created between a pair of pages that have only one hyperlink (in one direction) in Wikipedia. We show that for MDN, the best performance is achieved by including both reciprocal and non-reciprocal links, but with giving reciprocal links more weight. A recent study

[11] recommended using only reciprocal links when using the graph of Wikipedia in applications similar to MDN. In the graphs we used in our experiments, we show that constructing a graph with weighted edges and giving non-reciprocal links less weight than reciprocal links is better than either completely removing non-reciprocal links or weighting them the same as reciprocal links.

6. We also study different graph structures by including and excluding Wikipedia categories and article-template hyperlinks. We show that these graph elements have high but distinct influence on PPR and GES.

1.5 Dissertation Organization

In the next chapter we describe the BMDN system and present a user study where we evaluate the usability and usefulness of BMDN user interface. In Chapter 3, we introduce EMDN, including both the Universal and I&A models. We describe how we connect diagrams conforming to these models to Wikipedia and how we use internal connections in diagrams and Wikipedia to retrieve related Wikipedia pages in D2C queries. In Chapters 4 and 5, we introduce the PPR and GES ranking algorithms, respectively. Chapter 6 includes our first investigation regarding ranking in MDN, which covers D2C-query ranking in I&A-Model diagrams. Chapter 7 covers our investigation regarding D2C-query ranking in Universal-Model diagrams. In Chapter 8, we switch to C2D queries and study different approaches to compute them. We talk about related work in Chapter 9. In Chapter 10, we conclude and present directions for future work.

2 Basic MDN

Basic MDN (BMDN) is a limited version of MDN that uses only direct connections between diagrams and a collection. In Section 2.1, we describe our model of BMDN components, namely, diagram concepts, collection documents, and connections. In Section 2.2, we describe a simple graph algorithm to process BMDN queries. Section 2.3 includes a brief description of BMDN implementation. In Section 2.4, we talk about BMDN evaluation through a user study.

2.1 BMDN Components

To accommodate a broad range of diagrams, BMDN considers a diagram to be a set of *visual elements* that represents concepts in the diagrams. For example, in the Attack Graph (AG) and the Threat Tree (TT) diagrams shown in Figures 1.1 and 1.2, respectively, the visual elements represent the various computer security threats. In the Basic MDN design, the system is agnostic to the internal structure of the diagrams, such as the process structure in the AG and the hierarchy structure in the TT. However, the user can still benefit from these structures by seeing how the domain is organized. Similarly, MDN considers a *collection* to be a set of *content elements*, such as the Wikipedia pages related to the diagrams in the previous section. In Figure 2.1, we show an excerpt from the BMDN navigation system explained in Section 1.1, which includes excerpts from the AG and the TT and a set of related Wikipedia pages, Wiki_threats. Note that other (possibly related) Wikipedia pages

are not accessible in BMDN.

We use the term *cluster* to refer to either a diagram or a collection. Each visual and content element is represented by a pair (c, e) , where c is a cluster ID and e is an element ID. In this section, we use cluster and element titles as IDs for simplicity. For instance, the ‘phishing’ visual element in the AG is represented by the pair $(\text{‘AG’}, \text{‘Phishing’})$.

Figure 2.1: An overview of the components of a Basic MDN navigation system

A *connection* links a visual element in a diagram to a related content element. For example, there is a connection between ‘Backdoor’ in the TT and the ‘Backdoor’ page in the Wiki_threats collection. A connection may also link a visual element in a diagram to another related visual element in a different diagram. For instance, in Figure 2.1, there is a connection between ‘Phishing’ in the AG and ‘Phishing’ in the TT. For simplicity, we use the same weight for all connections in BMDN. Especially for connections between visual elements and their respective content elements, we did not need different weights given that we connected diagram elements (concepts) to their respective Wikipedia pages. Future MDN research can study different weights

for the described connections.

BMDN can use *connection by composition*. For example, ‘Phishing’ in Wiki_threats is connected by composition to ‘Phishing’ in the TT since both elements are connected to ‘Phishing’ in the AG. Considering all of the elements in MDN diagrams and collections as vertices in an undirected graph and connections as the graph edges, a *direct connection* is an edge in the graph and a *connection by composition* is an acyclic path.

Using BMDN, collection owner(s) can manually create connections. This task is relatively easy in small collections. Another approach is to encourage owners and expert users to collaboratively connect elements (e.g., in collections such as Wikipedia).

2.2 BMDN Visual Query Processing

Algorithm 1 Visual Query Processing Function: VQ

Function: $VQ((c, e), L, TC)$

Input: (c, e) is a selected vertex, L is the maximum path length, and TC is the set of target clusters.

Output: a set of vertices related to the input vertex.

1. $R =$ all vertices (rc, re) directly connected to (c, e) where $rc \in TC$
 - 2.
 3. $RComposition = \emptyset$;
 4. **if** $L > 1$ **then**
 5. **for** each $(rc, re) \in R$ **do**
 6. $RComposition = \cup VQ((rc, re), L - 1, TC - rc)$;
 - 7.
 8. Return $R \cup RComposition$;
-

Query processing in BMDN is a path traversal in the BMDN graph (e.g., the graph in Section 2.1) from a selected element in a visual query to connected (related) elements in other diagrams or collections. The VQ function, shown in Algorithm 1, is used in all types of visual queries (D2D, D2C, C2D) to retrieve related visual and content elements. We explain the algorithm using the first visual query shown in Fig-

ure 1.3. The query can be processed by calling $VQ(('Wiki_threats', 'Phishing\ page'), 1, \{'AG'\})$. The first parameter is the selected element. The cluster of the selected vertex, `Wiki_threats`, is considered the *source cluster*. The second parameter specifies the maximum path length for connection by composition. The third parameter lists the *target clusters* (TC) (or the domain of discourse), specifying where to look for related vertices.

The query processing starts by retrieving directly related vertices (line 1). If the query allows the use of connection by composition ($L > 1$), processing recursively invokes the VQ function (lines 5 and 6). The query processor ensures that it does not retrieve any vertices from the current cluster rc in the recursion by removing it from the set of target clusters. Visiting rc only once ensures that we only retrieve the most related elements in rc (i.e., coming back to the rc cluster would require longer paths leading to less relevant elements to be retrieved). Also, the approach of visiting a cluster only once simplifies the system. However, this requirement can be relaxed in other versions of the system if needed.

VQ accepts only one selected element. For queries with multiple selected elements, we compute the union of multiple VQ function calls. For example, the query in Figure 1.4.B is formed as:

$$VQ(('AG', 'Sell\ Bot\ Services'), 1, \{'Wiki_threats'\}) \cup \\ VQ(('TT', 'BackDoor'), 1, \{'Wiki_threats'\}).$$

2.3 BMDN Implementation

We have developed two MDN implementations. The first one was a prototype that we used in our evaluation described in the next section. In the prototype, we connected the AG and TT diagrams to related Wikipedia pages. This prototype supported only queries with a single selected element. An element is selected by

hovering the mouse in D2D or clicking in D2C. In C2D, the current page is considered as the selected content element.

The second MDN implementation is an MDN module [22] in the Drupal content management system. The module supports D2C and D2D queries with multiple selected elements. The module allows content providers to load diagrams and create connections between diagram concepts and Drupal content. The module development was supported by Google Summer of Code 2015 [76].

MDN relies on the Scalable Vector Graphics (SVG) format [53]. For a diagram saved in another image format, such as JPG and PNG, we embed the diagram in an SVG file. SVG can support two layers: a background to show the original diagram, and a layer on top of it to show semitransparent shapes. Adding shapes on top of a diagram can be done using any SVG editor, such as Inkscape or Adobe Illustrator. MDN uses the shapes for user interaction. It changes the color of the shapes to indicate selecting or highlighting a diagram element in the event of a mouse move or a click. MDN can also accept diagrams stored originally in SVG (i.e., with no image in the background). Many diagramming tools provides the choice of saving diagrams in the SVG format such as, InkScape, Adobe Illustrator, Microsoft Visio, and Google Docs.

2.4 BMDN Evaluation

We conducted a preliminary evaluation through a user study to get an indication of whether MDN is useful and easy to use in exploratory navigation and whether there are different MDN usage patterns. We targeted the following research questions:

- RQ1) Is MDN useful in exploratory navigation?
 - a) Is using overview diagrams useful in exploratory navigation?
 - b) Is using multiple overview diagrams useful in exploratory navigation?

- c) Are all types of visual queries useful in exploratory navigation?
- RQ2) Is MDN easy to use in exploratory navigation?
- RQ3) Are there distinguishable MDN usage patterns? Do these usage patterns correlate with the level of user experience?

2.4.1 Methods

We developed an interactive prototype of MDN with the AG, the TT, and related Wikipedia pages (the first implementation described in Section 2.3). We created 38 connections between the AG elements and Wikipedia pages, 20 connections between the TT elements and Wikipedia pages, and 16 connections between elements across the two diagrams. For D2C queries, users would click on a diagram visual element to see a list that included one or more related Wikipedia pages. For D2D queries, users could hover the mouse over visual elements to see related elements in the other diagram in a highlighted color. Because of the way we included pages from Wikipedia, cross-site-scripting limitations restricted C2D queries; prototype users had to paste or type a Wikipedia page title in a text box in order for the prototype to highlight visual elements related to that page. We included diagram zooming and panning and size options for each diagram.

We designed two questionnaires to collect users' feedback about using MDN. We used a scale of one to five with one indicating "basic" and five indicating "extensive" in six questions about users' levels of knowledge of the subject matter. All other questions included a scale of one to five with one indicating "strongly disagree" and five indicating "strongly agree".

We recruited ten students from the Computer Science department using the department mailing list and bulletin board ads. The user study started with a 15-minute presentation that introduced the features of MDN using two diagrams of the

animal kingdom. Then we briefly introduced the AG and TT diagrams. The study had five parts: the MDN prototype trial, a questionnaire with two questions, a first exploratory browsing task using the MDN prototype with only the AG, a second exploratory browsing task using the MDN prototype with both diagrams, and a final questionnaire. Subjects were given a \$10 gift card at the end of the user study. The first browsing task involved a comparison between malicious ads and identity theft, and whether malicious attachments could lead to either one. The second task involved comparing phishing and SQL injection. To emulate real exploratory browsing tasks, both tasks were ill structured and open ended as recommended by Wildemuth and Freund [105]. We include these tasks in Appendix A.

To study usage patterns in MDN, we logged the subjects’ clicks and hovers on the diagrams’ visual elements. When the subjects used C2D queries by copying a page title to a text box, we logged page titles of the pages that have connection(s) to visual elements. In addition, we logged the beginning and end of each browsing session. Because of the cross-site-scripting limitation mentioned earlier, subjects’ activity in Wikipedia was not logged. Due to an error in logging we were only able to log the activities of seven subjects.

2.4.2 Results and Discussion

2.4.2.1 Questionnaires

In the first questionnaire, we asked the subjects to report their prestudy level of experience in information security. The ten subjects had diverse backgrounds as shown in the second row of Table 2.1. Note that we use these levels and unique alphabet letters for user IDs shown in the first row.

In the final questionnaire, we asked the subjects to report their poststudy level of

Table 2.1: The subjects' level of experience before and after the user study.

User ID	A1	B1	C1	D2	E2	F2	G3	H4	I4	J4
Pretest Info. Security *	1	1	1	2	2	2	3	4	4	4
Δ Info. Security **	1	2	1	2	1	2	0	0	0	0
Pretest Adware vs. Identity Theft **	1	1	2	1	2	2	2	5	3	4
Δ Adware vs. Identity Theft **	0	4	1	3	1	2	1	0	1	0
Pretest Phishing **	1	1	1	2	2	3	2	4	5	4
Δ Phishing **	2	4	2	2	1	1	1	0	0	0
I use Wikipedia when I want to find out about new subjects and topics.	5	5	4	4	5	4	3	1	5	4

* Questions asked in the first questionnaire

** Questions asked in the final questionnaire

experience in information security; the difference (delta) is shown in the third row. Likewise, we included questions about prestudy and poststudy level of experience about concepts used in the first and second sessions, namely “the difference between adware and identity theft”, and “phishing”. Based on their prestudy level of experience in the three topics, we classified the subjects as: low-experience (LE) subjects (A1, B1, and C1); medium experienced (ME) subjects (D2, E2, F2, and G3); and high-experience (HE) subjects (H4, I4, and J4). We noticed that LE and ME subjects reported higher increases in level of experience in the three topics than HE subjects. In the last row in Table 2.1, we display subjects' responses about their likelihood to use Wikipedia in their exploratory search to read about new topics (with 1 being strongly disagree and 5 being strongly agree). Except for subject H4, all subjects stated that they are likely to use Wikipedia.

In five questions from the Technology Acceptance Model (TAM) [36]) shown in Table 2.2, subjects were very positive when they rated the usefulness and ease of use

Table 2.2: Questions (from TAM [36]) about usefulness and usability of MDN with mean and standard deviation for the ten subjects

Category	#	Question	Mean	Std. Dev.
Usefulness	1	MDN was useful for me to work on the given tasks.	4.4	0.49
	2	With the help of MDN I was able to accomplish these tasks quickly.	4.3	0.64
Usability	3	Interacting with the MDN was clear and understandable.	4.4	0.66
	4	It was easy to get the MDN to do what I wanted it to do.	4.3	0.78
	5	I think MDN was easy to use.	4.4	0.8

of MDN. Eight or nine subjects answered 4 or 5 for each of these questions; means are shown in Table 2.2. In another five questions, shown in Table 2.3, we wanted to assess the different types of visual queries in MDN. The overall response was positive. Two questions related to browsing support (#6 and #10) received higher scores than one question related to learning support (#9).

To compare the experience of using one versus two diagrams in the two sessions, we asked the subjects to rate four diagram configurations by answering the question “To what extent do you think the following approaches would help a user understand the big picture of information security”. The four methods were using: the AG, the TT, both diagrams, and three or more diagrams (assuming diagrams were useful and relevant). Subjects had to speculate about the three+ option since they did not use it. The averages in Figure 2.2.A show that the subjects preferred to use either the AG or both the AG and the TT. The more detailed results shown in Figure 2.2.B show that subjects had different preferences regarding using only the AG or both diagrams; two subjects on the left favored using the AG over both diagrams, the five subjects in the middle gave equal ratings, and the three subjects on the right preferred to use

Table 2.3: Questions about MDN visual queries: Diagram to Content (D2C), Content to Diagram (C2D), and Diagram to Diagram (D2D) with mean and standard deviation for the ten subjects

Category	#	Question	Mean	Std. Dev.
D2C	6	It was natural to navigate to web pages by clicking on boxes and ovals in the diagrams.	3.9	0.7
	7	Using the diagrams to navigate to the pages helped me explore the content.	4.3	0.78
C2D	8	I liked seeing my current location (current page), visited pages, and not yet visited pages highlighted in the diagrams.	4.3	0.46
D2D	9	Clicking on the attack graph boxes and seeing related topics in the tree and vice versa helped me learn about information security.	3.9	0.7
	10	Clicking on the attack graph boxes and seeing related topics in the tree and vice versa helped me choose the next topic to read in detail.	4.5	0.67

both diagrams over using only the AG.

To explore another aspect of subjects’ browsing styles, we asked the subjects to rate three browsing sequences: viewing diagrams before reading content, reading content before viewing diagrams, and doing both activities about equally. Five subjects thought that examining the diagrams first was helpful; they gave a high score to the first option and low scores to the second and third options. Four users favored doing both activities about equally. Only one user thought that reading content first was more helpful.

2.4.2.2 Activity Log

We cleaned the log by removing duplicate mouse clicks on the same visual element by the same user within one second. Duplicate clicks were likely a result of the user clicking twice or three times on an element that was not connected to Wikipedia

Figure 2.2: Preferred Number and Choice of Diagrams. Part A: Average response for ten subjects. Part B: All responses grouped to: two subjects on the left side favoring the AG over two diagrams, five subjects with equal ratings, and three subjects favoring two diagrams over only the AG.

pages. In that case, the prototype appeared to “do nothing”. We did the same cleaning for mouse hovers, which were used to issue D2D queries or to focus the mouse on different elements in the diagram for discovery. Figures 2.3.A and 2.3.B show heat maps, where white indicates the fewest and black indicates the most, for the total hovers for all users in session one and session two respectively. Note that container elements such as ‘persist on the system’ (the top element in the ‘Establish Presence’ column) had more mouse hovers because their edges were hovered whenever a mouse pointer moved towards their subelements (e.g., ‘Modify Existing Service’). Also, crossed elements in the AG had no mouse-event handlers. Only a few elements in the bottom of the TT, shown in their original yellow and orange colors, were not hovered or clicked at all.

In session one, one element on the left side of the AG (#1), and three elements on the right side (#2, #3, and #4) were mentioned explicitly or implicitly in the given exploratory browsing task; we call these the keyword elements. Given that the AG shows attack scenarios that start on the left side of the diagram, we consider elements located to the left of a keyword as pre-keyword elements. Similarly, we call elements located to the right of a keyword element as post-keyword elements. Figure 2.3.A shows that users focused their hovering on post-#1 elements and on the center

Figure 2.3: Part A: Heat map of total hovers for all subjects in session one (darker elements received more hovers) (numbers point to keyword elements mentioned in the browsing tasks). Part B: total hovers for all subjects in session two. Part C: total clicks for all subjects in session one. For top ten clicked elements, a pair shows # of clicks and # of users who clicked on the element. Blue/Red elements were not clicked.

of the diagram. On the right side of the AG, #2 and #3 caught little attention while almost no one paid attention to #4. The focus on the middle of the AG could be attributed to users trying to connect keyword elements on the left and right side of the AG. This observation is supported by the mouse-click heat map in Figure 2.3.C; for the top ten clicked elements in session one, we show pairs of the total number of clicks on an element and the number of users who clicked on that element (out of seven subjects). Most users focused on keywords and on the connecting elements in the middle of the AG.

In session two, there were three keyword elements: #5 in the AG; #6 and #7

Table 2.4: Averages of subjects’ hovering on different groups of visual elements (color scale is separate for the rightmost column). Clicks are shown as asterisks. Low-experience subjects (the top two rows) tended to spend more time on the diagrams (hover time) than high-experience subjects (the bottom two rows). Also, subjects focused their hovering and clicking on different groups of concepts or different sides of the diagram.

User ID	Ancestors (/5)	Siblings (/11)	Keywords (/3)	Pre-phishing (/5)	Post-phishing (/2)	Other (/83)	Overall (/109)
A1	*** *	2.00	1.82 *	9.33	3.40	4.50 *	3.61
B1		0.20	3.36 ***	6.67	11.60	** 7.00	2.07
D2		0.60	2.09	8.67 **	5.00	5.00 ***	2.82
E2		0.00	0.73	3.33	0.00	4.50	0.61
G3		1.20	2.18	8.33	0.00	2.00	0.95
I4	*** **	1.20	0.18 **	1.00	* 2.60	0.50	0.34
H4		0.60	0.27 *	2.33	2.80	0.50	1.46

in the TT. Subjects focused on these keyword elements and also on closely related elements (pre-keyword and post-keyword in the AG; siblings and ancestors in the TT). Table 2.4 provides more detail on the type of elements that the subjects focused their hovering and clicking on. We classified the elements in the two diagrams as follows: 3 as keywords, 5 as pre-keyword, 2 as post-keyword, 5 as ancestors, 11 as siblings, and 83 as other. We computed hover means for each subject per group using the formula: number of subject’s hovers on a group divided by the number of elements in that group. We also show the number of clicks on a group’s elements as asterisks in the group cell. Considering both hover means and the number of clicks, we see that all users were interested in keyword elements. With respect to other areas of the diagrams, subjects showed different preferences as they focused on different groups of related concepts or tended to look at different areas or sides of the diagrams.

Looking at Table 2.4 from top to bottom, specifically comparing LE and HE subjects, we notice that except for the ancestors group, hover averages correlate negatively with level of experience. This correlation suggests that compared to domain experts, domain novices spent more time on the diagrams to comprehend the subject and to look for keyword and related elements.

Since subjects’ browsing was centered around keyword elements, we traced each subject’s mouse activity to discover how each subject arrived at these elements. Table 2.5 lists the methods used. Note that some subjects used two methods and visited a keyword twice. Hovering (moving around a diagram until finding a keyword element) was the most popular method in session one. LE subjects also relied mostly on hovering in both sessions. The C2D query type was the second most popular method and was solely used by ME and HE subjects. Since all subjects used the AG before the TT, a D2D query was handy for finding ‘phishing’ in the TT (#7 in Figure 2.3) by hovering the ‘phishing’ element in the AG (#5); three subjects seemed to benefit from this query while the others used C2D or hovering.

Table 2.5: Types of visual queries used to find keyword elements. While high and medium experienced subjects used more C2D queries, low-experience subjects depended on hovering on the diagrams

User ID	A1	B1	D2	E2	G3	H4	I4
Email	Hover	Hover	Hover	C2D	Hover	Hover	Hover
Adware/Ads Displayed	Hover	Hover	Hover	C2D	Hover/C2D	Hover	Hover
AG Phishing	Hover	Hover	Hover	C2D	C2D	Hover	Hover/C2D
TT Phishing	Hover/D2D	Hover	D2D	C2D	D2D	C2D	C2D
TT SQL Injection	Hover	Hover	Hover	C2D	C2D	none	Hover/C2D

2.4.2.3 Summary of Discussion

In this section, we further discuss and summarize our discussion to answer our research questions stated in Section 2.4. We also identify areas for future work that can be pursued in other research projects.

RQ1) Is MDN useful in exploratory navigation? Our findings from the questionnaire shown in Table 2.2 indicate that subjects perceived MDN as useful in exploring a new subject. Subjects valued the additional support that MDN provided compared to the original Wikipedia interface (since most subjects are familiar with exploring

new subjects using Wikipedia (from Table 2.1)). In addition, the self-reported increase of knowledge by the LE subjects in Table 2.1 indicates that using MDN was a positive experience for domain novices who are the targeted end users for exploratory navigation.

(1-a) Is using overview diagrams useful in exploratory navigation? The use of the AG in session one (Figures 2.3.A and 2.3.C), and the high scores the AG received in Figure 2.2, suggest that utilizing well-prepared overview diagrams (even without coordinating multiple diagrams) seems to be useful in exploratory navigation. The low score of the TT in Figure 2.2 shows that not every diagram is useful, and suggests the importance of investigating characteristics such as size, structure, and layout as well as content of diagrams used in exploratory navigation.

(1-b) Is using multiple overview diagrams useful in exploratory navigation? There are also positive indications for using multiple diagrams in exploratory navigation. Eight subjects gave a score of 4 or 5 to the option of two diagrams in Figure 2.2. The low score for the three+ option suggests that using more than two diagrams should be facilitated with diagram browsing and selection features, where users can choose which diagrams they want to view at the same time. Findings from Section 2.4.2.2 also indicates the usefulness of multiple diagrams. Subjects utilized both diagrams as shown in Figure 2.3.B and Table 2.4. In addition, the high average of D2D queries in Question #10 in Table 2.3 implies that subjects liked seeing interesting concepts in different contexts.

(1-c) Are all types of visual queries useful in exploratory navigation? The subjects' positive responses on Questions #7, #8, and #10 in Table 2.3 indicate that subjects found D2C, C2D, and D2D queries useful. In addition, our analysis in Section 2.4.2.2 shows that all or most of the seven subjects used these queries. D2C was used by

most subjects in both sessions (according to the number of clicks in both sessions). C2D was also used by four users in each session despite the fact that it was relatively harder to use. The use of D2D queries was harder to track, since they were issued by hovering the mouse. However, the usefulness of D2D queries is indicated by the fact that they had the highest score in the questionnaire (Question #10 in Table 2.3). In addition, three users used this query type to find a keyword element (Table 2.5).

RQ2) Is MDN easy to use in exploratory navigation? The high averages of subject responses on ease-of-use questions shown in Table 2.2 indicate that subjects were able to operate MDN easily.

RQ3) Are there distinguishable MDN usage patterns? Do these usage patterns correlate with the level of user experience? Based on our log analysis, LE subjects (domain novices) seemed to spend more time on the diagrams than HE subjects (experienced users). Also, subjects seemed to focus their exploration on different areas of the diagrams. Note that these patterns were observed during relatively short session times compared to more lengthy real world exploratory search activities. Studies spanning longer periods of time may provide more insight. Another pattern indicated in the questionnaire was the preference of viewing diagrams before reading content or doing both activities together. Because of the lack of detailed knowledge about subjects' reading, browsing, and searching activities in Wikipedia, studying this pattern was not possible.

2.5 Conclusions

This chapter introduces Basic Multiple Diagram Navigation (BMDN). We present a design of a BMDN system, including our models for BMDN diagrams and collection; and a simple graph algorithm for processing BMDN queries. Our evaluation of BMDN indicates that users perceived MDN as useful and easy to use in exploratory

navigation. Our detailed findings suggest that with exploratory navigation, it is useful to incorporate single or multiple overview diagrams supported by diagram-to-content, diagram-to-diagram, and content-to-diagram visual queries. We also see indications of three MDN usage patterns, with one pattern correlating with users' level of experience.

In general, our evaluation in this chapter focuses on studying the effectiveness of the MDN interface, and also the behavior and preferences of MDN users. A possible direction of future research could be to continue on this path to study the human computer interaction (HCI) aspect of MDN. However, in the rest of this dissertation, we direct our efforts to the internals of MDN. For a collection that includes internal connections among the documents, we study different approaches to extend MDN to access all of those documents.

3 Extended MDN

Basic MDN is limited to accessing collection documents that are directly connected to the diagrams. In this Chapter, we introduce Extended MDN (EMDN), a navigation system that can access collection documents not directly connected to the diagrams. The rest of the chapters in this dissertation pertain to EMDN. We will use the terms EMDN and MDN interchangeably, and will use the term Basic MDN when specifically discussing the limited version of MDN described in Chapter 2.

To access collection documents not directly connected to the diagrams, we use internal connections in both diagrams and the collection. In collections, MDN can use any type of internal structures that connect related content elements. Examples include hyperlinks and the category hierarchy in Wikipedia; similarity and recommendation connections in online shopping websites; metadata and bibliographic citations in digital libraries; and tags in social-bookmarking websites. For example, consider the two diagrams in the domain of world languages shown in Figure 3.1.A; the top map organizes spoken languages by country while the bottom forest groups languages by trees and branches. In Figure 3.1.B, we show the connections between the language forest and Wikipedia in dashed lines such as the ones connecting ‘Dinka’ and ‘Nilo-Saharan’ in the diagram to their related Wikipedia pages. Those are the only connections used in Basic MDN. We show internal connections in solid lines. Those connections include hyperlinks in Wikipedia and relationships in the forest diagram (e.g., connecting the ‘Dinka’ language with the ‘Nilo-Saharan’ language family in the

diagram). Note that we use an undirected graph. We explain the graph structure later in this chapter.

Figure 3.1: A: a Diagram-to-Content (D2C) query in Extended MDN (EMDN). A user clicks ‘Dinka’ language in the World Languages Forest Diagram [45]; MDN displays a ranked list of related Wikipedia pages. B: Previously used connections between diagrams and Wikipedia in BMDN in dashed lines. Additional internal diagram and Wikipedia connections used in EMDN in solid lines.

With internal connections, MDN can access collection documents not directly connected to the diagram. Consider the set D of collection documents that are directly connected to diagram elements. In addition to those documents in D , MDN can access any collection document x where there exists a path between x and $y \in D$. For collections where documents are highly connected, MDN can often access the entire collection. For example, in Wikipedia, pages are highly connected via hyperlinks. Therefore, it is often possible to traverse from the diagram to any Wikipedia page and vice versa.

An example of accessing collection documents that are not connected to a diagram is shown in Figure 3.1.A. Selecting ‘Dinka’ in a D2C query, MDN can retrieve the ‘Dinka’ page, which is directly connected to the diagram as well as the ‘Nuer’ and the ‘Majang’ pages that have no direct connections to the diagram. Furthermore, visual elements in the diagram with no connections to the collection can still be used in

queries. For example, assume the ‘Zarma’ language in the diagram is not connected to a page in Wikipedia (or has no related page). By selecting ‘Zarma’ in a D2C query, MDN can use ‘Zarma’ connections to ‘Nilo-Saharan’ and ‘Songhay’ in the diagram to retrieve related pages such as ‘Amdang’ and ‘Dendi’.

Given the opportunity to access a large number of collection documents in EMDN, ranking documents and diagram elements becomes necessary. For example, in the last example when ‘Dinka’ was selected in a D2C query, MDN could rank the ‘Nuer’ page higher than the ‘Amdang’ page, since only the ‘Nuer’ page is connected to the highly relevant pages: ‘Dinka’, ‘Nilo-saharan’, and ‘E. Sudanic’. In Figure 3.1.A we show an example ranked list of Wikipedia pages when selecting ‘Dinka’ in the diagram. Ranking is also needed in C2D and D2D queries. For example, assuming the ‘Dendi’ page was selected by the user in a C2D query, MDN ranks ‘Zarma’ and ‘Songhay’ visual elements higher than ‘Dinka’ and ‘Eastern-Sudanic’. MDN presents the ranking on the diagram by using different colors when highlighting the visual elements in C2D and D2D queries. As mentioned in Section 1.3, we will focus on the ranking in D2C queries for now.

We also introduce *aspect-influenced ranking* in EMDN. Since MDN can present different aspects by using multiple diagrams, we can incorporate the user’s favored aspect in the ranking process. For example, a user may wish to retrieve languages spoken in the same country, in this instance ‘South Sudan’, by clicking on ‘Dinka’ on the map. On the other hand, selecting ‘Dinka’ on the forest returns languages in the same tree and branch (i.e., languages in the ‘South-Sudanic’ and ‘Nilo-Saharan’ language families). Aspect-influenced ranking can be useful in exploratory search scenarios when one of the diagrams is interesting to the user or matches the user’s needs. Traditional information retrieval features such as lookup search may involve

all aspects of a domain including the ones not desired by the user.

MDN can involve multiple aspects in the ranking process by combining multiple selected elements from multiple diagrams. For instance, combining both queries together in the previous example (selecting ‘Dinka’ languages in the map and forest) would rank pages based on their relatedness to both ‘South Sudan’ as a country and the language families ‘South-Sudanic’ and ‘Nilo-Saharan’ (e.g., by averaging the ranks of each page in two individual queries). Combining queries from multiple diagrams is similar to faceted search systems where users can filter content by using different attributes such as price, color, and brand. The facets in MDN are the different perspectives that can be shown in the diagrams.

Our work in EMDN is centered around two fundamental elements: diagram models and ranking algorithms. We introduce two diagram models in Sections 3.2 and 3.3, one of them is generic and can be used with any diagram while the other is specific to a class of diagrams. We also study two ranking algorithms. We study the use of the Personalized PageRank (PPR) [58, 84] in MDN in Chapter 4. PPR is a popular ranking algorithm used in similar applications such as in the domains of Natural Language Processing (NLP) and recommender systems. Applications in those domains use PPR to rank Wikipedia pages (such as in MDN D2C queries). We also introduce the Greedy Energy Spreading algorithm (GES) in Chapter 5. GES is an algorithm that maximizes the influence of diagrams on the ranking process while limiting the effect of the Wikipedia structure.

In this chapter, we start by introducing Wikipedia, our example collection, in Section 3.1. In each of the two diagram models explained in Sections 3.2 and 3.3, respectively, we describe how we model diagrams and how we build a graph of diagram elements and Wikipedia pages.

3.1 The Wikipedia Collection

On Sept. 9, 2007, Wikipedia, the free encyclopedia, became the largest encyclopedia in the world [33]. As of September 2018, the English Wikipedia included 5,713,705 articles with about 550 new articles per day [10]. In 2017, the site had 92.29 billion page views and 1.79 million new registered users in the English Wikipedia alone [8,9].

Wikipedia main entries are *articles* and *categories*. We sometimes use the term *Wikipedia page* to refer to either articles or categories. Each article has a unique name. Redirect pages may exist to represent synonyms or closely related concepts to an article; those redirect pages transfer readers to that article. For example, ‘Malicious Software’, ‘Badware’, and ‘Jokeware’ are redirect pages that redirect users to the ‘Malware’ article. In addition to the text and images in the article body, an article may contain an *information box* (*infobox*) that is usually displayed in the top right corner of the article page. The infobox lists important information and hyperlinks about the concept of the article. Articles may also include one or more *templates*, usually displayed at the bottom of the page. Each template contains hyperlinks to articles related to the current article in a particular aspect. For example, the ‘Cartoon Network’ article, shown in Figure 3.2, has seven templates reflecting different aspects of the article such as Turner Broadcasting System (the channel’s parent company), children’s TV channels in the US, and companies based in Atlanta, GA. Templates are usually rich in hyperlinks. They may actually contain more hyperlinks than the main body of the article.

A Wikipedia *category* groups articles that share a common aspect or attribute. An article can belong to multiple categories. A category can have multiple parent categories and multiple child categories. We can view the category hierarchy in Wikipedia as a directed graph where each category has outgoing links to parent and

Cartoon Network (abbreviated as **CN** since 2004) is an American pay television channel owned by Turner Broadcasting System, a subsidiary of AT&T's WarnerMedia. The channel was launched on October 1, 1992.

The channel primarily broadcasts animated television series, mostly children's programming, ranging from action to animated comedy. It operates daily from 6:00 AM to 9:00 PM. (ET) and primarily airs to young teenagers between the ages of 7 to 15, and to adults with mature content during its late night daypart block, Adult Swim, which is treated as a separate entity for promotional purposes and as a separate channel by Nielsen for ratings purposes.^[1] A Spanish language audio track for select programs is accessible via second audio programming (SAP); some cable and satellite companies offer the Spanish feed as a separate channel by removing the main English-language audio track. It is also the related channel of Boomerang.

As of January 2016, Cartoon Network is available to approximately 94.0 million pay television households (80.7% of households with television) in the

Launched	October 1, 1992; 26 years ago
Owned by	Turner Broadcasting System
Picture format	1080i HDTV (downscaled to 480i letterbox for SDTVs)
Slogan	<i>New New New New</i>
Country	United States
Language	English (Spanish with SAP)
Broadcast area	National
Headquarters	1050 Techwood Drive NW, Atlanta, Georgia (main) New York City (operational) Burbank, California (West Coast)
Sister channel(s)	Audience Adult Swim Boomerang

Programs & blocks (U.S.)

Current	Adult Swim (programming) • Boomerang (programming) • Toonami (programming) • Jetstream
Former	Acme Hour • Cartoon Cartoons • <i>Cartoon Planet</i> • DC Nation • High Noon Toons • Tickle-U

Asia and Oceania (TBSAP)

Australia & New Zealand • India • Japan • Pakistan • Philippines • Southeast Asia • South Korea • Taiwan
Arab World • Arab World, Africa and Cyprus • Central and Eastern Europe •

Turner Broadcasting System [show]

Family-oriented television channels in the United States [show]

Atlanta companies

Categories: Cartoon Network | Children's television networks in the United States | Television channels and stations established in 1992 | English-language television stations in the United States | Companies based in Atlanta | Companies based in Burbank, California | Companies based in New York City | Turner Television networks | American television networks | American companies established in 1992

Figure 3.2: An example Wikipedia article about the Cartoon Network TV Channel. The information box, displayed in the top right, contains important information and hyperlinks. Multiple templates and categories reflecting the multiple aspects of the article are displayed in the bottom. Templates usually have a multitude of hyperlinks.

child categories. The Wikipedia category graph contains cycles, so it is possible that a category ancestor can also be the child of that category. Bairi et al. [17] reported that Wikipedia contained 1766 and 2804 cycles in the Wikipedia category graph in 2012 and 2014, respectively, with cycle length ranging from four to fifty.

Another organizing structure in Wikipedia is *List_of* pages such as the 'List of

American Civil War battles’ and the ‘List of chocolate bar brands’. Those lists are technically article pages, but their main purpose is to group and list related concepts with hyperlinks to their associated articles.

3.2 The Items and Attributes (I&A) Diagram Model

In this section, we describe the *Items and Attributes* (I&A) diagram model. We introduce the I&A model to represent some diagrams that share a common pattern. The I&A model is a bipartite graph that includes two sets of nodes: items and attributes. The items set contains the entities or concepts that are usually the focus of the diagram. For example, we can consider world languages as the set of items in each of the world-languages diagrams in Figure 3.1. Also, we consider the candy bars as the set of items in the diagram shown in Figure 3.5 as the diagram presents an overview of candy bars.

Attributes are the set of concepts that are included in the diagram to describe the items. For example, in the language-forest diagram in Figure 3.1, the set of attributes includes the language families that describe the lineage of each language. In Figure 3.5, attributes represent the ingredients that describe the content of each candy bar.

Using the notation of the entity-relationship model, we can describe the I&A as a model that contains only two entity types: item and attribute (see Figure 3.3). The item entity type can have multiple relationships with multiple attributes and vice versa. In some diagrams, attributes (and also items) may have multiple subtypes or subclasses. For example, ingredients (represented as attributes) in Figure 3.5 have different types such as nuts (e.g., almond and peanut) and chocolate types (e.g., milk and dark chocolate). However, for simplicity we only consider those ingredients as attributes with no distinction.

Since multiple diagrams focusing on the same items can include different at-

Figure 3.3: Representing the I&A Model using the notation of the entity-relationship model.

tributes, they can represent different aspects of those items. For example, the two diagrams in Figure 3.1 show the classification and geography aspects of world languages by including different sets of attributes (language families in the forest and countries in the map). In our evaluation in Chapter 6, we use ten pairs of I&A diagrams in ten different domains. Each pair shows two different aspects of its domain. Using these pairs we study the effect of diagram aspects on ranking (i.e., aspect-influenced ranking).

The I&A model can be used with different types and shapes of diagrams where items, attributes, and relationships among them take different forms. In other words, the I&A model is not restricted to a specific visual pattern. In Figure 3.1, we assume an implicit connection between each language and its country in the map. In the forest, we connect every language (item) with related language families (attributes) shown either as a tree or a tree branch. (Note that the diagram can also be represented using a tree graph.) In Figures 3.4 and 3.5, we show two more diagrams, a “subway” map for database platforms and a ring-like diagram for candy bars. In the first diagram, the items are the database platforms while the attributes are the database features (shown on the right side), such as the database model, in-memory support, and grid support. Each attribute in this diagram was represented as a subway line that connects all items having the attribute. In the second diagram, the items are the candy bars in the ring while the attributes are the ingredients. A line connects

each candy bar node to its ingredient nodes.

The I&A diagram model is limited since it matches a subset of diagrams (compared to the universal diagram model we explain in Section 3.3, which is generic). In fact, the I&A diagram model is a subclass of the universal diagram model. We studied the I&A diagrams in more detail because of the way they present different aspects. With I&A diagrams we could often find pairs of diagrams that show different aspects and still include similar sets of concepts. For other classes of diagrams, we observed that multiple aspects are often presented using different sets of concepts. For example, two diagrams can focus on theoretical and practical concepts of a particular subject. By studying I&A diagrams we wanted to separate the effects of perspective from those of coverage. Moreover, compared to the universal model, it was easy to find pairs of I&A diagrams in multiple domains, or find one diagram and create another by using different attributes. As we explain in Chapter 6, we needed those pairs to study the aspect-influenced ranking.

Figure 3.4: An excerpt of a subway map for data platforms [7]. We model data platforms as items and features (some shown at the bottom right) as attributes. An attribute subway line connects all items that have that attribute.

3.2.1 Model Guidelines and Graph Construction

We explain how we model I&A diagrams and connect them to Wikipedia, and we explain the resulting graph. We use an undirected weighted graph where an edge weight represents the strength of the relationship between two nodes. We give less weight to some edges in Wikipedia, as we explain later in this section. In Figure 3.6, we show a part of the graph of the language-forest diagram. In the I&A model, we assume that an item can connect to one or more attributes within the diagram. In some I&A diagrams, an item typically connects to one attribute such as the connection between a candy bar and its manufacturer. Currently, we do not include connections among items and among attributes for simplicity (recall that the graph of an I&A

diagram is a bipartite graph).

Figure 3.6: Part of the graph of the language forest diagram. We model items and attributes diagrams as a bipartite graph where an item can connect to multiple attributes.

Looking at the Wikipedia hyperlink graph, we notice that item pages usually have links to their attribute pages and vice versa. For example, a language (item in this case) usually connects to the language families (attributes) that it belongs to. Also, a language-family page usually connects to its languages. Item-attribute (and attribute-item) links can be found in info boxes as well as other parts of the body of Wikipedia articles.

Wikipedia has an additional useful structure related to the I&A model. Categories and List_of pages usually group similar items. For example, there are categories or List_of pages for candy bars, world languages, and American Civil War battles. In our evaluation in Chapter 6, we harvest items (item pages) from such categories and List_of pages and construct a set of pages that we call the *Target Universe* (TU). We then direct queries issued from an I&A-Model diagram (e.g., world languages) to only pages in a TU that match the content of the diagram (e.g., the TU of world languages). The main reason we limited our Wikipedia navigation to only a set of pages is to simplify the page-labeling process that we used in our evaluation. Two

students labeled the top-ten ranked pages in each D2C query. It was simpler to maintain a consistent labeling criteria when dealing with pages that have the same type (e.g., label only world-language pages when issuing queries from world-language diagrams). For more details, we list the labeling criteria in Appendix B.

Another reason for using only pages in a target universe is that the size of the TU sets that we collected seemed large enough with regard to their domains. The size ranged from a few hundreds to several thousand items. We argue that important and popular items were likely to be included in those TU sets. Furthermore, a possible advantage of using a target universe is allowing a user to change it to another subset of the domain. For example, instead of retrieving languages, a user might query countries or ethnic groups related to languages in the forest shown in Figure 3.1.A by switching to a different target universe. To include this feature, multiple target universes can be connected to a specific diagram. Then, a user can select a specific target universe (e.g., from a menu) before issuing a D2C query.

Another design decision we made in our evaluation in Chapter 6 is using partial I&A-diagram graphs (i.e., we do not include and connect all of the items and attributes in I&A diagrams). For each D2C query we used a graph that included a selected item and its related attributes as well as Wikipedia pages. In Figure 3.7, we show an example partial graph for the D2C query in Figure 3.1, where the ‘Dinka’ language is selected in the language forest. In the diagram part, the graph only includes ‘Dinka’ and its related language families (other items and attributes are not included).

We used the partial graphs in early experiments where both tested algorithms showed very good performance in retrieving related Wikipedia pages in D2C queries and generating aspect-influenced ranking. We decided to keep using those partial

graphs as any ranking properties can be associated with using only attributes (e.g., aspect-influenced ranking is a result of switching to different attributes when using a pair of diagrams). This observation can be useful in future MDN studies that consider biased ranking in other types of diagrams and graphs by increasing the influence of certain diagram element(s). For example, in a tree graph, one can study biased ranking when maximizing the effect of the parent node of a selected diagram element in a D2C query. A similar study can investigate activating only certain neighbor element(s) of a selected element when those neighbor elements represent an interesting aspect or subdomain.

Note that the use of partial graphs for D2C queries can be implemented when working with a complete graph that includes all diagram items and attributes. One way to do so is to temporarily assign weight zero for edges that are not part of the partial graph for a certain D2C query. For example, for the D2C query in Figure 3.7, we assign weight one to the edge between ‘Eastern-Sudanic’ and ‘Dinka’ since this edge is part of the partial graph for this query. However, we assign weight zero to the edges that connect ‘Eastern-Sudanic’ to other items in the diagram (e.g., the edge between ‘Eastern-Sudanic’ and ‘Acholi’).

The Wikipedia part in Figure 3.7 includes item Wikipedia pages (pages A to D) that are related to the diagram. Recall that we target item Wikipedia pages that we harvest from categories and List_of pages. We add a target universe (TU) node that links to all of those pages and also links to the selected item. The TU resembles a container page (e.g., a category page that links to related item pages). The graph also contains attribute pages (‘Eastern-Sudanic’ and ‘Nilo-Saharan’).

In the graph shown in Figure 3.7, a node $n \in Level\ k$ ($L\ k$) if the shortest path between n and any diagram element e has the length k . For example, *Level 1* ($L1$)

contains the set of nodes whose shortest path to the diagram consists of one edge (i.e., *TU* node and Wikipedia pages that are directly connected to the diagram). The *Level 2* (*L2*) set includes Wikipedia pages whose shortest path to the diagram consists of two edges.

Figure 3.7: A partial graph for a single D2C query including the selected element ‘Dinka’ and its related attributes ‘Eastern-Sudanic’ and ‘Nilo-Saharan’. The target universe node *TU* connects the selected element to all Wikipedia pages (items) harvested from related categories and List_of pages.

As we discussed earlier, we rely on attributes to rank Wikipedia pages (items). One technique that we found useful is to connect a diagram attribute to multiple related Wikipedia articles and categories that have links to related items. For example, in the language-map diagram, the intuitive Wikipedia page to link to from ‘South Sudan’ in the diagram is the page about the country ‘South Sudan’ in Wikipedia. Even though the ‘South Sudan’ page does include content about languages in South Sudan, a more useful page is ‘Languages in South Sudan’ (each country has a separate page about its languages). This country-languages page is richer in content and hyperlinks to related languages in the context of the diagram. Connecting to both pages helps in the ranking process, as we explain in the next section.

The last part of the graph in Figure 3.7, is the edges among Wikipedia pages in Level 1 and Level 2. We create an edge between any two pages p_1 and p_2 if there is

a hyperlink from p_1 to p_2 or vice versa. We consider p_1 to be equally related to p_2 regardless of whether p_1 has an outgoing or incoming hyperlink to p_2 . This approach of omitting hyperlink direction may not fit other graphs, such as the links between pages in the web or citations between academic papers in digital libraries. Links in those graphs can be considered as an endorsement or authority transfer [84]. For example, in a digital library, a paper might be considered important if it is cited by an important paper. However, the assumption is not true if we reverse the direction of the citation [18]. In Wikipedia, we are interested in the relatedness aspect of hyperlinks more than the endorsement aspect. In the I&A model for example, an ingredient (attribute) can have only incoming links from related candy bars (items). Those incoming links can be very important as we traverse from the diagram to attributes and then to related items.

Hyperlinks in Wikipedia may not indicate strong relationships with respect to the diagram domain. For example, a candy bar may link to events when the bar is advertised, such as the Snickers advertising campaign during the FIFA World Cup and the NFL Super Bowl. To overcome this problem, we use a technique from Agirre et al. [11] that favors *reciprocal links* where a pair of pages p_1 and p_2 link to each other. Reciprocal links may increase the confidence that a pair of pages are in the same domain and they are highly related. Unlike Agirre et al. [11], who kept only reciprocal links, we also keep nonreciprocal links. In our experiments, we assign maximum weight (weight one) to reciprocal links and try different weights for nonreciprocal links, such as zero (i.e., use only reciprocal links), 0.5 (i.e., giving reciprocal links an advantage), and one (i.e., treat reciprocal and non-reciprocal links equally). We report on the performance of the different values we use for the nonreciprocal link weight in Chapter 6. For other edges in the graph, we give maximum weight to all

connections in the diagram and between the diagram and Wikipedia. We also give maximum weight to all connections linked to the *TU* node.

3.2.2 I&A-Model Support for D2C Queries

For D2C queries in I&A diagrams, we aim for item pages related to a selected item or attribute in the diagram. Finding those item pages can be beneficial, especially with overview diagrams that include only a few item elements as examples. Consider a set A of related attribute pages that are connected to a selected diagram item i (i.e., there is a two-edge path between each page $p \in A$ and i through some attribute in the diagram). We expect that any related item page q to be also connected to most attribute pages in A . For example, in Figure 3.7, ‘Dinka’ is the selected element and the relevant attributes are Eastern-Sudanic and Nilo-Saharan (‘Nilo-Saharan’ and ‘Eastern-Sudanic’ pages). We expect other languages related to these language families to be connected to the ‘Nilo-Saharan’ and ‘Eastern-Sudanic’ pages and therefore be ranked in the top of the list. These related languages (such as language D) have the advantage of the extra paths to the selected element ‘Dinka’ through the ‘Nilo-Saharan’ and ‘Eastern-Sudanic’ pages compared to unrelated languages (such as page C). In Chapter 6, we show that both ranking algorithms described in Chapters 4 and 5 performed reasonably well in terms of retrieving related items in D2C queries issued from I&A diagrams.

We also hypothesize that the I&A diagram model can help in producing aspect-influenced ranking. If we issue the same query by selecting the ‘Dinka’ language on the language map diagram, the available attribute is the country ‘South Sudan’ and its connected pages, and therefore we expect languages connected to this country to be ranked higher. Each diagram will empower different attributes and hence influence the ranking. In Chapter 6, we show that I&A diagrams have a large influence on the

ranking process.

3.3 The Universal Diagram Model

Not all diagrams can be represented using the I&A diagram model. In this section, we introduce the *Universal Diagram Model*, which can be used to represent any diagram. The universal model is simply a graph where connections among diagram visual elements have only one restriction. We require that the graph of the diagram be connected so it is possible to get from any visual element to any other visual element. In some cases, we have to add some extra nodes and edges in the diagram graph to achieve this goal (those nodes and edges are not part of the original diagram and they can be added automatically by an MDN system). This requirement is to ensure that in D2C queries we benefit from the whole diagram by traversing to all visual elements and to their associated Wikipedia pages. We use both directed and undirected graphs for universal-model diagrams, as we describe in Sections 3.3.1 and 3.3.2, respectively.

Based on the relationships among the visual elements in the diagram, the graph can take on different structures. In our experiments, which we describe in Chapter 7, we used the universal model with twenty diagrams. These diagrams were represented with four classes of graphs. These graph classes are: I&A graphs, tree graphs, path graphs, and generic graphs. The generic class has no particular structure and can be used with any diagram that cannot be represented using the other classes. We will explain these graph classes in turn.

We use the I&A class to represent diagrams that fit the I&A model. In this section, we consider items and attributes (I&A) as one of the graph classes in the universal model. We study the I&A class using a different approach than the one we use for the I&A model. We connect diagrams to diverse Wikipedia pages instead of

only connecting to item pages (i.e., we do not construct a target universe nor do we include a TU node in the graph). We also use a different evaluation approach as we explain in Chapter 7. The graph of the I&A class looks similar to the one shown in Figure 3.6, which shows a few items and attributes in the language forest diagram. However, to make sure the graph is connected we add some extra nodes and edges as we show in Figure 3.8 (node *C* and the nodes connected to it). It is possible to traverse from any node to any other node by using a path of four edges (through a central node *C*). From any selected node, related items and attributes are reachable with shorter paths of one or two edges.

Figure 3.8: Part of the graph resulting from representing the language forest as a universal model diagram. We make sure that all nodes in the graph are connected by adding edges to a central node *C*.

The path class can be used where concepts are laid out in the diagram as a sequence and where close concepts are more related than distant concepts (i.e., relatedness is negatively correlated with distance). In some diagrams, concepts (e.g., rectangles or ovals) are arranged in a sequence or a path, however they are equally related to each other. In this case, we represent the diagram using other graph classes as we explain later in this chapter. The distance-based relatedness in path graphs is natural for timeline diagrams where close events tend to be more related than the distant ones. In Figure 3.9, we show two diagrams represented with a path graph.

The first diagram is a timeline for the major events in World War II, while the second one is a list of the bridges in the city of Portland, Oregon. Figure 3.10 shows a part of the path graph of the World War II diagram. We do not use different weights for the edges in the graph that represents the diagram, therefore, we do not include exact distance information in path graphs (e.g., distance in days between adjacent events or in miles between adjacent bridges).

Figure 3.9: Part A: An excerpt of World World 2 timeline [88]. The diagram is a sequence of concepts where close concepts are more closely related in time than the far ones. Part B: An excerpt of a diagram [91] showing all bridges in Portland, Oregon. The diagram has a similar organization to the timeline, arranging concepts in a sequence, but based on geographic location.

The tree class is handy to model different types of diagrams when each area of the diagram contains a set of related visual elements. In Figures 3.11 and 3.12, we show excerpts from two diagrams that match the tree class. The diagrams are about JavaScript tools, and coffee preparation and production, respectively. The first diagram is a mind map. Mind maps usually use the tree structure. The coffee diagram has distinct groups of concepts, such as producers of coffee and examples of coffee pots and machines. In Figure 3.13, we show a part of the tree graph of the

Figure 3.10: A part of the path graph that represents the elements in the World War II timeline shown in Figure 3.9.A.

coffee diagram. We used a tree to keep related concepts together by connecting them to a single parent. The root and group parents may also be connected to related Wikipedia pages. We have no restriction on the depth of the tree. Notice that the path graph is not suitable to represent the coffee diagram even though part of the diagram looks like a sequence of concepts. Concepts in the same group are equally related (i.e., distance in the same group is irrelevant). For example, the espresso machine is equally related to all coffee pots and machines.

The last graph class in the universal model is the generic class, which can take on any structure. We used this graph to represent diagrams that did not show any common patterns. In Figures 3.14 and 3.15, we include two examples about programming languages and TV networks, respectively. The graphs look very similar to the diagrams so we omit them. In the TV networks diagram, the diagram has disconnected parts such as the AMC network. Therefore, we add extra edges as we did for the I&A graph shown in Figure 3.8. We connect all nodes in the graph to a central node, C , using 2-edge paths (i.e., there is a 4-edge path between any pair of nodes in the diagram).

Figure 3.11: An excerpt of the JavaScript libraries diagram [107] that we represent using a tree graph. The diagram is a mind map. Mind maps usually arrange concepts in a tree.

Figure 3.12: An excerpt of the coffee diagram [3]. The diagram contains distinct groups of related concepts, such as types of coffee beans, types of coffee drinks, and types of coffee pots and machines. Therefore, we modeled the diagram as a tree graph as shown in Figure 3.13 where each group has a separate parent.

Figure 3.13: A part of a tree graph that represents the coffee diagram where each group of related concepts has a separate parent.

Figure 3.14: An excerpt of the programming languages diagram [97]. This diagram is represented using a generic graph.

We note here that the four graph classes that we introduced are only based on the diagrams that we encountered and the ones that we studied in detail. Other graph classes for diagrams may exist. Those classes can be identified and studied in future MDN research.

Figure 3.15: An excerpt of the TV networks diagram [30]. This diagram is represented using a generic graph. The AMC network part is not connected to the rest of the diagram. Similar to Figure 3.8, we make the graph connected by adding a central node C that has a 2-edge path to every node in the graph.

Throughout our work with universal-model diagrams, we used the strategy of minimizing the distance between related elements while also maximizing the distance between unrelated elements. For instance, in the coffee tree, the distance between two children under the same parent is two while the distance between children of different parents is four. The same strategy is built-in in the path and the generic classes. In the cases where we add a central node C to get a connected graph, we keep unrelated elements connected but with a 4-edge path while related elements are reachable by shorter paths. We expect this strategy to be helpful in D2C queries. In addition to traversing from a selected node to Wikipedia, the traversal can go through close and related visual elements in the diagram, which maximizes the chance of finding related pages in Wikipedia.

In the following two sections, we describe how we connect universal-model dia-

grams to Wikipedia. In Section 3.3.1, we start with an approach that uses a directed unweighted graph. The main purpose of this graph is to find Wikipedia pages related to a diagram (the graph is used to crawl Wikipedia starting from a particular diagram). In Section 3.3.2, we explain the main approach, which uses an undirected weighted graph. This graph is used in our evaluation.

3.3.1 Directed Unweighted Graph for Universal-Model Diagrams and Wikipedia Pages

In this graph, we use directed edges to represent connections in diagrams and Wikipedia, as well as connections that link diagram elements with Wikipedia pages. In Figure 3.16.A, we show an example of linking a diagram to Wikipedia. We use a pair of directed edges to represent all internal connections in the diagram and also all connections between the diagram elements and related Wikipedia pages. For internal connections in Wikipedia, we add a directed edge from a page p_1 to a page p_2 if there is an outgoing hyperlink from p_1 to p_2 in Wikipedia.

The directed graph is used to find Wikipedia pages related to a diagram. Note that we do not consider incoming hyperlinks in Wikipedia. Therefore, it is possible to crawl Wikipedia starting from a diagram, and then just follow outgoing hyperlinks from any page. With this graph, we do a crawling on demand using both the PPR and GES algorithms. Any page an algorithm needs to traverse to is crawled. We do this crawling-on-demand process using the two algorithms to retrieve and rank Wikipedia pages related to a specific diagram. We use the the top N ranked pages returned from both the PPR and GES algorithms to build a set R of Wikipedia pages related to a diagram. Note that we repeat the crawling process multiple times starting from different diagram elements (i.e., we issue multiple D2C queries). Therefore the size of R is much larger than N , since we collect the top N pages in multiple ranked lists.

We use the set R to build an undirected graph that we use in our main experiments. We describe the structure of the undirected graph in the next Section and we explain the undirected graph preparation in more detail in Chapter 7.

Figure 3.16: Part A: The modeling of a universal-model diagram and Wikipedia pages using a directed unweighted graph. Part B: The modeling of a universal-model diagram and a subset of Wikipedia pages (the set R) using an undirected weighted graph. Note that we include only edges where both pages connected by the edge are in R (e.g., edge connecting pages X and Y is not included in the graph since Y is not in the set R).

3.3.2 Undirected Weighted Graph for Universal-Model Diagrams and Wikipedia Pages

As we mentioned in the previous section, for each diagram we create a set R that contains related Wikipedia pages to that diagram. Compared to TU (target universe) that we use for an I&A-Model diagram that contains only item pages, the set R contains diverse Wikipedia pages (any articles or categories). Also, R is typically much larger than TU. While TU usually contains a few hundred to a few thousand pages we planned to include 100,000 pages in R for each diagram.

We build an undirected weighted graph (V, E) , where V includes all diagram elements and all Wikipedia pages in R . An edge e between Wikipedia pages $p_1, p_2 \in R$ is included in E , if p_1 links to p_2 , p_2 links to p_1 , or both pages link to each other (i.e.,

p_1 and p_2 have a reciprocal link). As with the I&A-model graph, we assign maximum weight to reciprocal links (weight = 1) in Wikipedia and experiment with different weights for nonreciprocal links ($0 < \text{weight} \leq 1$). Figure 3.16.B shows an example graph. Note that the edge between pages X and Y is not included in E since Y is not in the set R (i.e., Y is not closely related to the diagram and therefore not included in R).

We also assign maximum weight to all connections in the diagram and to all connections between the diagram elements and Wikipedia pages. Note that for those connections (in diagrams and between diagrams and Wikipedia), we used the same weights to simplify the system and limit the number of variables that we have to study and tune in the experiments. A more advanced MDN system can use different weights for those connections to represent different levels of relatedness between nodes or different edge types. The same extension also applies to I&A-Model diagrams.

3.3.3 Universal-Model Support for D2C Queries

As with I&A-model diagrams, we expect universal-model diagrams to help our ranking algorithms in retrieving Wikipedia pages related to a selected element in a D2C query. In contrast to the I&A model, we do not study the aspect-influenced ranking in universal-model diagrams. We do expect that aspect-influenced ranking can be evaluated when two or more universal-model diagrams that contain different (or mostly different) sets of elements. For example, a pair of diagrams can focus on two related courses (possibly provided as a sequence and share a few concepts). Another pair of diagrams can focus on two slightly related historic eras (e.g., a diagram about the American Civil War and another about American history in general). Even if the diagrams contain a common element and this element is selected in a D2C query, traversing from each diagram is likely to arrive at different Wikipedia pages leading

to different rankings.

4 Existing Algorithms: Personalized PageRank (PPR)

In addition to diagram models, the second fundamental element in our research is studying ranking algorithms for MDN queries. In this chapter, we start with a popular ranking algorithm, Personalized PageRank (PPR) [58,84]. The next chapter will cover Greedy Energy Spreading (GES), an algorithm that we designed.

The main reason we chose PPR is its prior use in similar applications to MDN. For systems based on the hyperlink graph of Wikipedia, Agirre et. al. [11] achieved the state-of-the-art performance using PPR for both Named Entity Disambiguation (NED) and Word Relatedness applications. Both applications have similarity to MDN, as we discuss in related work.

We start this chapter by describing the PageRank and the Personalized PageRank algorithms in Sections 4.1 and 4.2, respectively. In Section 4.3, we explain how we use PPR in MDN D2C queries. In Section 4.4, we explain a fast algorithm to approximate PPR that we used in one of our experiments.

4.1 The PageRank Algorithm

PageRank is an algorithm to rank the quality of web pages that was introduced by Brin and Page to power the Google web search engine [27]. In PageRank, the rank (or score) of a page is determined by the number of links it has from other web pages as well as the rank of those web pages. Consider a directed graph of web pages $G = (V, E)$, where E is a subset of $V \times V$, and $|V| = N$. The algorithm

starts by assigning an arbitrary rank $PR(v)$ to all pages $v \in V$ (e.g., by assigning the same rank to all pages, $PR(v) = 1/N, v \in V$). Then, for a sufficient number of iterations (explained later), the rank $PR(v)$ for all pages in the graph is updated in each iteration using the following equation:¹

$$PR(v) = \frac{d}{N} + (1 - d) * \left(\frac{PR(u_1)}{O(u_1)} + \frac{PR(u_2)}{O(u_2)} + \cdots + \frac{PR(u_m)}{O(u_m)} \right). \quad (4.1)$$

where v has an incoming link from $u_i, 1 \leq i \leq m$, $PR(u_i)$ is the rank of the page u_i , and $O(u_i)$ is the number of outgoing links from u_i . The variable d is the *damping factor* and it is frequently set to 0.15. We explain d in more detail shortly.

The concept behind PageRank is that important pages are more likely to be referenced than less important pages. Therefore, each incoming link is considered as an endorsement or a vote from another page. Moreover, an endorsement coming from an important page with a high rank is given more influence than low-rank pages as each incoming link is weighted by a fraction of the rank of its source page.

In Figure 4.1, we show an example of computing ranks for a graph of four nodes (or pages). We start with equal ranks for all nodes and we recompute the ranks for 14 iterations. Note that page ranks converge in the last few iterations. Page 3 has the highest rank since it is the only page with three incoming links.

The PageRank computation is independent of search queries. In practice, scores of the PageRank algorithm can be combined with scores from text-matching techniques to generate a final ranking of search results. For example, we can break the tie

¹A different version of the equation, shown in the Google Search Engine paper [27], includes only d on the left side of the leftmost addition (it does not divide d by N). The version we show here is consistent with the matrix multiplication method we explain later in this section. Moreover, the sum of ranks of all pages is equal to one matching, the probabilistic point of view we explain later. The equation included in [27] requires a normalization step for the final page ranks (to make the sum = 1).

Figure 4.1: Part A: An example graph. Part B: Starting with equal scores, we compute PageRank algorithm for 14 iterations. Page scores converge in the last few iterations.

when two pages equally match the search query by boosting the page with the higher PageRank score.

The PageRank score can also be interpreted using the *random surfer model*. Assume a surfer starts at a random page and then surfs the web by randomly following a hyperlink in the current page at each step. After a sufficiently large number of steps, the rank of each page is equal to the probability that the random surfer is at the page at any time. Therefore, PageRank generates a probability distribution over the web pages where the sum of the scores of all pages is equal to one.

One problem PageRank had to deal with is pages with no outgoing links. Brin and Page [84] call those pages the dangling nodes. Those pages behave as rank sinks or traps as they have no way to distribute their rank [84]. The same problem can also happen when a set of pages have no outgoing links to the rest of the graph. To solve this problem, PageRank includes the damping factor d . The solution is to have each node distribute part of its score among all nodes in the graph. From the point of view of the random-surfer model, in addition to following a hyperlink in the current page with probability $1 - d$, we assume that with probability d the surfer jumps to a

random page in the graph (e.g., by typing a random url in the browser). Therefore, each page in the graph has new incoming links from all pages in the graph. The total score of these additional edges is equal to $\sum_{v \in V} \frac{1}{N} * d * PR(v) = \frac{d}{N}$, which is the leftmost term in Equation 4.1.

$$\begin{array}{c}
 \begin{array}{ccc}
 & A & \\
 (1-d) & \begin{bmatrix} 0 & 0.5 & 0 & 0.5 \\ 0.33 & 0 & 0.5 & 0 \\ 0.33 & 0.5 & 0 & 0.5 \\ 0.33 & 0 & 0.5 & 0 \end{bmatrix} & \\
 0.85 * & &
 \end{array}
 + d * \begin{array}{ccc}
 & B & \\
 & \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix} & \\
 0.15 * & &
 \end{array}
 = \begin{array}{ccc}
 & M & \\
 & \begin{bmatrix} 0.038 & 0.462 & 0.038 & 0.462 \\ 0.321 & 0.038 & 0.462 & 0.038 \\ 0.321 & 0.462 & 0.038 & 0.462 \\ 0.321 & 0.038 & 0.462 & 0.038 \end{bmatrix} &
 \end{array}
 \end{array}$$

$$\begin{array}{ccc}
 & M & \\
 & \begin{bmatrix} 0.038 & 0.462 & 0.038 & 0.462 \\ 0.321 & 0.038 & 0.462 & 0.038 \\ 0.321 & 0.462 & 0.038 & 0.462 \\ 0.321 & 0.038 & 0.462 & 0.038 \end{bmatrix} & \\
 & * & \\
 & \begin{array}{ccc}
 P_0 & & P_1 \\
 \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix} & = & \begin{bmatrix} 0.250 \\ 0.215 \\ 0.321 \\ 0.215 \end{bmatrix}
 \end{array}
 \end{array}$$

Figure 4.2: The first line shows a transition matrix M of a random surfer on the graph in Figure 4.1 using damping factor $d = 0.15$. M combines a probability $(d - 1)$ of following one of the edges on the graph (matrix A) and a probability d of jumping to a random page (matrix B). The second line shows one iteration of PageRank where we multiply M by the initial PageRank vector P_0 to get the PageRank vector P_1 which shows the probability of the random surfer to be at each page after one step of surfing the graph.

We now explain the example in Figure 4.1 from the perspective of the random surfer model. We represent the graph in Figure 4.1 as a transition matrix A , shown in Figure 4.2, with size $N * N$, where the rows and columns correspond to the nodes in the graph. Column j shows probabilities of the random surfer following the different outgoing links from node $v_j \in V$. We assign $1/O(v_j)$ to $A_{i,j}$ if there is an outgoing link from v_j to v_i , otherwise, we assign 0. We represent the damping factor part as $d * B$, where B is an $N * N$ matrix with all entries equals to $1/N$. The final transition matrix $M = (1 - d) * A + d * B$. We also represent the ranks of all pages as the PageRank vector P . In Figure 4.2, we assign equal scores in the initial PageRank

vector P_0 (just like we did in the Figure 4.1) meaning that the all pages have an equal probability to be the starting node for the random surfer. One iteration of the PageRank computation is equivalent to the matrix multiplication $P_{i+1} = M * P_i$, where P_0 is the initial distribution of the PageRank vector. For example, $P_1 = M * P_0$, shown in Figure 4.2, where P_1 contains the probability that the random surfer will be at each page after one step. After a sufficient number of iterations (i.e., a sufficient number of random-surfer steps), the PageRank vector converges and we reach a stationary distribution P_f . This stationary distribution is the eigenvector of the eigenvalue $c = 1$, for $M * P_f = cP_f$. In reality, we only need a relatively few iterations to reach a vector that is very close to the stationary distribution. The number of iterations was 52 for a web graph that included 322 million links [84].

We can rewrite the equation $P_{i+1} = M * P_i$ in Figure 4.2 as $P_{i+1} = ((1 - d) * A * P_i) + (d * B * P_i)$. The final form of the equation that we use in our experiments is the following:

$$P_{i+1} = (1 - d) * A * P_i + d * TV \quad (4.2)$$

where TV is called the teleport vector. For the example above, TV is the vector $[1/N, 1/N, \dots, 1/N]$, meaning that all pages have equal chances for being selected when the random surfer decides to jump to a random page. In the next section, we describe the Personalized PageRank algorithm that changes the teleport vector to bias the ranking towards a specific set of pages.

4.2 The Personalized PageRank Algorithm

In the PageRank algorithm, the damping factor was used to address the case of rank sinks. The damping factor turned out to be an important parameter that allowed using PageRank in various applications. In Personalized PageRank (PPR) [58, 84],

instead of jumping to a random page with probability d , the algorithm jumps or teleports to a specific set of pages that we call the *base set* [18]. As a result, the PageRank is biased towards the topic of the pages in the base set as the random surfer is likely to stay in their vicinity. For example, Brin and Page [84] used the page of John McCarthy, a famous Computer Scientist, as a base set with a single page. The result ranking showed higher ranks for pages related to Computer Science.

To update the teleport vector TV in Equation 4.2 to reflect a bias towards a base set BS , we set $v_i = 1/|BS|$ if $v_i \in BS$, otherwise, we set $v_i = 0$. Consider the example graph discussed in Figures 4.1 and 4.2. We use a base set that includes only node 2. We set $TV = [0, 1, 0, 0]^T$ in Equation 4.2. The resulting PageRank vector is $[0.216, 0.329, 0.277, 0.179]^T$. In contrast to the PageRank vector shown in Figure 4.2, Node 2 is the top ranked page now. In addition, Node 1 is ranked higher than Node 4 as the former has a link from Node 2 but the latter does not.

The idea of PPR was introduced in Brin and Page [84] but was not fully explored. Haveliwala [58] provided a detailed study of using PPR to produce topic-sensitive PageRank. Instead of computing a universal PageRank vector for the web, the introduced approach computed multiple PageRank vectors that represent different topics. Then, by knowing the topic of the search keywords, the appropriate PageRank vector is used.

PPR has been widely used in different applications including providing recommendations in Twitter [16]; predicting human movement in an urban environment [63], and analyzing protein-interaction networks [60]. The closest to ours is by Agirre et al. [11] for using PPR in Word Relatedness and Named Entity Disambiguation. Those researchers used the graph of Wikipedia and achieved state-of-the-art performance for the two applications. We build on their approach and also investigate other alter-

natives. We explain our approach in more detail in the next section, where we show how we use PPR to rank Wikipedia pages in MDN D2C queries.

4.3 Using PPR for MDN D2C Queries

Given a D2C query, we use a base set that contains only the selected element in the diagram. From the perspective of the random-surfer model, the surfer goes back to the selected element in the diagram with probability d . Then, the surfer traverses to Wikipedia directly from the selected diagram element and also through related elements in the diagram (most likely through elements close to the selected element). Consequently, we expect the surfer to frequently visit Wikipedia pages related to the selected element in the diagram. The final PageRank vector is used to rank Wikipedia pages.

In Figure 4.3.A, we connect a diagram containing three elements to the graph we used in the last section. We use *node* as a term for all graph vertices, including the three diagram elements and the four pages. We use an undirected-weighted graph, just as we do in our main experiments, and we assign equal edge weights for simplicity. Assuming Node 10 is selected in a D2C query, we show the returned ranked list of graph nodes below the graph. An MDN system can omit diagram elements from the ranking and return only the Wikipedia pages.

Before discussing the ranked list returned by the previous example, we discuss our intuition of what properties related Wikipedia pages have in the context of D2C queries. We hypothesize that compared to unrelated (or less related) pages, related pages have a larger number of shorter paths to the selected element. A desirable feature of the ranking algorithm used in D2C queries is the ability to tune the importance of these two properties, namely, the number and length of paths to a selected element.

Figure 4.3: Part A: An example diagram with three elements connected to four pages in Wikipedia. We use the term *node* to denote both elements and pages. We show a ranked list of diagram elements and Wikipedia pages when using PPR with a damping factor $d = 0.15$. Part B: Another example where a page with a high degree is ranked higher than pages connected directly to the diagram when setting $d = 0.15$. However, increasing d to 0.5 boosts Pages 1 and 2, which are closer to the diagram. Part C: Showing that a graph node can be also affected by the degree of nodes it connects to. Page 1 is ranked higher than Pages 2 and 3 even though each one of the three pages has a similar path to the selected element in the diagram.

Looking back at the example in Figure 4.3.A, the list returned by PPR satisfies our view to a certain extent. Pages 1 and 2 have the top two spots given their proximity to the selected element. Page 3 has two paths to the selected element, and therefore, it is ranked before Page 4. A minor issue is that Page 1 has a higher rank than Page 2 even though both pages have the same connectivity to element 10 in the diagram. The reason for Page 1 superiority is its higher node degree compared to Page 2. As the random surfer goes back and forth between the diagram and Wikipedia, he has a larger probability of landing at Page 1. One can also view Page 1's higher rank as getting feedback from Wikipedia in addition to the diagram. Page 1 benefits

from its extra link to Page 4, which may be considered related to the diagram to some extent. We denote the effect of page degrees in Wikipedia as *the influence of Wikipedia structure* on the ranking process.

Let us consider another example about the influence of Wikipedia structure. In Figure 4.3.B, we show a graph similar to the previous one, but with an extra central Page 5. Computing PPR with $d = 0.15$ ranks Page 5 higher than Pages 1 and 2, which have shorter paths to the selected element in the diagram. One can interpret ranking Page 5 first as PPR rewarding multiple paths (two 3-edge paths from Element 10 to Page 5) more than a single short path (e.g., one 2-edge path from Element 10 to Page 1). However, it is still possible that a node can get some rank boost by virtue of its high degree and not by only its strong connectivity to the selected element in the diagram. Recall that PageRank is designed to reward nodes with high degree.

To boost diagram influence and also decrease Wikipedia influence, we can adjust the main PageRank parameter, the damping factor d . In the last two examples, we set $d = 0.15$. In Figure 4.3.B, we show another ranked list when d is set to 0.5. This value forces the random surfer to go back more frequently to the selected element in the diagram. As a result, pages close to the diagram are ranked higher. In the example, Pages 1 and 2 are now ranked before Page 5. In our evaluation in Chapters 6 and 7, we show that a large damping factor benefits PPR for both I&A-model and Universal-model diagrams.

A node can also be affected by the degree of the nodes it connects to. Consider the example in Figure 4.3.C. Even though Pages 1, 2, and 3 have the same connectivity to Element 10 (2-edge path to Element 10), Page 1 gets a higher score than Pages 2 and 3. The reason for the difference is that Page 1 gets 0.425% of the visits when the random surfer is at Element 11 while the other 0.425% are visits back to Element 10.

(The remaining 0.15% of the visits, which represent the share of the damping factor part in Eq. 4.2, are directed to Element 10.) On the other hand, Pages 2 and 3 each get only about 0.28% of the visits when the random surfer is at Element 12. In this example, increasing the damping factor does not change the ranking.

We summarize our discussion of the use of PPR in MDN. PPR can be useful in ranking Wikipedia pages in MDN D2C queries. However, PPR’s sensitivity to node degrees may have a negative impact. Given that the majority of graph nodes are Wikipedia pages, the impact of page degrees (or the influence of Wikipedia structure) can be more influential than the influence of the diagram. On the one hand, this Wikipedia influence may mean important (or hub) pages in Wikipedia, with a high degree, get a higher rank along with other pages they connect to. On the other hand, large Wikipedia influence may minimize the influence of the diagram. Changing the damping factor can be used to choose which alternative should have more weight in the ranking process. We also pursue a different approach by designing GES, an algorithm that minimizes the effect of node degrees in the ranking process. We compare the performance of PPR and GES in both I&A-model diagrams and Universal-Model diagrams. We also report on the performance of choosing different values for the damping factor for PPR.

4.4 Monte Carlo Method for PageRank

In our main experiment for Universal-Model diagrams, we used a separate graph for each diagram. We built an undirected-weighted graph that included about 100,000 Wikipedia pages related to each diagram. In order to collect those pages, we ran multiple D2C queries using both GES and PPR algorithms and crawled Wikipedia pages close to each diagram. We used the ranked lists returned by both GES and PPR algorithms to build the undirected-weighted graph as we explain in more detail

in Section 3.3.2.

Algorithm 2

Function: $PPR-MC(s, t, d)$

Input: s is a selected element in a D2C query, t is the number of steps (visits) for the random surfer, d is the damping factor.

Output: a list of Wikipedia pages visited (crawled) by the random surfer ranked by their relevance to e .

1. $VB \leftarrow$ an empty bag (a relational database table).
 2. $n \leftarrow s$ // current node n is set to the selected element in the D2C query.
 3. **for** $k \leftarrow 1$ to t **do** // number of steps (visits) for the random surfer .
 4. $r \leftarrow$ a real random number, where $0 < r \leq 1$
 5. **if** $r \leq d$ **then**
 6. $n \leftarrow s$ // return to the selected element.
 7. **else**
 8. Crawl to page n and extract its outgoing links.
 9. $\triangleright$ We fetch a previously crawled page from the database.
 10. Randomly select a page x where n links to x .
 11. $n \leftarrow x$ // the random surfer jumped to a new page.
 12. Store n in VB
 - 13.
 14. $L \leftarrow$ an empty list of $(page-id, score)$ tuples.
 15. $VS \leftarrow$ The set of pages in VB
 16. **for** each page p in VS **do**
 17. Insert $(p, \frac{cp}{|VB|})$ in L , where cp is the count of p in VB .
 18. sort L by $score$ in a descending order.
 19. return L .
-

Given that we did not have the whole graph before crawling, using PPR was not possible (since we did not have the transition matrix A in Equation 4.2). Therefore, we used a Monte-Carlo method, $PPR-MC$, to compute a fast approximation of PPR that only needs on-demand crawling. In D2C queries, we start from a selected element in a diagram and for a number of steps (visits) we crawl to a diagram element or a page that the random surfer picks randomly. We use a variation of Algorithm 4 from Avrachenkov et al. [15].

We show a pseudo-code for the $PPR-MC$ method in Algorithm 2. The concept

is to simulate (run) a sufficient number of steps (visits) of the random surfer process (line 3 in Algorithm 2) and keep track of visited pages (line 12). Then, the rank of a page is the number of times the surfer visited the page divided by the total number of visitations to all visited nodes (line 17).

The algorithm in Avrachenkov et al. [15] starts a fixed number of times from each page in the graph. We modify the algorithm to start at a selected element in a D2C query (line 2). We also consider graph nodes to include diagram elements and visited Wikipedia pages. Early experiments showed that this method had reasonable performance. In general, we aim to use PPR-MC only to collect pages to build the undirected-weighted graph and not to rank D2C query results in our main experiments.

This section marks the end of our introduction to Personalized PageRank algorithm. In the next chapter, we introduce Greedy Energy Spreading, an algorithm we designed to increase the influence of diagrams on the ranking process.

5 The Greedy Energy Spreading Algorithm

In the previous chapter, we discussed the possible influence of the Wikipedia graph structure on the ranking process and how that may overshadow diagram influence. We also described that the iterative nature of PPR may consider feedback from Wikipedia in addition to the information (visits) coming from the diagram. In this chapter, we introduce the Greedy Energy Spreading (GES) Algorithm where we try to address these two issues. We introduce the concept of energy spreading in Section 5.1. We describe GES in Section 5.2, including energy spreading in diagrams in Section 5.2.1, and energy spreading in Wikipedia in Section 5.2.2.

5.1 Energy Spreading

Energy spreading ES (or *spreading activation* [35]) has been used in information retrieval (IR) since the early 1980s [35]. The ES method represents documents in a collection or terms in a thesaurus as nodes in a graph (or an *associative network* [35]). Edges in the graph can represent relationships between the nodes such as bibliographic citations or thesaurus-like relationships [35]. To retrieve related nodes, ES spreads energy from a node or nodes of interest and then ranks the relevance of the other graph nodes by the total energy each receives. Recent research has used ES methods on various tasks in IR including movie recommendation [41] and concept relatedness [51]. The latter used the graph of Wikipedia hyperlinks.

5.2 The Greedy Energy Spreading (GES) Algorithm

We use energy spreading to rank Wikipedia pages in D2C queries. We spread energy from a selected element in a diagram to Wikipedia pages. We rank Wikipedia pages by the total amount of energy each receives. Our approach has two distinctive properties when compared to PPR. The first property is spreading energy from the diagram to Wikipedia in a Breadth First Search (BFS) approach using one pass, which is different from the iterative approach in PPR. The second property is reducing sensitivity to node degree by computing node final energies using *Series Adjusted Summation* (SAS), a method that we introduce later in this chapter. In addition to these properties, we also use greediness by allowing only high-energy nodes to spread energy in the graph. We explain the greediness in our algorithm later in this chapter.

5.2.1 GES-Diagram: Propagating Energy in the Diagram

Spreading energy in the diagram is based on BFS starting at the selected element in a D2C query. Each node's energy depends on the length of its shortest path (or distance) to the selected element. For example, consider the energy (or score) distribution array $Dist1[1 \cdots 10] = [1, 1/2, 1/4, 1/8, \dots, 1/(2)^{10}]$ for the distances one to ten. We assume the longest distance between a diagram element and a D2C selected element is ten and the selected element is at distance zero. Then, a node with distance x gets the energy $Dist1[x]$.

We show pseudo-code for the method in Algorithm 3. We assign distance zero to the selected element s (Line 2). We increment the current distance (e.g., zero for element s) and assign it to elements connected to the current element (Line 13). Every element in the diagram gets its energy based on its distance (Line 14). We also spread the same element energy to its connected Wikipedia pages (Line 18). Note

Algorithm 3 Greedy Energy Spreading (GES) in the Diagram

Function: GES-DIAGRAM($Dist, G, s$)

Input: $Dist$ is an array of scores that will be assigned to elements based on their distances from the selected element, G is the graph of diagram elements and Wikipedia pages, and s is the selected element in the diagram.

Output: a list of Wikipedia pages in Level 1 with their scores.

1. $L1 \leftarrow$ an empty list.
 2. $s.distance \leftarrow 0$
 3. $Q \leftarrow$ an empty queue.
 4. $Visited \leftarrow$ an empty set.
 5. $Q.push(s)$.
 6. $Visited.add(s)$
 7. **while** Q is not empty **do**
 8. $u \leftarrow Q.pop$
 9. $D \leftarrow$ an empty set.
 10. $D \leftarrow$ diagram elements connected to u in the graph G .
 11. **for** each $e \in D$ **do**
 12. **if** $e \notin Visited$ **then**
 13. $e.distance \leftarrow u.distance + 1$
 14. $e.energy \leftarrow Dist[e.distance]$
 15. $W \leftarrow$ an empty set.
 16. $W \leftarrow$ Wikipedia pages connected to e in the graph G .
 17. **for** each $p \in W$ **do**
 18. $p.energy \leftarrow e.energy$
 19. Insert p in $L1$
 20. **return** $L1$.
-

that if an element has multiple paths to a D2C selected element (i.e., receive multiple energies), only the the shortest path is considered (i.e., count only the highest received energy) (Line 12). In Figure 5.3, we show an example energy spreading in a diagram modeled using a tree graph. We assume Node 3 is selected in a D2C query, and we show scores next to diagram elements and $L1$ Wikipedia pages. We continue with the same example in the next section.

The energy-distribution array reflects the decay in energy as we travel further from the selected element. The array $Dist1$ represents a sharp decay in energy that powers only elements close to the selected element. This distribution reduces the set

of Wikipedia pages that get high energy. Therefore, only the selected element and its close neighbors will have a significant effect on the ranking. Using a different distribution array with a gradual decay, such as $Dist3[1..10] = [1, 0.9, 0.81, 0.729, 0.656, 0.590, 0.531, \dots, 0.349]$, has a different effect on the ranking process. Even an element with distance six gets about half of the energy of the selected element. In Figure 5.1, we show five different distributions that we used in our experiments. In the experiments we have conducted for Universal-Model diagrams, we test how the choice of energy distribution array in diagrams affects the ranking. For I&A diagrams experiments, we only use $Dist5$, since the diagram contains only attributes related and connected to a selected element; we provide more details in Chapter 6.

Figure 5.1: Different distributions of energy for diagram elements based on their distance (or shortest path) from a selected element in a diagram-to-content (D2C) query. Dist1 has a sharp decay in energy as we travel further from the selected element compared to the gradual decay in Dist2 and Dist3. Dist4 starts with a gradual decay in the beginning, and then decreases sharply. Dist5 assigns the same energy to all diagram elements regardless of their distance.

5.2.2 GES-Wikipedia: Propagating Energy in Wikipedia

GES-DIAGRAM spreads energy from a D2C selected element to L1 pages. GES-WIKIPEDIA continues the spreading of energy to other levels using the same BFS approach. We first describe how we compute a node energy from energies received

from other nodes.

One of the properties we wanted in GES is reduced sensitivity to node degrees. First, when a node spreads energy, it spreads all of its energy (current score) to all other nodes it connects to regardless of the number of those nodes. The approach is different from PPR, where the probability of jumping to the next node depends on the number of connected nodes.

The spreading approach is simple, but accumulating *received energies* (REs) to compute a node energy is more complicated. Recall that our intuition about related pages is that they have shorter and a larger number of paths to the selected element. Therefore, we wanted a function that rewards nodes with both high REs and a high number of REs. For example, averaging REs does not reward a high number of REs. Consider the nodes a and b with REs $(1, 1)$ and $(1, 1, 0.7)$, respectively. Averaging REs results in node a having a higher energy than b even though b has an additional RE. Summing on the other hand can reward nodes with low REs. Consider node c having 21 REs with each equal to 0.1. Summing REs ranks node c higher than node a even though c is only connected to marginal nodes.

We introduce *Series Adjusted Summation* (SAS) to accumulate REs. The concept is to give more weight to higher REs. Therefore, even if a node had a large number of low REs, those REs would have a small effect on the node's final energy. We sort REs in a non-increasing order and weight each RE using terms from a decreasing series, such as the geometric series $S1 = (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots, (\frac{1}{2})^{20})$. For a list of received energies $LRE = (e_1, e_2, \dots, e_n)$, where LRE is in a non-increasing order, and a decreasing series $S = (t_1, t_2, \dots, t_n)$, we formally define *SAS* in the following equation:

$$SAS(LRE, S) = \sum_{i=1}^n e_i * t_i \quad (5.1)$$

SAS discounts low REs by multiplying them with the small terms in the series tail. We truncate the SAS computation after some point n . Note that the loss of precision by doing so is relatively low given that tails of both S and LRE contain relatively smaller numbers. In most of our experiments, we set n between 15 and 20. These numbers proved sufficient in early experiments. Using *SAS* with $S1$ in the previous example yields the ranking: b , a , then c , with scores 0.84, 0.75, and 0.1, respectively. This ranking is a better estimate of the connectivity to relevant nodes in our view.

Figure 5.2: Different series used to compute and tune Series Adjusted Summation (SAS).

One feature of SAS is the ability to tune the computation by using different series. For example, using $S1$ leads to only the top REs having a significant effect on the node energy (i.e., reward high REs more than high number of REs). On the other hand, using a gradually declining series, such as $S3 = (\frac{20}{210}, \frac{19}{210}, \frac{18}{210}, \dots, \frac{1}{210})$ distributes the effect over more REs.

In Figure 5.2, we show six series we used in our experiments. Note that $S4$ is equivalent to averaging the highest n REs, as it assigns the same weight to all REs. $S4$ can be considered as a baseline for SAS. Also, for the series $S1$ to $S4$, the sum of

the terms in each series is equal to one. Therefore, for those series, nodes in the graph never accumulate energy larger than the full energy that we started with (i.e., if we started with full energy equals to one, all nodes in the graph will have energy equal to one or less). However, the sum of the terms in each of $S5$ and $S6$ is greater than one (1.6 in $S5$ and 2.7 in $S6$). With these totals it is possible for nodes in the graph to exceed the starting full score. In this case, we can consider the new maximum score in the graph as the full score. Recall that we are only interested in ranking, therefore changing the full score has no effects in our application. The use of $S5$ and $S6$ turned out to be effective in D2C ranking for Universal-Mode diagrams, as we describe in Chapter 7.

The last important property of GES is greediness in propagating energy. Given that we attenuate low REs or even discard them, it is most likely useless to spread energy from nodes with low energy (score). At each stage of energy spreading in Wikipedia, described in more detail shortly, we only allow pages whose energy is greater than a specific threshold to spread energy. As we limit energy spreading in the graph, we reduce the number of REs each node receives. Consequently, we reduce the cost of sorting REs when computing a node scores using SAS.

Algorithm 4 Greedy Energy Spreading (GES) in Wikipedia

Function: GES-WIKIPEDIA($G, L1, srs, ll, we, F, n$)

Input: $L1$ is a list of Level1 pages (a page is an object with an id, energy, and level attributes), srs is the series used to combine page energies, ll is the last level to send energy to, G is the graph of diagram elements and Wikipedia pages, we is weights of the graph edges, F is the greediness factor, and n is the number of REs to consider in computing SAS.

Output: a ranked list of Wikipedia pages with their scores.

1. $Stages \leftarrow ['1-2', '2-2', '2-3', '3-3', \dots, ll]$.
 2. $Q \leftarrow$ an empty queue.
 3. $finalList \leftarrow$ an empty list of page objects.
 4. sort $L1$ by $page.score$ in a descending order, then insert $L1$ in $finalList$
 5. push $L1$ pages in Q
 - 6.
 7. $RE \leftarrow$ an empty list of ($page-id, received-energy$) tuples.
 8. **for** each $stage \in Stages$ **do**
 9. $p \leftarrow Q.pop$.
 10. $W \leftarrow$ Wikipedia pages connected to p in the graph G .
 11. **for** each $q \in W$ **do**
 12. **if** spreading energy among pages in the same level **then**
 13. **if** $p.level = q.level$ **then**
 14. insert $(q, p.energy * we(p,q))$ in RE^1 .
 15. **else** // spreading energy from a level to the next.
 16. **if** $p.level = q.level + 1$ **then**
 17. insert $(q, p.energy * we(p,q))$ in RE .
 - 18.
 19. sort RE by page-id then by received-energy in a descending order
 20. $L \leftarrow$ an empty list of page objects.
 21. **for** each page $p \in$ (get the set of pages in RE) **do**
 22. compute $p.energy$ using SAS (using top n REs and terms in srs).
 23. insert p in L .
 24. sort L by $p.energy$ then let $energy-threshold \leftarrow$ top energy in $L * F$
 25. add pages p in L to Q , where $p.energy > energy-threshold$.
 26. **if** spreading energy among pages in the same level **then**
 27. $finalList \leftarrow$ merge-sort $finalList$ and L
 28. $RE \leftarrow$ an empty list of page-id, received-energy tuples.
 - 29.
 30. return $finalList$
-

¹In the algorithm we used in our experiment, we also insert $(p, q.energy * we(q,p))$ in RE if q is not in the queue Q (i.e., if $q.energy < energy-threshold$).

We show complete pseudo-code of GES-Wikipedia in Algorithm 4. We describe our algorithm using the example in Figure 5.3, and we also refer to the pseudo-code. We spread energy in Wikipedia in stages (Line 1). A stage $i-j$, where $j = i + 1$ refers to spreading energy between a level L_i and the following one L_j (e.g., spreading energy from L_1 to L_2 is denoted ‘1-2’). A stage $i-i$ refers to spreading energy among nodes within the same level (e.g., spreading energy within L_2 is noted ‘2-2’). The parameter ll (last level) controls which level the GES stops at (Line 1). We do not send energy within L_1 (‘1-1’) to keep the influence of the diagram unchanged. We build the final ranking list, `finalList`, by first inserting L_1 pages in a non-increasing order (Line 4). Then, after spreading energy to a level (e.g., ‘1-2’) and spreading energy within the level (e.g., ‘2-2’), we insert the level pages in `finalList` (Lines 26-27). As a result of computing the greediness threshold, which we explain shortly, we get a list of the pages in the current level sorted by score in a non-increasing order. Therefore, we can use merge sort to combine this list with `finalList`, which is already sorted (Line 27).

Figure 5.3 shows node received energies (REs) and computed scores for all nodes for each stage. We use S_1 to compute SAS and assume all edges have weight one. For example, after spreading energy from L_1 to L_2 (Line 17), Node 21 received the energies 1 and $1/4$ from Nodes 11 and 12, respectively. These REs are sorted in a non-increasing order for each node (Line 19). Using SAS , we compute Node 21 energy as $1 * 1/2 + 1/4 * 1/4 = 0.56$ (Line 22).

At the end of the ‘1-2’ stage, we sort pages in L_2 to get the highest score (Line 24), which we use along with the greediness factor F to compute the energy threshold (*energy-threshold*) (Line 24). In the example, assuming $F = 0.9$, *energy-threshold* = $0.56 * 0.9 = 0.50$. We only allow nodes with scores greater than or

equal *energy-threshold* to propagate energy in the next stage. We insert such nodes in the queue to be processed in the next stage (Line 25). In the example, only Nodes 21 and 22 are inserted in the queue. The F factor can be used to tune the greediness of the algorithm where lower F values allow more energy to spread in the graph. The GES algorithm can use separate F values for each stage. Setting $F > 1$ prohibits spreading energy in a specific stage (e.g., ‘2–2’ or ‘3–3’).

Figure 5.3: An example of spreading energy using GES-Diagram and GES-WIKIPEDIA. We show energies (scores) next to nodes for diagram elements and $L1$ pages. For $L2$ and $L3$ Wikipedia pages, we show how their received energies (REs) and scores change in each stage of energy spreading.

In the ‘2–2’ stage, Nodes 23 and 24 cannot send energy. However, after receiving energy from Node 21, we allow Node 23 to send its energy back (the footnote in Line 14). Note that allowing Node 23 to only receive and not send energy could favor Node 23 in case both nodes had close scores, but only Node 21 was above the threshold. At the end of the ‘2–2’ stage, we recompute node scores using both energies coming from $L1$ and $L2$. Given that node scores change, we recompute *energy-threshold* to know which nodes are allowed to send energy to $L3$. Similar to the previous stage, only Nodes 21 and 22 are allowed to send energy in the ‘2–3’ stage. We show REs and scores for $L3$ pages after processing the ‘2–3’ stage, which we consider as the last stage in this example.

Finally, we show the main structure of GES in Algorithm 5. GES-DIAGRAM returns the energies (scores) of L1 pages in the list $L1$, which we pass to GES-WIKIPEDIA.

Algorithm 5 Greedy Energy Spreading (GES)

Function: $GES(dist, G, s, srs, ll, we, F, n)$

Input: Please refer to GES-DIAGRAM, and GES-WIKIPEDIA for input description.

Output: A list of Wikipedia pages ranked by their relevance to a selected element in a D2C query.

1. $L1 \leftarrow \text{GES-DIAGRAM}(Dist, G, s)$
 2. $RankedList \leftarrow \text{GES-WIKIPEDIA}(G, L1, srs, ll, wl, F, n)$
 3. return $RankedList$
-

In this chapter we described GES, an algorithm that we designed to counter some of the PPR properties that we think may have a negative effect on D2C-query ranking. We presented SAS, a method that we designed to compute node scores that is insensitive to node degrees. Using SAS, we expect that high-degree nodes, especially those in Wikipedia, to have less influence on the ranking. In addition, we described how energy is spread in one pass from the diagram to Wikipedia pages as opposed to the iterative process in PPR. The insensitivity to node degrees and the one-pass spreading techniques used in GES may boost diagram influence on the ranking process. In the next chapter, we talk about our experiment regarding I&A-Model diagrams. One of the things that we study in this experiment is how PPR and GES algorithms perform in ranking D2C-query results.

6 Evaluation with I&A-Model Diagrams

In this chapter, we describe our first investigation with Extended MDN (EMDN). Recall that in our first evaluation, described in Chapter 2, we conducted a user study that focused on the usability and usefulness of the BMDN user interface. In this chapter, we evaluate the ranking returned by the MDN (EMDN) D2C-query type in I&A-model diagrams using both the PPR and GES algorithms. In the next chapter, we describe our second investigation for EMDN, which focuses on Universal-Model diagrams.

We designed the first investigation to answer the following research questions:

- RQ1) Do I&A-model diagrams positively bias the ranking process to retrieve pages related to the selected element in a D2C query? Assuming a user is interested in a diagram, a desirable feature is to retrieve Wikipedia pages related to the diagram and the selected element in a D2C query. We wanted to know whether the content and structure of I&A-model diagrams are useful in this regard.
- RQ2) If the answer to RQ1 is yes, how much impact should the content and structure of the I&A-model diagrams have on the ranking process? We study this research question using these indicators:
 - What is the best damping factor setting for the PPR algorithm? A high damping factor increases the effect of diagrams' content and structure on

the ranking process, while a low damping factor value increases the effect of the Wikipedia’s content and structure. Related work [11,108] that used the Wikipedia collection set the damping factor value to the popular value 0.15. We wanted to know whether this value works well for MDN D2C queries or should be increased.

- How greedy should the GES algorithm be in each stage of energy spreading? Optimal levels of greediness and the number of nodes allowed to propagate energy in each stage of the energy spreading can provide important indications. A large diagram impact translates to GES allowing more nodes in L1 (nodes connected to the diagram) to propagate energy than nodes in L2 (nodes with no direct connections to the diagram).
- RQ3) What are the best weights for reciprocal and nonreciprocal links? Agirre et al. [11] achieved best results in two applications using the PPR algorithm on the Wikipedia graph when using only reciprocal links. We wanted to know whether this graph configuration is also optimal for MDN D2C queries. We also tested a graph configuration where we also keep nonreciprocal links.
- RQ4) What are the best settings for the PPR and GES algorithms in D2C queries issued from I&A-model diagrams? Which algorithm is better? Was the better algorithm robust for all I&A-model diagrams?
- RQ5) In I&A-model diagrams, given that we can show two aspects of a domain by using two diagrams, how different are the rankings we get from these diagrams?

Note that we do not focus on the running time of the PPR and GES algorithms in this study. Our plan is to process D2C queries and store results offline. Then, we plan to use the stored data to process D2C and C2D queries online. Therefore, we

assume that the speed of online queries is not an important factor.

6.1 Methods.

We used 20 diagrams in ten domains, shown in Table 6.1. Each domain contained two diagrams using different attributes. Recall that in I&A-model diagrams, we direct D2C queries to a Target Universe (TU), which is a collection of item pages we harvest from the domain’s categories and List_of pages. For example, in the candy-bars domain, we collected 140 candy-bar pages. As we show in Table 6.1, the size of the TU collection ranged from 140 item pages in the candy-bars domain to 14,736 pages in the landforms domain. While we harvested all item pages from categories, we used CSS selectors to extract item pages (hyperlinks to item pages) from specific parts of List_of pages. For example, sometimes we targeted a specific column in a table or a specific section in a List_of page. We used this technique to get only item pages whose type matches the items in the diagrams. However, the resulting TU collections did include some odd pages. For example, in the world-languages domain, the TU collection included pages about regions and ethnic groups related to world languages, which negatively affected the ranking precision for both the PPR and GES algorithms.

For convenience, In Figure 6.1, we show the same partial graph we showed in Figure 3.7, which includes an example I&A-model diagram (language family) connected to a TU collection. As we describe in Section 3.3.2, we build a partial graph for each D2C query, containing the selected item, related attributes, and pages in the TU collection. Recall that we use an undirected weighted graph where we assign weight 1, which is the maximum weight, to internal connections in the diagram, connections between the diagram and related attribute pages, and all connections to the the TU node. These connections are shown in thicker lines in Figure 6.1. For connections

Table 6.1: The 20 diagrams we used in the I&A-model experiment. The diagram column indicates the diagram’s attributes. We show the size (number of item pages) of the Target Universe (TU) Collection. For the best setting in both the PPR and the GES algorithms, we show the average rating (Rtng) of the top ten pages (1 is best and 3 is worst) in all queries in each diagram. Lighter background colors highlight the best Rtngs. In general, GES was slightly better than PPR in all diagrams with Rtng difference > 0.15 in 7 diagrams. The rightmost column: given the top ten pages $p_1, p_2, \dots, p_{10}$ when selecting an element e in diagram D_1 shown on the second column, we show the average position of the pages p_1 to p_{10} when selecting e on diagram D_2 , where D_1 and D_2 present different aspects of the same domain. The high averages indicate that two diagrams showing two aspects of the same domain produce different rankings.

#	Domain	Diagram	TU Size	Top 10 Avg. Rtng for PPR (0.85, 0.1)	Top 10 Avg. Rtng for GES (2, 0.7, 0.9)	Avg. Position of Top 10 Pages When Using the Other Diagram
1	Candy bars	Company	140	1.14	1.10	39
2	Candy bars	Ingredients	140	1.59	1.52	36
3	Car models	Make	1100	1.03	1.01	158
4	Car models	Size	1100	1.54	1.47	219
5	Cheese types	Animal milk	620	1.80	1.13	97
6	Cheese types	Country	620	1.50	1.07	169
7	Civil war battles	State	385	1.27	1.22	106
8	Civil war battles	Timeline	385	1.25	1.09	70
9	Database platforms	DB features	211	1.55	1.38	65
10	Database platforms	DB model	211	1.43	1.38	68
11	World languages	Country	1243	1.19	1.10	102
12	World languages	Language family	1243	1.32	1.27	90
13	Animal species	Animal Kingdom	3699	1.90	1.54	640
14	Animal species	Extinction	3699	1.51	1.29	1465
15	Landforms	Country	14736	1.26	1.16	890
16	Landforms	Landform	14736	1.08	1.02	1199
17	Vegetables & Fruits	Minerals	1233	1.62	1.37	92
18	Vegetables & Fruits	Vitamins	1233	1.26	1.25	72
19	World Heritage Sites	Civilization	2223	1.04	1.02	208
20	World Heritage Sites	Country	2223	1.00	1.00	158

Figure 6.1: A graph of a single D2C query including the selected element ‘Dinka’ and its related attributes. The target universe node TU connects the selected element to all Wikipedia pages (items) harvested from related categories and List_of pages.

between attributes and TU pages, among attributes, and among TU pages, we assign maximum weight to edges between pages that have reciprocal links (one link in each direction) in Wikipedia. We test different weights for nonreciprocal links, where two pages have one link (in one direction) in Wikipedia.

We developed a Wikipedia crawler to extract hyperlinks from the attribute and item pages. We also extracted hyperlinks from categories and List_of pages to build the TU collection. We stored the graph edges in a MYSQL-database table for subsequent experiments. We discarded hyperlinks in templates in most of the pages (templates are tables of hyperlinks usually placed in the bottom of Wikipedia articles). Templates can shift the topic of a page to a broader subject. For example, a candy-bar page may include a manufacturer (brand) template with manufacturer-specific hyperlinks. In such cases we only include one hyperlink to the manufacturer’s page. If the same template was found in the manufacturer page, then we include all hyperlinks in the template. A template may also emphasize one aspect of the domain (candy-bar manufacturer in the previous example). In this case, the template may reverse the influence of a diagram that focuses on a different aspect (e.g., candy-bar

ingredients).

We ran 20 D2C queries in each domain (10 queries from each diagram) using both the PPR and GES algorithms. For the most part, we used items that exist in both diagrams to compare how diagrams influence the ranking process (e.g., selecting the ‘Dinka’ language in both the language-forest and language-map diagrams). We chose the selected items randomly and made sure the selected items activate different sets of attributes in the same diagram.

We conducted two experiments. In the first experiment, we tuned the PPR and GES algorithms using 120 queries issued from the first 12 diagrams in Table 6.1. Each query was executed using 45 PPR and 36 GES settings. Then, we used the best few settings in both algorithms in the remaining 80 queries issued from the last 8 diagrams in Table 6.1.

To evaluate the usefulness of diagrams and the algorithms in retrieving related item pages from Wikipedia, a graduate and a senior undergraduate CS student labeled the top ten pages for each query-setting. The graduate student did most of the labeling for the tuning session, including all 12 diagrams for the GES algorithm and 8 diagrams for the PPR algorithm. The undergraduate student labeled results for the remaining 4 diagrams (in the cars and world languages domains) for the PPR algorithm in the tuning session. For these 4 diagrams, we made sure the same criteria (explained shortly) were used for both GES and PPR results. The undergraduate student also labeled the results for both the PPR and GES algorithms in the last 8 diagrams.

In the labeling process, given a D2C query with a selected item, we used the following rating scale for the top ten pages: 1 for a page similar to the selected item, 2 for a slightly similar page, and 3 for a dissimilar page. For example, in the query

where the ‘Dinka’ language is selected in the language-forest diagram, the students used the following criterion. Languages in the same branch (‘Eastern-Sudanic’ in this example) as the selected language are considered similar, languages in the same tree but a different branch are slightly similar, and languages in different trees are dissimilar. We include the students’ labeling criteria for all diagrams in Appendix B. We reviewed the labeled data and discussed conflicts to reach an agreement.

6.1.1 Tuning the PPR Algorithm.

We used the PPR algorithm described in Sections 4.2 and 4.3, where the base set (teleport vector) contains the selected item in a D2C query. We used the graph shown in Figure 6.1. We computed Equation 4.2 for 30 iterations, which is the same number used in related work [11, 108]. We used the L_1 error to monitor the convergence of the PageRank vector, where L_1 error = $\sum_{j=1}^n |v_j|$, where $v_j \in (P_{i+1} - P_i)$, n is the size of the PageRank vectors P_{i+1} and P_i , and i is the iteration number. Computing 30 iterations was sufficient to reach a convergence of the PageRank vector with L_1 error < 0.0001 . That convergence condition is the same as in Yeh et al. [108].

We tested two parameters for the PPR algorithm, the weight of non-reciprocal links $nRcp$ and the damping factor d . We tested the values 0.5, 0.1, 0.066, 0.033, and 0 for the $nRcp$ parameter while using the weight 1 for reciprocal links. Early results indicated that best performance was with a damping factor between 0.05 and 0.1, so the tested values are close to this range. With $nRcp = 0$, PPR was allowed to use only reciprocal links. With $0 < nRcp < 1$, the random surfer is biased towards pages connected to the current page by a reciprocal link, but can still navigate through non-reciprocal links.

The best Wikipedia graph configuration in Agirre et al. [11] included only reciprocal links when using the PPR algorithm. Reciprocal links do increase the confidence

that two pages are in the same domain and they are closely related. However, in MDN, we hypothesized that nonreciprocal links are also important. For instance, the link between an attribute page (e.g., a language family) and a related item page (e.g., a language) in Wikipedia can be of any type (outgoing, incoming, or reciprocal). In addition, using only reciprocal links removes a large portion of the graph edges that could be useful in finding related Wikipedia pages. Therefore, we used weighted edges and gave reciprocal links some advantage.

For the PPR damping factor, we tested the values (0.15, 0.25, 0.35, ..., 0.95). Various studies that used PPR (especially in Wikipedia) used the value 0.15. We tested higher damping factor values, which increase the effect of diagrams on the ranking process, to see whether the PPR algorithm benefits from the context provided in the diagrams.

6.1.2 Tuning the GES Algorithm.

We used the GES algorithm described in Chapter 5. For energy spreading in the diagram, we used *Dist5* shown in Figure 5.1. Given that we rely on attributes for retrieving similar item pages, we use *Dist5* to assign full score (score 1) to attributes related to a selected item. For example, in Figure 6.1, the attributes ‘Eastern-Sudanic’ and ‘Nilo-Saharan’ were assigned a full score. Since we only have two levels for Wikipedia pages (L1 and L2 in Figure 6.1), we let the algorithm run for two stages: stage ‘1-2’ for spreading energy from the TU node and attribute pages, and stage ‘2-2’ for spreading energy among pages in the TU collection.

We did not use greediness in the ‘1-2’ stage as it has no effect. All attribute pages in Level 1 have full score. Therefore, they are allowed to send energy to Level 2 regardless of whether greediness is used or not. For energy spreading in the ‘2-2’ stage, we tested three values for the greediness factor F : 1.1, 0.9, and 0.75. Setting

F to 1.1 did not allow energy spreading in the ‘2–2’ stage. Therefore, page scores depended only on energy received from attribute pages. Using the value 0.75 allowed more energy spreading in the ‘2–2’ stage compared to the value 0.9. With I&A-model diagrams, we use the parameter F to refer only to greediness in the ‘2–2’ stage.

The second parameter we tested for the GES algorithm was the nonreciprocal links weight ($nRcp$). We can think of a link weight in the context of the GES algorithm as a wire conductance when sending energy. We wanted to assign higher weights for links between highly relevant nodes. We assigned weight 1, which is the maximum weight, to all reciprocal links. For nonreciprocal links weight, we tested four values: 1, 0.7, 0.4, and 0. Setting $nRcp$ to 1 gave the same weight for both types of links while setting $nRcp$ to 0 means the GES algorithm uses only reciprocal links. The values 0.7 and 0.4 gave different amounts of preference to reciprocal links.

The third parameter for the GES algorithm is the series that the SAS computation uses. We used three series¹: $S1$, $S2$, and $S3$ shown in Figure 5.2 to study the trade-off between rewarding high received energies and rewarding a high number of received energies. Recall that $S1$ favors the first option while $S3$ favors the second. $S2$ has an intermediate behavior. We also set n to 15 in Algorithm 4 to consider only the top 15 received energies when computing a node energy using SAS. In early experiments, we found this number to be sufficient to achieve high performance for GES.

In the first 120 queries issued in the first 12 diagrams, we ran each query using 36 GES settings, which are all combinations of four $nRcp$ values (1, 0.7, 0.4, and 0), three series ($S1$, $S2$, and $S3$), and three F values (1.1, 0.9, and 0.75). Then, we chose the best two settings to run the remaining 80 queries issued from the last 8 diagrams. We show and discuss the results of these settings in Section 6.2

¹The series $S4$, $S5$, and $S6$, were only introduced and used in the universal-model experiments described in the next chapter.

Table 6.2: Tuning results for 45 PPR settings based on 120 D2C queries issued from 12 diagrams. We show the average similarity rating (Rtng) for the top ten pages in all queries, where $1 \leq \text{Rtng} \leq 3$ (lower values are better). The performance of the PPR algorithm was better with a high damping factor ($d \geq 0.45$), indicating the importance of increasing the effect of diagrams on the ranking process. Similar to GES, PPR achieved the best performance with keeping nonreciprocal links but giving reciprocal links more importance (nRcp values: 0.1, 0.06, and 0.03).

		Damping Factor								
		0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.95
Non-reciprocal Link Weight	0.50	1.83	1.68	1.62	1.57	1.54	1.51	1.50	1.48	1.47
	0.10	1.71	1.55	1.46	1.43	1.42	1.40	1.39	1.39	1.38
	0.06	1.68	1.54	1.47	1.42	1.41	1.41	1.40	1.40	1.39
	0.03	1.65	1.55	1.48	1.44	1.42	1.41	1.40	1.39	1.40
	0.00	1.72	1.64	1.61	1.59	1.57	1.56	1.56	1.56	1.56

6.2 Results and Discussion.

We show the tuning results for the PPR algorithm in Table 6.2. For each setting, we compute the average similarity rating (Rtng) for the top ten pages in the 120 queries issued from the first 12 diagrams. The best average rating for the PPR algorithm was 1.38, which was achieved by setting the damping factor (d) to 0.95 and using weight 0.1 for reciprocal links (setting (0.95, 0.1)). PPR performed better with $d \geq 0.45$, indicating a positive effect when increasing the influence of diagrams on the ranking process. The best average rating with $d = 0.15$, which is the value used in related work [11, 108], was 1.65 with $nRcp = 0.03$. Compared to setting $d = 0.15$, using a larger damping factor such as $d = 0.95$ (Rtng=1.38) enhanced the rating of about three pages in the top ten by one point (or one page by about two points and another page by one point).

For the non-reciprocal links weight $nRcp$, the best value was 0.1 in the best setting (0.95, 0.1). Compared to a graph configuration of only reciprocal links ($nRcp = 0$), using $nRcp = 0.1$ enhanced the average rating by 0.18. Considering all values of the

Table 6.3: Tuning results for 36 GES settings based on 120 D2C queries issued from 12 diagrams. We show and sort by average similarity rating ($Rtng$) for the top ten pages in all queries, where $1 \leq Rtng \leq 3$. The GES parameters are series (Srs), non-reciprocal-link conductance ($nRcp$), and greediness factor (F). A moderate decrease of $nRcp$ (0.7 or 0.4) had the best performance ($Rtng$ close to 1) compared to removing nonreciprocal links ($nRcp = 0$) or giving the same weight to reciprocal and nonreciprocal links ($nRcp = 1$)

Srs	nRcp	F	Rtng	Srs	nRcp	F	Rtng	Srs	nRcp	F	Rtng	Srs	nRcp	F	Rtng
2	0.7	0.9	1.23	3	0.7	0.75	1.25	3	1	0.9	1.39	2	0	0.9	1.55
3	0.4	0.75	1.23	1	0.4	0.75	1.27	2	1	0.9	1.40	3	0	0.75	1.55
3	0.7	0.9	1.24	1	0.4	0.9	1.27	3	1	0.75	1.40	3	0	0.9	1.55
3	0.4	0.9	1.24	3	0.7	1.1	1.27	1	1	0.9	1.40	2	0	0.75	1.55
1	0.7	0.9	1.24	2	0.7	1.1	1.28	1	1	0.75	1.40	1	0	0.9	1.56
1	0.7	0.75	1.24	3	0.4	1.1	1.28	2	1	0.75	1.42	1	0	0.75	1.56
2	0.7	0.75	1.24	1	0.4	1.1	1.28	1	1	1.1	1.49	1	0	1.1	1.61
2	0.4	0.75	1.25	1	0.7	1.1	1.28	2	1	1.1	1.49	2	0	1.1	1.61
2	0.4	0.9	1.25	2	0.4	1.1	1.28	3	1	1.1	1.49	3	0	1.1	1.61

damping factor, the weights 0.1, 0.06, and 0.03 for $nRcp$ had about the same average rating (each weight was the best in at least one damping factor). Also, these weights had a better average rating than weight 0.5.

We show the tuning results for the GES algorithm in Table 6.3. For the listed 36 settings, we show the average rating ($Rtng$) for the top ten pages in the 120 queries issued from the first 12 diagrams. The best setting was (2, 0.7, 0.9) (series $S2$, $nRcp = 0.7$, and $F = 0.9$) with $Rtng=1.23$. The first 18 settings (the best group) used $nRcp = 0.7$ or 0.4 with $1.23 \leq Rtng \leq 1.28$. The second group used $nRcp = 1$ (9 settings) with worse $Rtng$ (1.39 to 1.49). The group with worst performance used $nRcp = 0$ (i.e. used only reciprocal links) with $Rtng$ between 1.55 and 1.61. These differences based on $nRcp$ value indicates that for I&A-model diagrams and the GES algorithm, both reciprocal and non-reciprocal links should be used but with some preference for reciprocal links.

Note that the difference in average rating between the best setting (with $Rtng$

$= 1.23$) and the worst setting with $Rtng = 1.61$ might be noticeable by end users. Having $Rtng = 1.23$ is equivalent to having about eight similar pages and two slightly similar pages in the top ten (or nine similar pages and one dissimilar page). However, when $Rtng$ equals 1.61 that translates to having about four similar pages and six slightly similar pages (or seven similar pages and three dissimilar pages).

Using different series ($S1$, $S2$, and $S3$) did not have a large effect on the average rating. This result may be attributed to the small graph and the small number of elements in each diagram. We have seen more effect of the series parameter in the experiments with Universal-Model diagrams where we use larger graphs. Therefore, we plan to discuss this parameter in more detail when we describe our experiments regarding Universal-Model diagrams.

The performance of the different values of the greediness factor F indicate that the influence of diagrams is very important in the graph configuration we had. Energy spreading in $L2$ ($F = 0.9$ or 0.75) had only a slight positive effect on $Rtng$ compared to spreading energy only from $L1$ to $L2$ ($F = 1.1$). In addition, processing more pages in $L2$ ($F = 0.75$) did not enhance the average rating indicating that greediness reduced running time without harming the ranking. In Table 6.4, we show the number of pages the GES algorithm processed in $L2$ for the best two settings $(2, 0.7, 0.9)$ and $(3, 0.4, 0.75)$. We also show the $(2, 0.7, 0.75)$ and $(3, 0.4, 0.9)$ settings to show the effect of changing the F value. For both $(2, 0.7)$ and $(3, 0.4)$ settings, switching from 0.75 to 0.9 not only was harmless but also reduced the number of processed pages by about 40%. In general, the numbers of processed pages in $L2$ shown in Table 6.4 were quite small when considering the sizes of the TU collections used, which ranged from 140 to 14736 pages.

For the last 80 queries, we tested only the best two settings in the GES algorithm

Table 6.4: The number of processed pages (pages allowed to send energy) in L2 in four different settings including the best two settings in Table 6.3. Comparing (2, 0.7, 0.9) and (2, 0.7, 0.75) as well as (3, 0.4, 0.75) and (3, 0.4, 0.9) shows that switching from greediness factor 0.75 to 0.9 reduced the number of processed pages to about 60% without harming the average rating

GES Setting	Mean Number of Processed Pages in L2	Median Number of Processed Pages in L2	Average Rating (Rtng) for the Top Ten Pages
(2,0.7,0.9)	24.8	12.0	1.23
(2,0.7,0.75)	42.2	19.5	1.24
(3,0.4,0.9)	18.7	6.0	1.24
(3,0.4,0.75)	31.0	13.0	1.23

Table 6.5: For the best settings in the PPR and GES algorithms, we show the average similarity rating for the top ten pages in all queries issued from three data sets: the first 12 diagrams in table 6.1, the last 8 diagrams in the same table, and all 20 diagrams. Overall, the GES algorithm (2, 0.7, 0.9) is slightly better than the PPR algorithm. Using PPR instead of GES would decrease similarity (increase rating by 1) for about one and a half page in the top-ten pages.

		Average Rating For Top 10 Pages		
Algorithm	Setting	12 Diagrams	8 Diagrams	All 20 Diagrams
GES	(2, 0.7, 0.9)	1.23	1.21	1.22
GES	(3, 0.4, 0.75)	1.23	1.22	1.23
PPR	(0.85, 0.1)	1.39	1.33	1.37
PPR	(0.95, 0.1)	1.38	1.35	1.37
PPR	(0.75, 0.1)	1.39	1.34	1.37

and three of the best settings in the PPR algorithm. In Table 6.5, We show the average rating for the top ten pages using these settings in the three data sets: all queries from the first 12 diagrams, all queries from the last eight diagrams, and all queries from all of the 20 diagrams. GES was slightly better overall. The best GES setting was (2, 0.7, 0.9) with average rating 1.22 in the 20 diagrams. The best PPR setting was (0.85, 0.1) with average rating 1.37 in the 20 diagrams. We also show the average rating per diagram for these two settings in Table 6.1. GES was slightly better in 19 diagrams with Rtng difference > 0.15 in 7 diagrams.

The GES setting (2, 0.7, 0.9), which was the best overall setting in our experiment,

Table 6.6: When selecting the ‘Buffalo Mozzarella’ cheese in the cheese-by-country diagram, we show the top ten pages returned by each algorithm (GES on the left and PPR on the right). Compared to GES, PPR had greater tendency to elevate the rank of high-degree pages. Those pages are usually important concepts in the domain such as ‘Milk’ and ‘Cheese’ or pages that were part of the noise in the TU collection such as page # 9.

Rank	GES	Label	PPR	Label
1	Ricotta	1	Ricotta	1
2	Pasta filata	1	Mascarpone	1
3	Mascarpone	1	Cheese	2
4	Buffalo mozzarella	1	Blue cheese	2
5	Mozzarella	1	Brined cheese	2
6	Pecorino	1	Milk	3
7	Pecorino Romano	1	Mozzarella	1
8	Parmigiano-Reggiano	1	Parmigiano-Reggiano	1
			Geographical indications and traditional specialities in the European Union	3
9	Pecorino Siciliano	1		
10	Gorgonzola	1	Pecorino	1

had an average rating less than 1.4 in 17 diagrams (when average rating is 1.4, between 6 and 8 pages in the top ten have rating 1). In the other three diagrams, the setting average rating was about 1.5 (between 5 and 7 pages in the top ten have rating 1). These rating averages indicate that the GES algorithm was robust to this setting in the I&A-model diagrams. In addition, the overall average in all diagrams indicates that the top ten ranked pages were reasonably similar to the selected item in the diagram.

Compared to GES, PPR seemed to be more troubled by important pages in the domain, which are likely to have higher degrees than other pages. PPR was also affected by the noise in the TU collection more than the GES algorithm. For example, in Table 6.6, we show the ranking of the GES (2, 0.7, 0.9) and the PPR (0.85, 0.1) settings when selecting the element ‘Buffalo Mozzarella’ in the cheese-by-country diagram. This type of cheese originated in Italy, so we expected algorithms to return

Table 6.7: Top ten Wikipedia pages with similarity ratings when selecting the ‘Dinka’ language on the language forest (left) vs. selecting the same language on the language map (right). As with all diagram pairs, pages were quite similar to the selected element (‘Dinka’) based on the used diagram; languages in the same language family (left) vs. languages spoken in the same country (right).

Rank	Ranking using the language-map diagram	Label	Ranking using the language-forest diagram	Label
1	Dinka language	1	Nuer language	1
2	Nuer language	1	Dinka language	1
3	Arabic	1	Niger–Congo languages	3
4	Bari language	1	Maasai language	1
5	English language	2	Kuliak languages	1
6	Swahili language	1	Teso language	1
7	Zande language	1	Murle language	1
8	Murle language	1	Luo dialect	1
9	Lingua franca	3	Meroitic language	2
10	Spanish language	1	Luo languages	1

cheese types originated from the same country. We see that PPR gave a high rank for general pages in the domain such as ‘Milk’ and ‘Cheese’. These pages are likely to be referenced or have references to many cheese types. In addition, page # 9 was an irrelevant page that was collected during the TU-collection preparation. This page has a large number of links to cheese types.

In Table 6.7, we show an example of the different rankings returned from different diagrams in the same domain. The table shows the top ten pages when selecting the language ‘Dinka’ once in the language map (left) and once on the language forest (right). The two rankings were clearly influenced by the diagram used as they contained mostly different languages. The two lists share only three languages (highlighted in gray color). To provide an overview about how different the rankings are when using two diagrams, we used the best setting (2, 0.7, 0.9) to compute the following. For a query (e.g., selecting ‘Dinka’ on the forest), we retrieved the top-ten pages

and computed their average position when selecting the same element in the other diagram (e.g., selecting ‘Dinka’ on the map). Getting an average close to 5.5 means that both diagrams had similar pages in the top ten. However, the high numbers shown in the rightmost column in Table 6.1 (average of averages for all queries) indicate that a pair of I&A diagrams showing different aspects of a domain can produce quite different rankings.

6.3 Discussion Summary

In this section, we summarize our findings in the previous section and return to our research questions listed in the beginning of the chapter.

RQ1: Do I&A-model diagrams positively bias the ranking process to retrieve pages related to the selected element in a D2C query? Are the content and structure in I&A-model diagrams useful in this regard?

On average, considering the best overall setting (2, 0.7, 0.9), the top ten pages were reasonably similar to the selected item in the diagram. The success of the GES algorithm, which we designed to maximize the influence of diagrams, suggests that the content and the structure in the diagrams were useful. Our analysis of the greediness factor F also shows that sending energy only from pages connected to the diagrams performed close to the best result (setting (3, 0.7, 1.1) had average rating 1.27). Also, additional energy spreading in L2 ($F = 0.75$) was not useful. Furthermore, in all ten domains, using a pair of diagrams, showing different aspects, returned different rankings that reflected the diagrams’ different aspects. Note that any other ranking approaches, such as the Wikipedia keyword search, cannot be useful in both aspects. These ranking approaches can at most match the ranking returned from one diagram, but then must be different from the ranking returned from the other. They can also combine the two aspects of the domain or completely disregard them to generate a

third different ranking.

In the PPR algorithm, the utility of diagrams is indicated by the positive effect of a high damping factor values on the average rating for the top ten pages (Figure 6.2). The more the random surfer stayed close to the diagrams, the better the returned top ten pages were. Similar applications that used the PPR algorithm on the Wikipedia graph considered the value 0.15 for the damping factor to be optimal. However, MDN D2C queries benefited from high damping factor values (0.75 to 0.95), indicating the usefulness of the context that diagrams provide.

RQ2: How much impact should the content and structure of I&A-model diagrams have on the ranking process? The results of the tuning session for both the PPR and GES algorithms indicate that the impact of diagrams on the ranking process should be maximized. The best performance of PPR was with damping factor close to 0.85. In GES, spreading less energy in L2 when setting F to 0.9 (compared to $F = 0.75$) saved time and did not harm performance. Note that setting F to 0.9 rely on L1 pages (i.e., attribute pages) and a relatively few related pages in L2 (see Table 6.4).

RQ3: What are the best weights for reciprocal and nonreciprocal links? The results we had from the tuning session of both the PPR and GES algorithms indicate that reciprocal links are important, which agrees with the findings of Agirre et al. [11]. However, unlike that work, we found that we should also use reduced-weight nonreciprocal links. In both the PPR and GES algorithms, the difference in the average rating between the best setting overall and the best setting that used only reciprocal links was about 0.2 (about two pages in the top ten would have a worse rating by one point if only reciprocal links were used). Regarding the optimal weight for non-reciprocal links, GES performed about the same with weights 0.7 and

0.4. The PPR algorithm performed better with the weight 0.1 ($R_{tng} = 1.38$) than the weight 0.5 ($R_{tng} = 1.47$).

RQ4: What are the best settings for the PPR and GES algorithms in D2C queries issued from I&A-model diagrams? Which algorithm is better? Was the better algorithm robust for all I&A-model diagrams? According to Table 6.5, the best setting for the GES algorithm was (2, 0.7, 0.9) while the best setting for the PPR algorithm was (0.85, 0.1) ((0.75, 0.1) and (0.95, 0.1) had similar performance). The GES algorithm was slightly better than the PPR algorithm with 0.15 difference in the overall average rating for the top-ten pages. This difference is equivalent to enhancing the ranking of 1.5 pages in the top ten by one point. Also, GES outperformed PPR in 19 diagrams with average rating difference greater than 0.15 in 7 diagrams. The average rating per diagram shown in Table 6.1 indicates that GES was robust in most diagrams with 14 diagrams having average rating ≤ 1.3 (at least seven pages with rating one in the top ten) and 3 diagrams having average rating between 1.37 and 1.38.

RQ5: In I&A-model diagrams, given that we can show two different aspects of a domain by using two different diagrams, how different are the rankings we get from these diagrams? The results we showed in Table 6.1 (in the rightmost column) indicate that using two I&A-model diagrams can provide users with different rankings influenced by the aspect of the diagram used in the query.

We remind the reader that our approach only relied on using attributes connected to a selected item in a D2C query (recall that we used partial graphs, which we explained in Section 3.2.1). Given the positive results we had using this approach, we argue that future studies and implementations that include I&A-model diagrams should focus on increasing the influence of attributes on ranking. The approach may

also be studied with other graph structures where major influence is given to certain diagram elements that represent a desired aspect or subdomain.

7 Evaluation with Universal-Model Diagrams

In this chapter, we describe our second investigation regarding Extended MDN (EMDN). In this investigation, we evaluate the ranking returned by the MDN (EMDN) D2C-query type in Universal-Model diagrams using both the PPR and GES algorithms.

The research questions for the second investigation are quite similar to the ones listed for the first investigation. The main difference is that we use Universal-Model diagrams. Also, we evaluate more graph configurations to test the effect of including Wikipedia categories and article templates. The methods in this investigation are also different compared to the methods used in the first investigation. We used an automatic evaluation that allowed us to test a large number of settings in both GES and PPR.

We designed the second investigation to answer the following research questions:

- RQ1) Do Universal-Model diagrams positively bias the ranking process to retrieve pages related to the selected element in a D2C query? Assuming a user is interested in a diagram, a desirable feature is to retrieve Wikipedia pages related to the diagram and the selected element in the D2C query. We wanted to know whether the content and structure of Universal-Model diagrams are useful in this regard.
- RQ2) If the answer to RQ1 is yes, how much effect should the content and structure of the Universal-Model diagrams have on the ranking process? We

study this research question using these indicators:

- What is the best damping factor setting for the PPR algorithm? A high damping factor increases the effect of diagrams’ content and structure on the ranking process, while a low damping factor value increases the impact of the Wikipedia’s content and structure. Related work [11,108] that used the Wikipedia collection set the damping factor value to the popular value 0.15. We wanted to know whether this value is also optimal for MDN D2C queries.
- How does the GES algorithm perform in D2C queries, and how greedy should the algorithm be in Wikipedia? We designed the GES algorithm to maximize the diagrams’ influence on the ranking process while minimizing the Wikipedia influence. Also, optimal levels of greediness in each stage of energy spreading may indicate whether we should send more energy from pages connected to diagram elements or from other Wikipedia pages to achieve the best performance.
- RQ3) What are the best weights for reciprocal and nonreciprocal links? Agirre et al. [11] achieved their best results in two applications using the PPR algorithm on the Wikipedia graph when using only reciprocal links. We wanted to know whether this graph configuration is also optimal for MDN D2C queries. We also tested a graph configuration where we include nonreciprocal links.
- RQ4) Should we include hyperlinks that appear in article templates? In the first investigation, we noticed that an article template may shift the topic of the article or emphasize a specific aspect that is different from the one emphasized by the used diagram. Therefore, we discarded hyperlinks in article templates (see Section 6.1). In this investigation, we test the effect of including and

excluding template hyperlinks on both the PPR and GES algorithms. The automatic evaluation method (described in the next section) allowed us to test such settings that we discarded in the first investigation to reduce the amount of data that needed human labeling.

- RQ5) Should we include Wikipedia categories? Agirre et al. [11] achieved their best performance for the PPR algorithm with a graph that discarded Wikipedia categories (included only articles). We wanted to know how the inclusion and exclusion of categories affects both the PPR and GES algorithms.
- RQ6) What are the best settings for the PPR and GES algorithms in D2C queries issued from Universal-Model diagrams? Which algorithm is better? Was the better algorithm robust for all graph classes in Universal-Model diagrams?

Similar to Chapter 6 we do not focus on the running time of the PPR and GES algorithms. Our plan is to process D2C queries and store results offline. Then, we plan to use the stored data to process D2C and C2D queries online. Therefore, we assume that the speed of online queries is not an important factor.

7.1 Methods

For our experiments in this chapter we used 20 diagrams shown in Table 7.1. We represented the diagrams using the four graph classes in the Universal Model. The sizes of the diagrams (number of elements) ranged from 11 to 179. We created all diagram internal connections as well as the connections between diagram elements and Wikipedia pages. For the most part, a diagram element is connected to a single Wikipedia article (where both the element and the article cover the same concept). Following the same approach we used in the previous chapter, we used a set of 120 D2C queries issued from the first 12 diagrams (10 from each diagram) for tuning purposes. We used the full set of 200 D2C queries for the complete evaluation in

Table 7.1: The 20 tables we used for Universal-Model-Diagrams Investigation.

#	Diagram	Size	Graph Class
1	JavaScript Libraries	41	Tree
2	Portland Attractions Map	41	Tree
3	Attack Graph	29	Generic
4	Coffee	32	Tree
5	Portland Bridges	12	Path
6	Nursing Career Flowchart	11	Generic
7	Programming Languages	38	Generic
8	Data Science Tools	163	Tree
9	Culinary Herbs	16	Tree
10	WW2 Timeline	25	Path
11	Data Platforms	179	I&A
12	Chocolate Ingredients	89	I&A
13	South America Map	13	Generic
14	Body Systems	36	Tree
15	Geological Eras	22	Generic
16	Rock Types	16	I&A
17	Space Discovery Timeline	13	Path
18	TV Networks	58	Generic
19	Periodic Table	156	I&A
20	Cheese Types	59	I&A

Section 7.4. For each D2C query we only consider and store the top 5000 Wikipedia pages from the ranked list returned by the tested algorithm.

In contrast to the manual-labeling method we used with I&A-Model diagrams, we used an automatic evaluation method with Universal-Model diagrams to measure the performance of the tested algorithms and settings. For each D2C query, we cut the connection between the selected element in the diagram and its related page (or one of its related pages if the element is connected to multiple pages). We use the rank of the disconnected page (or target page) as an evaluation metric. Based on the diagram content, we assume the target page is highly relevant. Therefore, ranking this page at the top of the ranked list is desirable. We characterize high performance for an algorithm or a setting as the ability to rank a large number of the target pages

(over all issued D2C queries) in the top k . Note that an algorithm setting may not be able to rank all target pages in the top 5000.

For most of the figures in this chapter we compare settings (and algorithms) using the following process. For each setting, we sort queries by the rank of the target page in a non-decreasing order. Then, we compute the mean and median target-page rank in the best k queries. For the tuning session, we show results in the range $60 \leq k \leq 120$ (i.e., 50% to 100% of the queries). For example, Figure 7.6 shows the mean and median target-page rank of multiple GES settings. As different settings may perform better in different queries, each setting may have different queries in the best k . In most of our experiments, settings rank only between 95% to 98% in the top 5000 (i.e., 115 and 118 out of the 120 target pages). In addition, the worst 5 to 10% of queries tend to have very high ranks compared to the majority of the queries. For example, the setting $(1, S5, TF, 0.4)$, which is the best setting in Figure 7.6, has a mean rank equal to 42 for the best 80% of the queries. However, the mean rank exceeds 150 when considering 95% of the queries or more. Therefore, we consider the worst 5% to 10% of the queries for a specific setting to be outliers that do not represent the performance of a setting or an algorithm.

7.1.1 Preparation of the Undirected Weighted Graphs

Before going into the details of this chapter, we explain how we built the graphs we used in our experiments. As we described in Sections 3.3.1 and 3.3.2, we had two graph versions for Universal-Model diagrams: a directed unweighted graph and an undirected weighted graph. Recall that in the latter graph, we use different weights for connections between pages (to test different weights for non-reciprocal links) and we use full weights for connections in the diagram and between diagram elements and Wikipedia pages. We show example graphs in Figure 7.1. We repeat the same figure

from Section 3.3.1. We use the undirected weighted graphs in our main experiments. However, before the main experiments we used directed unweighted graphs to crawl Wikipedia and collect pages related to each diagram. We explain the crawling process in this section.

Figure 7.1: Part A: The modeling of a universal-model diagram and Wikipedia pages using a directed unweighted graph. Part B: The modeling of a universal-model diagram (the set D) and a subset of Wikipedia pages (the set R) using an undirected weighted graph. Note that we include only edges where both nodes connected by the edge are in $D \cup R$ (e.g., edge connecting pages X and Y is not included in the graph since Y is not in the set $D \cup R$).

For each diagram, we planned to use a separate undirected weighted graph that includes the set D of diagram elements and the set R , which includes Wikipedia pages that represent the domain of the diagram. We decided to include 100,000 pages in R . From early experiments, this many pages seemed sufficient to cover the domain of a given diagram and also run a large number of experiments in a reasonable amount of time (to test a large number of settings for each algorithm).

Our way to collect pages related to a diagram was based on crawling Wikipedia starting from the diagram using the directed unweighted graph. The crawling was done using both the PPR-MC (described in Section 4.4) and GES algorithms. Using only page outgoing links in Wikipedia (the graph is directed unweighted) we crawled Wikipedia on demand. We fetched a page if it was requested by one of the algorithms

(e.g., when the random surfer in PPR selects a page to jump to). Our plan was to use the top N ranked pages returned by both algorithms to build the set R . However, to enhance our chances of finding all pages related to a diagram, we needed to tune the PPR-MC and GES algorithms.

We tuned the algorithms using 120 queries issued from the first 12 diagrams in Table 7.1. We used PPR-MC with 20,000 steps per D2C query for the random surfer. We tested damping factor (d) values 0.1, 0.15, and 0.25. We discarded higher damping factor values as they decreased the number of pages returned by the algorithm (only a few pages close to the diagram get most of the visits). For GES, we tested the series $S1$ to $S5$ and array distributions $Dist2$, $Dist3$, and $Dist4$ (described in Section 5.2.2). $Dist1$ and $S6$ were introduced later in the experiment described in Section 7.1.2). We ran GES for 4 stages (until the ‘3–3’ stage to reach pages in L3). Propagating energy until L3 was sufficient as most target pages were in L2. We discuss this observation in more detail later in this chapter. We used greediness factor 0 for the ‘1–2’ stage (i.e., propagate energy from all pages connected to the diagram), and used 0.9 for the rest of the stages. We used these greediness factors as they achieved the best performance in our previous investigation regarding I&A-Model diagrams and they also showed high performance in our early experiments with Universal-Model diagrams (i.e., they ranked target pages higher than other settings).

We show the performance of the best two settings for each algorithm in Figure 7.2. The best setting for PPR was PPR(0.25) with recall 80% (retrieving only 96 target pages out of 120). The recall for GES(2, $S2$) was slightly better at 85%. At $k = 96$, the mean rank for PPR-MC(0.25) was 229 while it was 158 for GES(2, $S2$). Note that this tuning was not done to evaluate algorithms or settings. The main goal was to make sure crawling and ranking is effective enough to retrieve pages related

Figure 7.2: Best two settings of PPR and GES on the directed unweighted graph using a data set of 120 D2C queries. We show the mean rank of the target page at different k values.

to a given diagram. The results in Figure 7.2 indicate that if we include the top few thousand pages ranked by GES and PPR in the set R , we are likely to include most of the pages related to the given diagram.

To build the set R for a given diagram, we issued D2C queries from all diagram elements using both PPR(0.25) and GES(2,S2). We collected the returned ranked lists and computed the union of the top 3000 pages. For PPR(0.25), the top 3000 pages in a D2C query mostly included pages that are visited frequently by the random surfer (visited at least twice). Those pages are more likely to be related to the diagram compared to pages visited only once. We used the same number with GES(2,S2) in order to avoid the influence of a particular algorithm on the building of the set R . The number of pages added to the set R from this step (the result of the union) ranged

between 24,000 and 56,000 Wikipedia pages. For example, in a diagram of size 50, we could collect as many as 300,000 pages (i.e., 2 algorithms * 50 D2C queries * 3000 pages per query). However, many pages are returned in multiple D2C queries and therefore the actual set we get is much smaller.

To reach the size of 100,000 for the set R , we harvested pages from Wikipedia categories related to a given diagram. We collected categories referenced by L1 pages. We sorted those categories by the number of references they had from L1 pages in a non-increasing order and inserted them in a queue. Each time we dequeue a category from the queue, we add its pages to the set R and also enqueue its subcategories. We continued this process until the set R reached 100,000 pages. We discarded categories whose size is larger than 2000, as those categories are too generic and are likely not representing the domain of the diagram (e.g., the category of living people).

Given the set R , we built an undirected weighted graph for each diagram. We built multiple versions of this graph that differ in terms of: (1) inclusion or exclusion of Wikipedia categories (include article and categories or articles only); (2) inclusion or exclusion of article-template hyperlinks; (3) weights of reciprocal and non-reciprocal links. Both the PPR and the GES were tested using all of these graph versions. Note that the term *category* we use in this chapter includes List_of pages as well (the same approach used in Agirre et al. [11]). Also, graphs that exclude categories are about 15% smaller (about 85,000 pages) compared to the ones that include both articles and categories.

7.1.2 Tuning the PPR and GES Algorithms

In this section, we explain our tuning for the PPR and GES algorithms using a set of 120 D2C queries issued from 12 diagrams. We briefly describe the parameters and the tested values for each algorithm. We discuss those parameters in more detail

Table 7.2: Tested parameters for the PPR algorithm. We show best values in bold font.

Parameter	Tested Values
The damping factor (d).	0.15, 0.25, 0.35, 0.45, 0.55, 0.65 , 0.75, and 0.85
Inclusion of categories and article templates	Only articles with no templates. (FF); only articles with templates (FT); categories and articles with no templates (TF); categories and articles with templates (TT).
Weight of non-reciprocal links.	1, 0.5 , 0.1, and 0.01.

when presenting the tuning results in Sections 7.2 and 7.3.

For PPR we tested the parameters and the values shown in Table 7.2. For each query in the set of 120 D2C queries, we ran PPR for 128 settings, which comprise all the combinations of parameter values in Table 7.2. We denote the different settings of the PPR algorithm using those parameter values. For example, the best setting in our experiments was (0.65, TT, 0.5), which refers to assigning 0.65 to the damping factor and using the TT graph. The last parameter is the weight of non-reciprocal links (while assigning one to reciprocal-link weights).

We used the PPR algorithm described in Sections 4.2 and 4.3, where the base set (teleport vector) contains the selected element in a D2C query. We computed the Equation 4.2 for 40 iterations to reach a convergence of the PageRank vector with L_1 error < 0.0001 . The need for more iterations (we used 30 iterations in the I&A-Model experiment) is probably due to the larger graphs we used in this experiment.

For GES we tested the parameters and the values shown in Table 7.3. We denote the different settings of the GES algorithm using those parameters. For example, the best setting in our experiments was (1, S6, TF, 0.4), which refers to using Dist1 (1 for short) in propagating energy in the diagram and S6 (series) in the SAS computation. TF refers to the undirected weighted graph configuration that includes categories

Table 7.3: Tested parameters for the GES algorithm. We show best values in bold font.

Parameter	Tested Values
Energy distribution in the diagram.	Dist1 , Dist2, Dist3, and Dist4.
The series used in SAS computation	S1, S2, S3, S4, S5, and S6 .
Inclusion of categories and article-template hyperlinks	Only articles with no template hyperlinks (FF); only articles with template hyperlinks (FT); categories and articles with no template hyperlinks (TF); categories and articles with template hyperlinks (TT).
Weight of non-reciprocal links.	0.1, 0.4 , and 0.7.
Greediness factor for level ‘1–2’ (F_1)	0, and 0.5
Greediness factor for levels ‘2–2’ and ‘2–3’ (F_2)	0.75, 0.9 , and 1.1
Greediness factor for level ‘3–3’ (F_3)	0.65, 0.85 , and 1.1

(and articles) as well as article-template hyperlinks. The value 0.4 is the weight of non-reciprocal links (while assigning 1 to the weight of reciprocal links). The best values for the greediness factors F_1 , F_2 , and F_3 were 0.5, 0.9, and 0.85, respectively. We use the value of F_2 when propagating energy among L2 pages (the ‘2–2’ stage) and when propagating energy from L2 to L3 (the ‘2–3’ stage). For brevity, we do not include greediness factors when referring to GES settings.

We ran GES for four stages (until the ‘3–3’ stage). The number of stages was sufficient as 95% of the target pages were in L2 while the other 5% were in L3. We started with 360 settings, which are the combinations of the four distribution arrays, five series (S_6 was not included), three non-reciprocal-link weights, and four graph configurations. We added some settings in later experiments as we describe in the next section. When computing a page energy (score), we limited the SAS computation to the the highest 20 received energies ($n = 20$ in Algorithm 5). Early

experiments showed that the optimal number is between 20 and 25. For greediness factors, we used the values $(0, 0.9, 0.85)$ for (F_1, F_2, F_3) , respectively. We used those values based on our experience in the I&A-Model experiment, but we did tune these parameters as we describe in the next section.

We further discuss PPR and GES parameters while presenting the tuning results in the next two sections. In Section 7.4, we present and compare the performance of both algorithms using the set of 200 D2C queries.

7.2 PPR Tuning Results

Graph Configuration. We started the PPR tuning by examining the parameters related to graph construction. We examined the effect of including categories and article-template hyperlinks as well as the effect of using different weights for non-reciprocal links. The inclusion of categories and the special treatment of reciprocal links was studied in related work [11] and found to be influential in similar applications using the graph of Wikipedia. We also explained in the previous chapter that article templates may reverse the influence of a diagram on the ranking process. We wanted to test the effect of all of those parameters on the performance of PPR.

We compare the PPR performance on the four graph configurations listed in Table 7.2. Figure 7.3 shows the best setting in each graph. For both mean and median target-page rank, lower values are better (i.e., indicating that a setting is capable of ranking highly related pages in the top n). In Figure 7.3, we underline the parameter (inside the setting) that we are currently investigating, which is the graph configuration (TT, FF, TT, and FT) in this case. We use the same method for the rest of the figures in the chapter. The best graphs for PPR were TT, FT, and FF. We consider the TT graph as the best graph for the PPR algorithm, given that it includes articles and categories. Using the TT graph allows MDN to include all types

of those pages when ranking Wikipedia content in D2C queries (some categories and List_of pages can be related to the selected diagram element). For the rest of the parameters discussed in this section we consider only the TT graph.

We notice that including article templates had a positive effect on PPR: FT and TT graphs had a better mean rank than the FF and TF graphs at most k values. In addition, TT and FT had a better overall median @ $k = 117$ (the difference was 3 compared to the FF graph and 10 compared to the TF graph). We also notice that including categories alone without article templates had a negative effect on the PPR algorithm. This result aligns with the finding of Agirre et. al [11] that including categories hurts PPR performance. However, we do not know whether the graph used in their study included article-template hyperlinks or not.

Figure 7.3: For the best setting in each of the four graph configurations listed in Table 7.2, we show the mean and median of target-page ranks at different k values. In the mean’s top left figure, we show a zoomed-in view for k values up to 107. The best setting (0.65, TT, 0.5) performs close to (0.75, FT, 0.5) and (0.45, FF, 0.1), but (0.65, TT, 0.5) is considered better given that it works with a larger graph that includes both articles and categories.

The second parameter we examined is the weight of non-reciprocal links. In our

experiments, we assigned the weight 1 to reciprocal links and tested the weights 1, 0.5, 0.1, and 0.01 for non-reciprocal links. In Figure 7.4, we show mean and median rank of target pages at different k values of the best setting for each non-reciprocal link weight. While the median was close for all settings, the mean shows an advantage of using weight 0.5 for non-reciprocal links. At $k = 108$, the mean for (0.65, TT, 0.5) was 109 while it was 119 for (0.75, TT, 1) and 136 for (0.75, TT, 0.1).

Figure 7.4: Mean and median rank of target pages at different k values for the best setting for each non-reciprocal link weight listed in Table 7.2. PPR achieves the best performance with keeping low-weight non-reciprocal links as the setting (0.65, TT, 0.5) achieves best mean rank @ $k = 108$.

Damping factor. The third parameter we tested is the damping factor. Overall, higher damping factors had better performance in MDN D2C queries, which matches our finding in the I&A-Model experiment. In Figure 7.5, we compare the best setting (0.65, TT, 0.5) with all the settings that used damping factor 0.15. The value 0.15 is popular in related work, including ones that use the graph of Wikipedia [11, 108].

Figure 7.5: Mean and median rank of target pages at different k values for the best PPR setting $(0.65, TT, 0.5)$ and all settings that used 0.15 as a damping factor (similar to related work).

Among the settings that used the value 0.15 for d , the best one was $(0.15, TT, 0.1)$. However, the superiority of $(0.65, TT, 0.5)$ is obvious in both the mean and median rank metrics.

We consider the $(0.65, TT, 0.5)$ to be the best PPR setting. We use this setting in our detailed comparison of PPR and GES in Section 7.4. Before doing that, we show our tuning for the GES algorithm in the next section.

7.3 GES Tuning Results

Graph Configuration. In Figure 7.6, we show the performance of GES in the four graph configurations we listed in Table 7.3. We only show the best setting in each graph (the best setting out of 90 settings tested in each of the four graphs). The results in Figure 7.6 show the negative influence of article-template hyperlinks on GES. As we discussed in the previous chapter, article templates may shift the topic of an article to a broader topic or emphasize a certain aspect leading to increasing

the score of pages unrelated to the diagram. Categories on the other hand seem to help GES. A category that is relevant to multiple diagram elements is likely to accumulate a large amount of energy, which boosts the rank of other possibly related pages included in the category. For the rest of the tested parameters in GES, we only focus on the TF graph, given that GES had low performance in the TT and FT graphs. Also, we discard the FF graph since it is smaller (did not include categories). Note that some categories and List_of pages might be related and including them in the ranked list might be desirable for the end user.

Figure 7.6: Testing GES with four graph configurations. The algorithm performs the best with the TF graph which includes Wikipedia articles and categories and excludes hyperlinks found in article templates.

For non-reciprocal link weights, Figure 7.7 shows the performance of the best settings for weights 0.1, 0.3, and 0.7 in the TF graph (while giving weight one to reciprocal links). We see that the best GES performance was achieved with the weight 0.4 (giving non-reciprocal links about half the weight given to reciprocal links). The worst performance was observed with weight 0.1. The weight 0.4 did perform slightly better than 0.7 (@k = 108 the mean ranks were 93 and 103 for weights 0.4 and 0.7, respectively). We also did fine tuning using the weights 0.3 and 0.5, which showed

that GES reaches peak performance with weight 0.4.

Figure 7.7: Testing GES with three weights for non-reciprocal links in the TF graph. GES achieved best performance by giving non-reciprocal links about half the weight (0.4) of reciprocal links.

SAS Series and Energy Distribution in Diagrams. Using the TF graph, we tested the performance of the different SAS series and diagram-energy distribution arrays. SAS dictates the behaviour of computing a page score (controls the trade-off between rewarding high received energies and rewarding a high number of received energies). Similarly, distribution arrays used to propagate energy in diagrams dictate which diagram elements have the largest influence on the ranking process (only elements close to the selected element or most diagram elements). SAS Series and distribution arrays are explained in more detail in Section 5.2.2.

In Figure 7.8, we show the performance of the best setting for each of the six tested series. Before including $S6$, the best series we had were $S5$ and $S2$ with mean ranks 93 and 98 @ $k = 108$, respectively. The fine tuning of these two series yielded $S6$, which is a variation of $S5$. In the TF graph, GES performs the best when combining $S6$ and Dist1 (i.e., the setting (1, $S6$, TF, 0.4)) with a mean rank equal to 83 @ $k = 108$ and overall median equal to 17 @ $k = 116$.

Figure 7.8: Among the tested six series, $S6$ was the best choice with a mean rank equal to 83 @ $k = 108$ and overall median equal to 17 @ $k = 116$

Looking at the low performance of $S4$, which is equivalent to averaging the top 20 received energies, we observe the importance of using SAS where we use a declining series to assign more weight to higher received energies. Comparing the best two series $S2$ and $S6$, we note that the sum of the terms in $S6$ is greater than one (the sum of the terms of $S5$ is also greater than one). This property of $S6$ and $S5$ helped elevate the rank of L2 and L3 pages (including the target page) as we discuss in more detail in Section 7.4.

The fourth parameter we look at for GES is the distribution array used to propagate energy from a selected element in a D2C-query to other elements in the diagram. We show the best setting for each distribution array in Figure 7.9. Dist1, which favors elements close to the D2C selected element, achieves the best performance. The dominance of Dist1 can be attributed to the fact that most related elements (to a D2C selected element) are one or two edges away (e.g., the parent and siblings in tree diagrams and attributes and related items in I&A diagrams). Dist1 accommodates

this pattern by assigning little or no energy to elements far from the selected element. For example, Dist1 allows only elements with distance 1 or 2 to get an energy greater than or equal 0.25 (compared to Dist3 where any element with distance less than 11 can get that energy).

Figure 7.9: The performance of four energy distribution arrays (used to propagate energy among diagram elements). Dist1 (1 for short) achieves the best performance by giving more weight to diagram elements close to the selected element (see the sharp decline of Dist1 in Figure 5.1)

Greediness. Finally we look at the performance of greediness in GES. A greediness factor F for a stage t dictates which pages are allowed to propagate energy in t (i.e., which pages GES has to process when propagating energy). The higher the value of F the fewer nodes are allowed to propagate energy (i.e., making the algorithm more greedy). When setting F to 1.1 no nodes are allowed to send energy in a stage. When tuning the previous parameters we used the greediness factors (0, 0.9, 0.85) for (F_1, F_2, F_3) , respectively. In Figure 7.10, we show the performance of four greediness-factor combinations that we tested with the best setting so far (1, S6, TF, 0.4). In Table 7.4, we show the mean and median number of pages allowed to propagate energy in each stage for these greediness-factor combinations.

The best greediness factors (F_1, F_2, F_3) were (0.5, 0.9, 0.85) for the stages ('1–

Figure 7.10: Comparing different greediness-factor combinations for (F_1, F_2, F_3) . High greediness in the ‘2–2’ and ‘2–3’ stages ($F_2 = 0.9$) is better than both lower greediness ($F_2 = 0.75$) and extremely high greediness ($F_2 = 1.1$), which is equivalent to canceling energy propagation in L2 and L3.

Table 7.4: A comparison between the best greediness setting (row 1) and two alternatives. For each setting, we show the (mean, median) number of nodes allowed to propagate energy in each stage.

Greediness	‘1–2’	‘2–2’	‘2–3’	‘3–3’
(0.5, 0.9, 0.85)	(16, 9)	(2, 2)	(2, 1)	(1, 1)
(0.00, 0.9, 0.85)	(57, 39)	(3, 2)	(2, 1)	(1, 1)
(0.5, 0.75, 0.65)	(16, 9)	(6, 4)	(5, 3)	(63, 13)

2’, ‘2–2’ & ‘2–3’, ‘3–3’). Overall, the total number of pages that need to propagate energy is quite small, which reduces execution time without harming the ranking performance. In the first row in Table 7.4, the total number of pages (total means) that propagate energy in the best setting in all stages is 21 pages (total medians is only 13). These small totals indicate that for MDN D2C queries, GES can find related pages by only considering Wikipedia pages with strong connections to the selected element in the diagram. As 95% of the target pages were in L2, we see that GES relied mostly on propagating energy from L1 pages (16 L1 pages and only two L2 pages). These numbers may be attributed to the rich context that diagrams provided

(i.e., the highly relevant pages were already connected to the diagram).

Comparing (0.5, 0.9, 0.85) and (0, 0.9, 0.85), the values 0.5 and 0 for F_1 had similar performance. However, we consider 0.5 to be better as it requires processing fewer nodes in the ‘1–2’ stage (the value 0 indicates processing all L1 pages). Looking at the first two rows in Table 7.4, we see that setting F_1 to 0.5 reduced the number of processed pages by 72%.

When we switched (0.5, 0.9, 0.85) to (0.5, 0.75, 0.65) we observed that the number of processed pages in the ‘2–2’ stage increased slightly (about 4 pages). However, the performance dropped, especially when looking at the mean rank (96 and 131 @ $k = 108$ for the two settings, respectively). On the other hand, when we tested the factors (0.5, 1.1, 1.1), which process no nodes in the ‘2–2’, ‘2–3’, and ‘3–3’ stages, GES recorded the worst performance with mean rank equal to 172 @ $k = 108$ and median rank equal to 51 @ $k = 114$.

At the end of the GES tuning, we considered the setting (1, S_6 , TF, 0.4) to be the best GES setting with greediness factors (0.5, 0.9, 0.85). We use this setting in the final evaluation in the next section where we compare PPR and GES.

7.4 Comparison of GES and PPR Algorithms

In this section, we compare the performance of GES(1, S_6 , TF, 0.4) and PPR(0.65, TT, 0.5), which are the best settings in GES and PPR, respectively, based on our tuning in the previous two sections. We use a data set of 200 D2C queries, which include the 120 queries used for tuning. The 200-query set includes ten queries from each of the diagrams in Table 7.1. We also compare the two algorithms using the same metric we used for tuning (mean and median rank of target pages) and also based on other aspects of their D2C ranking.

We start by showing the mean and median rank of target pages at different k

Figure 7.11: Mean and median rank of target pages at different k values for the best settings of GES and PPR.

values for the two algorithms in Figure 7.11. GES outperforms PPR with mean rank equal to 58 in GES and 81 in PPR @ $k = 180$. Also, the median @ $k = 194$ was 13.5 in GES and 29 in PPR. However, with a detailed analysis we observed that the treatment (the ranking) of L1 pages had an influential role in these results. PPR tended to rank more L1 pages in the top N compared to GES. Therefore, the rank of the target page (which is in L2 or L3) was harmed in PPR.

To illustrate the matter in more detail, in Table 7.5, we include GES and PPR ranked lists for an example D2C query when ‘Cappuccino’ was selected in the coffee diagram. This diagram (an excerpt is shown in Figure 3.12) has groups of related concepts such as coffee drinks, coffee-producing countries, and coffee preparation machines. Given that the selected element is a coffee drink, we assumed that returned pages should be coffee drinks. Therefore, we highlight irrelevant pages with shaded background in Table 7.5. We also use bold and italic font for pages that are in L1. In PPR, the top 22 positions were L1 pages including 11 pages that were not related to coffee drinks, which contributed to degrading the rank of the target page (at 24). On the other hand, GES focused only on coffee drinks, including pages from L1 (pages connected to the group of coffee drinks in the diagram) and pages from other levels.

Table 7.5: Ranked lists returned from PPR and GES when ‘Cappuccino’ is selected in the coffee diagram. L1 pages (pages connected to the diagram) are in bold italic font. Pages less relevant to the selected element are in shaded background.

Rank	GES	PPR
1	Coffee drinks	<i>List of coffee drinks</i>
2	<i>List of coffee drinks</i>	<i>Espresso</i>
3	*** Cappuccino ***	<i>Caffè mocha</i>
4	Cortado	<i>Latte</i>
5	Lungo	<i>Caffè macchiato</i>
6	Long black	<i>Irish coffee</i>
7	Ristretto	<i>Caffè Americano</i>
8	Latte macchiato	<i>Café au lait</i>
9	Caffè crema	<i>Espresso con panna</i>
10	<i>Espresso</i>	<i>Coffee</i>
11	<i>Latte</i>	<i>Coffee production</i>
12	<i>Irish coffee</i>	<i>Coffee preparation</i>
13	<i>Caffè macchiato</i>	<i>List of countries by coffee production</i>
14	<i>Coffee</i>	<i>Coffea arabica</i>
15	<i>Café au lait</i>	<i>Coffea</i>
16	<i>Caffè mocha</i>	<i>Decaffeination</i>
17	<i>Caffè Americano</i>	<i>Coffee roasting</i>
18	<i>Espresso con panna</i>	<i>Espresso machine</i>
19	Flat white	<i>Brewed coffee</i>
20	Galão	<i>Moka pot</i>
21	Cuban espresso	<i>Coffee percolator</i>
22	Iced coffee	<i>French press</i>
23	Turkish coffee	Wiener Melange
24	Mazagran (drink)	*** Cappuccino ***
25	Café con leche	Latte macchiato

To get an overall picture of how GES and PPR rank L1 pages, we studied where each algorithm places L1 pages. For each query’s top 100 positions, we split the list into 10-position intervals and computed the percentage of L1 pages in those intervals. In Table 7.6, we show the average percentage of L1 pages in those intervals across all 200 D2C queries. PPR clearly places most L1 pages in the top N compared to GES.

L1 pages constitute about 75% of the top 20 in PPR ranked lists while in GES they are only 25%.

Table 7.6: In a D2C-query’s top 100 ranked pages, we compute the percentage of pages that are L1 pages in each 10-position interval. We show the average percentages of all 200 D2C queries. PPR has more tendency to rank L1 pages in the top compared to GES.

Rank Range	Average Percentage of L1 pages (GES)	Average Percentage of L1 pages (PPR)
1 to 10	29%	92%
11 to 20	20%	57%
21 to 30	15%	33%
31 to 40	8%	22%
41 to 50	6%	17%
51 to 60	5%	13%
61 to 70	7%	11%
71 to 80	8%	10%
81 to 90	9%	9%
91 to 100	7%	7%

The high damping factor used in PPR is a possible reason of elevating the rank of L1 pages. While a high damping factor enhances performance by keeping the random surfer close to the diagram, it may also increase the surfer’s visits to L1 pages. The random surfer did not seem to focus on diagram elements (and L1 pages) that are close to the selected element (and avoid the less related ones). On the other hand, the GES setting (1, S_6 , TF, 0.4) used the series S_6 (whose sum is greater than one) and Dist1 (which favors elements close to the selected element). The combination of Dist1 and S_6 helps L2 pages compete with L1 pages. While Dist1 degrades the energy of diagram elements that are distant from the selected element, S_6 boosts the energy of L2 pages. Furthermore, the SAS computation and greediness in GES generally favor high-energy pages, which are likely to be close to the selected element in the diagram.

For a better understanding of performance, we removed L1 pages from the ranked lists. In Figure 7.12, we show the mean and median rank of the target pages after

this process. We also show a distribution of the target-page ranks in Figure 7.13. With the removal of L1 pages, we see that GES and PPR have similar performance. In Figure 7.13, both algorithms rank about 50% of the target pages in the top ten (53% in PPR and 50% in GES) and about 75% in the top 100 (75% in PPR and 79% in GES).

Figure 7.12: After removing L1 pages from D2C ranked lists, we show mean and median rank of target pages at different k values for the best PPR and GES settings. GES and PPR had similar performance when L1 pages are not considered in the evaluation.

Overall, our analysis showed that GES and PPR had similar performance when we do not consider L1 pages in our evaluation. However, excluding L1 pages is not always appropriate. An implementation of D2C-query ranking needs an approach that mixes and ranks pages from L1 and other levels effectively. Factors such as the diagram size and user preferences are important in choosing the right approach. For example, if the diagram used in a D2C query covers most of the domain (i.e., L1 pages constitute most of the domain), then it is important to show and rank L1 pages in the top N . In this case, an approach similar to the one used by PPR is preferred. However, when a diagram shows only an overview (a subset of the domain) and the user is more interested in seeing concepts not included in the diagram, then a GES-like approach

Figure 7.13: Distribution of 200 target-page ranks (L1 pages are removed) for PPR and GES best settings. GES and PPR had similar performance.

is more useful.

Finally, to study the performance of D2C queries on the different diagrams and graph classes in the Universal Model, we computed the mean and median rank of the target pages per diagram. Table 7.7 shows the results. We focus on medians as means are highly influenced by outliers. We see that both GES and PPR were robust to all graph classes in the Universal Model. In each graph class, we had some good performing queries (or diagrams) and bad ones. About 14 diagrams had relatively satisfactory performance with medians less than 20 when considering the GES ranks that excluded L1 pages (13 diagrams if we considered GES ranks that included L1 pages).

For the other six diagrams, we believe performance was harmed by the noise caused by context elements (i.e., diagram elements close to a D2C selected element). When

Table 7.7: Performance of GES and PPR in each diagram (diagrams used for tuning have shaded background). GES and PPR were robust to all graph classes in the Universal Model. Bad performance in some diagrams (e.g., WW2 Timeline and Data Science) was likely due to low relatedness between a D2C selected element and elements in its neighborhood (i.e., close elements acted as noise in the ranking process).

Diagram	Class	Rank Including L1 pages				Rank Excluding L1 pages			
		GES Avg.	GES Med.	PPR Avg.	PPR Med.	GES Avg.	GES Med.	PPR Avg.	PPR Med.
Attack Graph	Generic	318	55	450	39	304	43	426	29
Geological Eras	Generic	11	8	20	20	9	6	4	4
Nursing Career Flowchart	Generic	38	8	35	15	34	4	24	4
Programming Languages	Generic	322	10	112	46	317	9	80	12
South America Map	Generic	13	1	13	13	13	1	1	1
TV Networks	Generic	28	13	34	20	26	11	17	5
Cheese Types	I&A	89	19	104	63	73	14	73	34
Chocolate Ingredients	I&A	36	30	223	160	24	25	161	87
Data Platforms	I&A	272	79	422	127	219	47	340	21
Periodic Table	I&A	5	3	158	154	5	3	25	3
Rock Types	I&A	15	11	100	21	12	5	88	9
Portland Bridges	Path	19	13	8	8	16	10	2	2
Space Discovery Timeline	Path	798	648	368	327	793	643	360	320
WW2 Timeline	Path	774	213	403	94	769	208	391	83
Body Systems	Tree	27	14	146	32	23	9	125	13
Coffee	Tree	8	4	26	25	5	3	7	4
Data Science Tools	Tree	904	588	1463	237	815	495	1393	193
Culinary Herbs	Tree	22	17	35	27	17	16	20	12
JavaScript Libraries	Tree	104	17	119	22	94	7	102	6
Portland Attractions Map	Tree	351	24	285	34	341	15	272	21

context elements are highly related to the selected element, an algorithm can benefit from additional paths to related Wikipedia pages. However, if context elements are themselves unrelated (or only slightly related), they act as noise that disperses the search to different parts of the domain. For diagrams created specifically for MDN navigation, this problem can be mitigated by carefully designing those diagrams so each element is surrounded by some closely related elements (useful context). Alternatively, a diagram element can be connected to multiple Wikipedia pages to provide this useful context (e.g., when a diagram is already created). We discuss a related technique as well in Chapter 10, when we talk about future work.

7.5 Discussion Summary

In this section, we summarize our findings in this investigation and discuss our research questions listed in the beginning of the chapter.

- RQ1) Do Universal-Model diagrams positively bias the ranking process to retrieve pages related to the selected element in a D2C query? Similar to our investigation regarding I&A-Model diagrams, Universal-Model diagrams had a positive influence on ranking. We noticed this positive influence in both the GES and PPR algorithms. GES performed the best with high greediness values in L2 and L3. The results in Table 7.4 show that GES achieved the best performance when it mostly propagated energy from L1 pages (pages connected to a diagram). Increasing the number of processed pages in L2 and L3 in the setting (0.5, 0.75, 0.65) had a negative effect on performance (increased the mean from 96 in the best setting to 131 @ $k = 108$). PPR also performed the best with increasing the damping factor (0.65 to 0.75) (compared to related work [11, 108] that set damping factor to 0.15). This change in the damping factor reduced the mean @ $k = 108$ from 165 in (0.15, TT, 0.1) to 109 in (0.65, TT, 0.5) and also reduced the median @ $k = 114$ from 39 to 32.

When looking at the overall performance of D2C queries in Universal-Model diagrams, we see that the results are encouraging. Considering the best GES setting in Figure 7.11 (when including L1 pages), the mean target-page rank is 58 @ $k = 180$ and the median is 13.5 @ $k = 194$. However, we think a detailed study involving feedback from end users may bring more insight about the performance of D2C queries.

- RQ2) How much effect should the content and structure of the Universal-Model diagrams have on the ranking process? The results of tuning the GES and

PPR algorithms indicate that diagram impact on the ranking process should be high. As per the discussion in the previous question regarding greediness in GES and damping factor in PPR, we conclude that the best D2C query ranking is achieved when increasing the diagram influence.

- RQ3) What are the best weights for reciprocal and non-reciprocal links? From Figures 7.7 and 7.4, we see that both algorithms reached the best performance by giving non-reciprocal links about half the weight of reciprocal links. This weight scheme for MDN outperforms the recommendation of Agirre et al. [11] of keeping only reciprocal links. In PPR, the mean of weight 0.5 was 109 @k = 108 while for weight 0.01 the mean was 153 (note that weight 0.01 is close to using only reciprocal links). In GES, weight 0.1 was the closest to using only reciprocal links. The means of weights 0.4 and 0.1 were 93 and 106, respectively, @k = 108 (overall median was also better for weight 0.4).
- RQ4) Should we include hyperlinks that appear in article templates? Including article-template hyperlinks had a positive effect on PPR especially in the graph that included categories (comparing the TT and TF graphs). On the other hand, article-template hyperlinks had a large negative impact on GES. Templates can provide useful hyperlinks when the template's domain matches the domain or the concept of the article or the diagram used. However, templates can also act as noise if their domain is different. We see that PPR is able to cope with this noise better than GES.
- RQ5) Should we include Wikipedia categories? Agirre et al. [11] achieved the best performance of the PPR algorithm with a graph that discarded Wikipedia categories (included only articles). We observed a similar pattern for PPR in the TF graph, where PPR recorded the worst performance. However, PPR achieved

best performance when including both categories and template hyperlinks (the TT graph). For GES, the algorithm performed better when including categories. Overall, the parameters of categories and template hyperlinks inclusion had major influence on both GES and PPR. Even though we ran our experiments on small graphs compared to the full Wikipedia graph, we think testing for those parameters is important for research and applications that use the Wikipedia graph.

- RQ6) What are the best settings for the PPR and GES algorithms in D2C queries issued from Universal-Model diagrams? Which algorithm is better? Was the better algorithm robust for all graph classes in Universal-Model diagrams? The best settings for PPR and GES were (0.65, TT, 0.5) and (1, S6, TF, 0.4), respectively. Both algorithms showed consistent performance in all graph classes in the Universal Model. Also the two algorithms showed similar performance when we removed L1 pages from the ranked lists. However, when including L1 pages, PPR tended to rank a large number of those pages (including less relevant ones) in the top, which pushed down highly related pages in L2 and L3 (e.g., target pages). On the other hand, GES ranked a small percentage of L1 pages (the ones close to the selected element) in the top, which raised the rank of target pages.

We think that the choice of using GES or PPR depends on the diagram used and the target domain. GES can be useful when a diagram includes a small subset (showing few examples) of the domain concepts. In this case, we argue that a user is likely to be interested in related pages in the collection (pages in L2 and L3) as related L1 pages are visible in the diagram (i.e., the visual elements they connect to). Also, when the diagram includes diverse groups of

related elements (e.g., coffee drinks and coffee machines), GES focuses on the most related group, which yields different and more customized ranked lists in different D2C queries.

PPR, on the other hand, can be useful when a diagram includes most of the concepts in the domain. In this case, the ranking algorithm needs to focus on L1 pages as most relevant pages are in this level. However, it is important to note that we do not know much about how PPR ranks L1 pages (our study focused on target pages which are in L2 or L3). For example, in addition to the distance to the selected element, ranking can also be influenced by the degree of visual elements and L1 pages.

8 Evaluating Approaches for C2D-Query Computation

We focused most of our research regarding Extended MDN (EMDN) on D2C queries, including the two investigations that we described in the previous two chapters. In this chapter, we switch our focus to C2D queries and evaluate multiple approaches to compute them. We hypothesize that C2D queries can be computed using the ranked lists we get from D2C queries. We describe four approaches that use this idea. Also, we describe a native approach that start traversing the graph from a Wikipedia page selected by the user.

Similarly to D2C queries, we study C2D queries that include a single selected page. Assuming a user selects a page xp in Figure 8.1, our goal is to rank diagram elements according to their relatedness to xp . If the page xp is connected to the diagram (L1 page), then ranking diagram elements is relatively easy. A simple approach can rely on the distance between the element that xp connects to and other diagram elements. In this investigation, we focus on the case where xp is not in L1.

The ranking of a C2D query can be displayed on the diagram in different ways. One approach is to display a heat map of diagram elements based on their scores or ranks. Another approach can highlight only the top few elements on the diagram. Highlighting few elements allows users to explore nearby related elements in an experience similar to exploring places close to a specific location on a spatial map. In this chapter, we do not study the support of a specific approach of displaying C2D ranking. However, we do discuss when a certain approach can be better supported

by MDN C2D ranking.

In the next section we introduce five different approaches to compute C2D queries. In Section 8.2, we list our research questions for this investigation. In Section 8.3, we describe our evaluation methods and in Sections 8.4 and 8.5 we show and discuss the results. Finally, in Section 8.6, we summarize and answer the research questions.

8.1 Approaches for C2D-Query Computation

The first approach to compute C2D-query ranking is based on using the PPR algorithm with the page selected by the user as the base set (teleport vector). This approach simply directs the random surfer to teleport to a Wikipedia page (instead of teleporting to a selected diagram element as when computing a D2C query). Then, we rank diagram elements using their PageRank scores (we omit scores of Wikipedia pages). We refer to this approach as PPR-C2D.

A similar approach can use GES where we propagate energy from the selected Wikipedia page. However, we designed GES to start from diagrams for D2C queries. Starting energy propagation from a single Wikipedia page is likely to miss the rich context diagrams provide. Therefore, we did not evaluate this approach.

A key difference between D2C and C2D queries is the chance of finding related nodes (finding related Wikipedia pages in D2C or related diagram elements in C2D). Given a diagram D and the Wikipedia collection W , we know that $|D| \ll |W|$. Therefore, it is more likely that a diagram element $e \in D$ to have related Wikipedia pages than for a Wikipedia pages $p \in W$ to have related diagram elements. In other words, most Wikipedia pages do not have related elements in a particular diagram. Especially for a Wikipedia page q that is slightly related to a diagram D , a ranking algorithm would need to travel far away from q to reach and rank elements in D . PPR is able to do that by using a low damping factor. GES on the other hand, is designed

to start from a number of related elements in the diagram. Then, by using greediness, GES directs energy spreading towards related pages in Wikipedia. Therefore, if GES starts from an unrelated (or slightly related) page in Wikipedia, it may spread energy far away from the diagram. A possible tuning for GES would decrease greediness to increase the chance for diagram elements to receive energy. However, we do not study this approach in this thesis.

Figure 8.1: An example graph with the page xp selected in a C2D query.

The second approach we consider to compute the ranking of C2D queries is based on using D2C-query ranked lists returned from PPR. Given a selected page xp , we look at the rank of xp in the ranked lists for all (single-element) D2C queries issued from the diagram. We first define $d2c\text{-rank}(e, xp)$ as the rank of page xp when the diagram element e is selected in a D2C query. Consider two diagram elements b and f that rank the page xp (when b or f is selected in a D2C query) in the positions $d2c\text{-rank}(b, xp)$ and $d2c\text{-rank}(f, xp)$, respectively. In the C2D query (when xp is selected), we rank b before f if $d2c\text{-rank}(b, xp) < d2c\text{-rank}(f, xp)$ (i.e., $d2c\text{-rank}(b, xp)$ has a higher rank) or rank f before b if $d2c\text{-rank}(f, xp) < d2c\text{-rank}(b, xp)$. If $d2c\text{-rank}(b, xp) = d2c\text{-rank}(f, xp)$, we randomly choose either b or f to be ranked higher. For example, assume the elements b , c , and f in Figure 8.1 rank the page xp in the positions 15, 10, and 22, respectively. When xp is selected in a C2D query, we rank the three diagram elements as c , b , then f . We refer to this approach as PPR-D2C-Rank since it uses

the ranks of a page in the D2C lists returned by PPR.

The third approach is mostly similar to the previous one. The only difference is that we look at the scores (instead of ranks) of the page xp in the D2C lists. We call this approach PPR-D2C-Score. The fourth and fifth approaches use the D2C ranked lists returned by the GES algorithm. We refer to those approaches as GES-D2C-Rank and GES-D2C-Score. Those two approaches follow the same method used by their counterparts PPR-D2C-Rank and PPR-D2C-Score, respectively (except for using GES D2C lists instead of the PPR D2C lists).

8.2 Research Questions

We designed this investigation to answer the following research questions:

- RQ1) What is the best PPR setting to compute C2D ranking using the PPR-C2D approach? The best setting for PPR in D2C queries may not be the same when computing C2D queries (as we start from a Wikipedia page and not from a diagram element). Therefore, we wanted to tune PPR specifically for C2D queries.
- RQ2) How does using GES or PPR D2C ranked lists compare to using the native approach (PPR-C2D) in computing the ranking of C2D queries?
- RQ2) If using D2C ranked lists is sufficient to compute C2D ranking, which one of the approaches that use those lists has the best performance?

Overall, we only compare approaches to compute C2D queries. Knowing the best approach, we can study the performance or accuracy of C2D-query ranking in detail in future research projects. In Chapter 10, we describe a direction of future work that evaluates the performance of C2D queries.

8.3 Methods

We used the set of 200 queries that we used in our Universal-Model-Diagram investigation. The main difference is that we select a page instead of a diagram element and we rank diagram elements instead of ranking Wikipedia pages.

To evaluate the five approaches we described in Section 8.1, we needed to compare the ranked lists we get from those approaches to a list that represents an accurate ranking of diagram elements. We call this list the *reference list* and we describe it in detail next.

Figure 8.2: An example graph with the page *ap* selected in a C2D query. After disconnecting *ap* from its related diagram element *a*, we can evaluate a C2D ranking with the help of the structure provided by the diagram (e.g., we expect *a* to be ranked in the top and *f* in the bottom)

Consider the graph in Figure 8.2 where the diagram element *a* is connected to the Wikipedia page *ap*. Assume that we disconnect the link between *a* and *ap*. The page *ap* is not in L1 anymore, which is the case we are interested in as we described earlier in this chapter (we assume here that *ap* is only connected to one diagram element *a*, which is the usual case in the 20 diagrams we used). Assume *ap* is selected in a C2D query and we need to rank diagram elements. A good ranking of diagram elements should place *a* in the top position. Given that *a* is the most related element

to the selected page, we can benefit from the ranked list returned when a is selected in a D2C query. This ranked list is mostly based on the diagram structure, which is desirable (we assume users are interested in the diagram perspective). For example, GES ranks diagram elements only based on their distance to the selected element (a in this case). Also, PPR uses a high damping factor that increases the diagram influence on ranking the diagram elements. Therefore, we consult the ranked lists we get from both GES and PPR in D2C queries to build the reference list that we use in our evaluation of the five approaches described earlier. We explain how we built this list next.

For each query in the set of 200 queries used in our experiment, we issued D2C queries using the best setting in both the GES and PPR algorithms. For those D2C queries we kept the link between the selected element and its connected page(s) (no page disconnecting since we are only interested in the ranking of diagram elements). Given that score ranges in GES and PPR can be different (in GES, scores of diagram elements are usually higher), we normalize the scores of diagram elements as follows:

$$GES\text{-}norm(e) = GES\text{-}raw(e) / \sum_{v \in D} GES\text{-}raw(v) \quad (10.1)$$

$$PPR\text{-}norm(e) = PPR\text{-}raw(e) / \sum_{v \in D} PPR\text{-}raw(v) \quad (10.2)$$

where e and v are diagram elements; D is the set of diagram elements; $GES\text{-}raw(x)$ and $PPR\text{-}raw(x)$ are the scores of x before the normalization in GES and PPR, respectively; and $GES\text{-}norm(x)$ and $PPR\text{-}norm(x)$ are the scores of x after normalization in GES and PPR, respectively. For each algorithm, we divide the score of each diagram element by the sum of the scores of all diagram elements. The normalized scores for each algorithm form a distribution that sums to one. We also tested a normaliza-

tion method where we divide each element's score by the maximum diagram-element score, and we found out that the two methods yield the same reference list.

To get the final score $Score(e)$ of a diagram element e in the reference list, we average the normalized scores of e in GES and PPR.

$$Score(e) = (GES\text{-}norm(e) + PPR\text{-}norm(e))/2 \quad (10.3)$$

We sort the diagram elements by their final scores and obtain the rank $rank(L, e)$ for each element e in the reference list L . The only issue we had is that diagrams had different sizes. Therefore, it would be hard to interpret ranks and statistics based on 200 queries from different diagrams. To tackle this issue, we converted ranks in the reference lists to scaled ranks (relative to diagram sizes) as follows.

$$srank(L, e) = rank(L, e)/|D| \quad (10.4)$$

where $srank(L, e)$ is the scaled rank of e in the reference list L . Note that the range of scaled ranks is between $1/|D|$ and 1.

Assume we have a D2C query q_a with a selected diagram element a and a C2D query q_{ap} with a selected Wikipedia page ap . Using the reference list L of a , we can evaluate the goodness of a C2D-computation approach in the query q_{ap} when a and ap are connected (we cut the connection for the sake of the experiment). To compare the ranked list returned from a query q_{ap} using one of the tested approaches p and the reference list L , we compute the *Average Pairwise Comparison* (APC) as follows:

$$APC = \frac{\sum_{v \in D} ABS(srank(p, v) - srank(L, v))}{|D|} * 50 \quad (10.5)$$

where ABS is the absolute value, and $srank(p, v)$ is the scaled rank of the element v

returned by the approach p (similar to the reference list, we divide the original ranks returned by p by the size of the diagram). APC simply computes the average change (shift) in the rank of diagram elements. To simplify our discussion of the performance of the different approaches, we normalize the score by multiplying by 50. We consider 50 to be a typical diagram size. The set of the 20 diagrams used in our experiments have an average diagram size equals to 52. With this normalization, the minimum APC is zero, which indicates the ranking of a C2D query matches the reference list, while the maximum is 25 (e.g., when the ranking returned by p is the reverse of the reference list).

To further explain APC, consider the example in Figure 8.2. Assume that the reference list ranks the diagram element as a, b, c, d , then f with the ranks $1/5, 2/5, 3/5, 4/5$, and $5/5$, respectively. Assume that GES-D2C-Rank, which is one of the approaches to compute C2D queries, ranks the diagram elements as d, a, b, c , then f with the ranks $1/5, 2/5, 3/5, 4/5$, and $5/5$, respectively. Then, the APC for GES-D2C-Rank in this C2D query is $((ABS(\frac{1-2}{5}) + ABS(\frac{2-3}{5}) + ABS(\frac{3-4}{5}) + ABS(\frac{4-1}{5}) + ABS(\frac{5-5}{5}))/5) * 50 = 12$. Note that the average change of element ranks in the original diagram (with size 5) was 1.2 (about one fourth of the size of the diagram). When normalizing that to a 50-element diagram, the average becomes 12. One interpretation of the APC value for this query is that compared to the reference list, GES-D2C-Rank shifted the rank of each diagram element 12 positions on average in a diagram of size 50 (1.2 position in a diagram of size 5 or 24 positions in a diagram of size 100).

Note that APC is computed for a single C2D query. To have an overall metric for a set of queries (such as the set of 200 queries), we compute both Mean Average Pairwise Comparison MNAPC and Median Average Pairwise Comparison MDAPC. Similar to the previous chapter, we compute these metrics at different k values.

8.4 Tuning PPR-C2D

In this section we describe our tuning for PPR-C2D. Note that the best setting for PPR in D2C queries cannot be assumed to be the best in C2D queries, too. Therefore we tuned PPR to work starting from a Wikipedia page.

We assumed that D2C and C2D queries will run on the same graph in a future MDN implementation. Therefore, we used the TT graph to tune PPR-C2D since it was the best graph for PPR D2C queries. We tested 32 settings to test all combinations of the eight damping factor values and the four non-reciprocal-link weights in Table 7.2. We used 120 queries issued from the first 12 diagrams in Table 7.1.

In Figure 8.3, we show the results of $(0.15, 0.5)$ and $(0.85, 0.01)$ where the first parameter is the damping factor and the second one is the non-reciprocal-link weight. These settings were the best and worst setting, respectively. We show MNAPC and MDAPC at different k values (number of best queries). The performance was close for all settings. In general, the parameter of non-reciprocal-link weights had almost no effect on performance. In contrast to D2C queries, for C2D queries, PPR performed better with small damping factor values. High damping factors are less helpful in C2D queries since the selected page has relatively less useful context (compared to a selected element in D2C query, which has the context of the diagram). In fact, the low damping factor value 0.15 may have allowed the random surfer to visit diagram elements more frequently given that those elements were relatively far away from the selected page (recall that the selected page was in L2).

We use the setting $(0.15, 0.5)$ for PPR-C2D in the next section when we compare the five approaches to compute C2D queries. Thus, in the next section, when we refer to PPR-C2D we implicitly refer to this setting.

Figure 8.3: The best and worst setting for C2D-PPR (i.e., running PPR from a selected page). The low damping factor value 0.15 performed better from the high one 0.85 (opposite behaviour from the one we saw in D2C queries).

8.5 A Comparison of Five Approaches to Compute C2D Queries

The main experiment was about comparing the five approaches to compute C2D queries, which are PPR-C2D, PPR-D2C-Rank, PPR-D2C-Score, GES-D2C-Rank, and GES-D2C-Score. We used the full set of 200 C2D queries issued from 20 diagrams (10 per diagram). We computed MNAPC and MDAPC at different k values. Figure 8.4 shows the performance of the five approaches. PPR-C2D was the best choice with overall (@k = 200) MNAPC and MDAPC equal to 12.17 and 11.73, respectively. With overall MNAPC and MDAPC equal to 12.37 and 12.36, respectively, GES-D2C-Rank was close to C2D-PPR indicating that an approach based on using D2C-query ranked lists to compute C2D queries is indeed reasonable. The difference between the two approaches is about a half position on average in a diagram of size 50. GES-D2C-Rank was also close to PPR-D2C-Rank (GES-D2C-Rank was better by about a half position). We also notice that approaches that used D2C ranked lists performed better when using ranks instead of scores of the C2D-query selected page.

As we discussed in the beginning of this chapter, one of the approaches to display the ranking of C2D queries is to highlight only the top few elements in the diagram

Figure 8.4: Performance of five approaches of computing C2D queries. GES-D2C-Rank was very close to C2D-PPR indicating that computing C2D queries using D2C-query ranked lists is reasonable.

(instead of displaying a heat map and highlighting all diagram elements based on their scores or ranks). To study this particular display method, we focused on the rank of the only top element in the reference list (i.e., the element whose link to the selected page was disconnected). We considered only this element and computed MNAPC and MDAPC (where D in Equation 10.5 contains only the top element). Note that since this element's rank is 1 in the reference list, the MNAPC and MDAPC metrics simply show how many positions this element was shifted downward. In this case, the minimum and maximum values for these metrics are 0 and 49, respectively. Also, this metric is symmetric to our metric used in the previous chapter with D2C queries (instead of measuring the rank of a Wikipedia page whose rank should be one, we measure the rank of a diagram element whose rank should be one).

In Figure 8.5, we show the performance of the five approaches in ranking the most related element in the diagram. The best approach was GES-D2C-Rank with overall MNAPC and MDAPC equal to 7.57 and 3.48, respectively. Considering only 180 queries (90% of the 200 queries), the MNAPC was 4.63. These numbers show that ranking only the top element is more accurate than ranking all of the elements in the

Figure 8.5: Performance of five approaches of computing C2D queries in ranking the most related element in the diagram. GES-D2C-Rank clearly outperforms C2D-PPR. GES-D2C-Rank is also better than PPR-D2C-Rank (difference is about 5.65 positions when considering MNAPC and 4.36 positions in MDAPC).

diagram. Therefore, we argue that the accuracy of highlighting the top few elements is likely to be higher than showing a heat map that highlights all elements in the diagram. We think that any future studies regarding C2D queries should take this observation into consideration.

We note that PPR-C2D was the worst approach in Figure 8.5. One reason for the bad performance of PPR-C2D is that the top element in the diagram (e.g., a in Figure 8.2) is the only element in the diagram that has no connections to Wikipedia. Therefore, it gets less visits from the random surfer.

Comparing GES and PPR in Figure 8.5, we see that GES-D2C-Rank performed better than PPR-D2C-Rank (the difference is about 5.65 positions when considering MNAPC and 4.36 positions in MDAPC). One factor that may contribute to this difference is the way GES favors close elements in the diagram when propagating energy (i.e., using Dist1). Therefore, different groups or sides in the diagram mutually lower the rank of related pages of the other group or side. For example, in Figure 8.2, consider two D2C queries that select a and f . The difference of the rank of the page

previous chapter, this useful context can increase the performance of D2C queries, which in turn increases the performance of C2D queries for some of the approaches (since we rely on D2C lists to compute C2D in those approaches).

In addition to the usefulness of having related elements in each cluster, C2D ranking can benefit from having only relatively unrelated elements in other clusters. For example, in Figure 8.6, the rank of the ‘CBS News’ page when the ‘CBS News’ element was selected in a D2C query was 132 (it was 55 for ‘Showtime’, 134 for ‘CBS’, and 195 for ‘CBS Sports’). Even though those D2C queries did not rank ‘CBS News’ in the top 10 or 20, other D2C queries issued from other clusters in the diagram had even lower ranks (e.g., the rank of ‘CBS News’ page was 943 when ESPN element was selected and 5001 when Disney element was selected). This large contrast in the ranks helped generate the C2D ranking shown in Figure 8.6, which successfully highlighted elements close to ‘CBS News’.

One last note about the example in Figure 8.6. We see that having elements in the same group visually close to each other in the diagram can help the user make sense of the query results. Errors in the ranking can be spotted as outliers and may be easily ignored. For example, ‘ABC News’ element was in the top five and highlighted by MDN (possibly because the selected page is a news channel)¹. However, the user can notice that the selected page (‘CBS News’) is probably more related to the CBS group by seeing many elements in the top five there.

¹The aspect of the diagram in Figure 8.6 is based on grouping channels by their parent company. Therefore, we consider the ‘ABC News’ element to be unrelated (or slightly related) to the selected page ‘CBS News’. Note that two channels can be highly related in another diagram that groups channels by genre.

8.6 Discussion Summary

In this section, we summarize our findings in this investigation and discuss our research questions listed in the beginning of the chapter.

- RQ1) What is the best PPR-C2D setting? The best setting for PPR-C2D was $(0.15, 2)$. Compared to D2C queries where PPR favored large damping factor (d) values, in C2D queries, the algorithm worked the best when setting d to 0.15. Given that the algorithm did not have the context of the diagram, making the random surfer return to the selected Wikipedia page did not seem to increase the performance of the ranking of diagram elements. In fact, allowing the random surfer to go further (from the selected page) may have increased the chance of visiting diagram elements and therefore enhanced the ranking of those elements.
- RQ2) How does using GES or PPR D2C ranked lists compare to using the native approach (PPR-C2D) in computing the ranking of C2D queries? Figure 8.4 shows that PPR-C2D and GES-D2C-Rank, which relied on D2C ranked lists, had close performance. The difference was about 0.5 position when considering MNAPC @ $k = 180$ and 0.6 position when considering MDAPC @ $k = 200$. Furthermore, in the task of ranking the top element in the diagram (Figure 8.5), GES-D2C-Rank outperformed PPR-C2D. These results show that we can indeed rely on D2C-query ranked lists to compute the ranking of C2D queries.
- RQ3) If using D2C ranked lists is sufficient to compute C2D ranking, which one of the approaches that use those lists has the best performance? The results in Figures 8.4 and 8.5, show that GES-D2C-Rank was the best approach among the ones that used D2C ranked lists. While the performance of GES-D2C-Rank and PPR-D2C-Rank was close in Figure 8.4 (GES-D2C-Rank was better by

about half a position), the difference between the two approaches was bigger in Figure 8.5 (when considering only the rank of the top element). In GES-D2C-Rank, the mean rank of the most related element in the diagram was 5.63 @k = 180 (median was only 4.4 @k = 200) while in PPR-D2C-Rank the mean was 11.2 @k = 180 (median was 8.8 @k = 200). For both algorithms, using ranks of the C2D selected page was better than using scores.

In this chapter, we covered our last investigation which analyzed different approaches to compute C2D queries. In the next chapter, we switch to related work and discuss different research projects that tackled similar problems or used similar techniques to the ones we have in MDN.

9 Related Work

In this chapter, we describe work related to MDN. We start by describing some popular solutions that target exploratory search in Section 9.1. In Sections 9.2 and 9.3, we focus on induced visualizations and entity ranking, two exploratory search approaches that present similar functionality to MDN. Section 9.4 lists exploratory search solutions that focus on the Wikipedia collection. In Section 9.5, we talk about Word Relatedness (WR) and Named Entity Disambiguation (NED), two applications that are similar to MDN in terms of using the Wikipedia collection and the PPR algorithm. In the last three sections, we discuss related work that used techniques we use in MDN. In Section 9.6, we describe superimposed information and list some projects that used the technique in supporting navigation of collections. In Sections 9.7 and 9.8, we describe related work that used the PPR and energy-spreading algorithms, respectively.

9.1 Exploratory Search Tools

Exploratory search tools have used different approaches to provide interactive IR interfaces. A popular approach is faceted-search interfaces that are included in online shopping websites such as Amazon and eBay. Faceted search systems usually start with a keyword search to retrieve an initial set of data items or documents. The system extracts important properties (facets) from this set to be presented to the user for further filtering. For example, searching for ‘candy bar’ in Amazon presents

facets such as brand, fat content, and specialty food types (e.g., kosher and gluten free) for the returned candy-bar items.

MDN can provide facets through diagrams showing multiple aspects of a domain, such as the language diagrams shown in Figure 3.1. A facet in MDN provide an overview to the domain from a particular aspect. The overview facets can organize concepts in different structures and provide some example concepts to help users understand the facet. Note that in this research we do not provide a full faceted search system as we do not study the process of combining queries issued from multiple diagrams. For example, in Figure 3.1, the system does not support filtering Wikipedia pages by combining two D2C queries issued from the language forest and the language map diagrams. However, this feature could be easily added to the system. A naive ranking algorithm can average page scores in separate queries issued from multiple diagrams.

Social bookmarking is another exploratory search tool that can support exploratory search in multiple ways [80]. Delicious is one of the early and popular social-bookmarking services that allows users to store, tag, and share their webpage bookmarks. (Delicious was acquired by Pinboard, another social bookmarking service.) Delicious allows users to search using tags to retrieve their own bookmarks as well as bookmarks shared by other users. Accessing bookmarks similar to a user's bookmarks facilitates the discovery of interesting content that might be missed when using traditional web-search engines. A fundamental difference between MDN and social bookmarking tools is that the latter mostly target web content where MDN targets a document collection. In addition, social bookmarking systems only rely on induced visualizations (mainly tag clouds) to provide an overview of the content. In MDN, we can present different types of overviews.

Another popular exploratory search tool is providing recommendations in large collections such as YouTube, Amazon, and Netflix. MDN is similar to systems that use graph-based approaches such as in Twitter [73] and YouTube [20]. These approaches start at some content (nodes in the graph) that is interesting to a user and traverse the graph to find highly related content, which is then recommended to the user. We discuss some of these systems in Sections 9.7 and 9.8, when we discuss PPR and energy spreading algorithms, respectively.

9.2 Induced Visualizations

Multiple research projects in exploratory search have focused on providing visualizations extracted from documents, collections, or datasets. These visualizations provide overviews and allow users to manipulate the data in different ways [104]. Sarrafzadeh et al. [92] extract a knowledge graph (concept map) from the text of a single document. Hall and Clough [56] use thesauri attached to documents in a digital library to create a map-based visualization (“resembling Google Maps”). Rauber and Bina [89] create a bookshelf visualization by mapping metadata to visual metaphors. Book attributes such as size, publisher, and the amount of dust on the book are used to create a book representation on the bookshelf. Alonso [12] extracts a timeline from DBLP search results. Dork [42] provides a spatial map (based on Google Maps), word cloud, and a timeline for web-search results. These visualizations (because they are induced) provide connections between the overviews and the content automatically, but they are limited to specific diagram and visualization types. MDN differs in: (1) the types of diagrams that can be used and (2) how connections among overviews and documents are created. We argue that some aspects useful to exploratory searchers can only be shown using a wide range of diagrams.

MDN draws on work [61, 82, 103] intended to help data analysts understand data

sets by showing Coordinated Multiple Visualizations (CMV) such as histograms and scatter plots. The visualizations are based on the same data set. The MDN interface can also show multiple diagrams that are linked to the same collection. CMV research is mainly led by the HCI community and focuses on user interfaces, visualization and coordination techniques. CMV can also target collections of documents. For example, Thudt [100] uses CMV to visualize metadata of a collection of books to allow library users to serendipitously find books using facets such as book author, color, and size.

9.3 Entity Ranking and Recommendation

Another exploratory search approach is entity ranking and recommendation in web search. Many information needs are related to entities such as people, products, and locations; these needs are better served by presenting related entities instead of web pages [19]. Pound et al. [87] found that 40% of queries submitted to a commercial search engine are about specific entities with additional 12% of the queries are about entity types (e.g., doctors or attractions). Popular web search engines including Google, Bing, and Yahoo! display related entities along with search results. Figure 9.1 shows entities recommended by the Google search engine when searching for ‘Portland State University’. Entity recommendation is based on linking the entities in the search keywords to a knowledge base (e.g., the Google Knowledge Graph [95]). Then, the graph of the knowledge base is used to retrieve and rank related entities.

Entity recommendation provided by web search engines targets the web, compared to MDN, where we targeted collection exploration (theoretically, MDN can support the exploration of a part of the web, however, we have not studied this scenario). Also, the context provided by diagrams is richer than that provided by search keywords. In addition, compared to MDN, where we focused on the graph of Wikipedia, web-scale entity ranking rely on larger knowledge graphs. For example, the knowledge

Figure 9.1: Related entities to ‘Portland State University’ (displayed in the bottom right) ranked by the Google search engine.

graph used by Yahoo! Spark [25], the system that ranks and recommends entities in the Yahoo! web search, includes information from Wikipedia, Flickr, and Twitter among other sources [25, 79]. Google and Bing have not disclosed information about the knowledge graphs they use.

One research project that provides entity ranking in the Wikipedia collection was conducted by Vercoistre et al. [102]. The task, which was designed by the entity-ranking track in INEX 2007 [38], involves few query terms and example entities. The goal is to return related entities (pages) from Wikipedia that fulfil the search terms and are similar to the ones in the examples. For example, given a query “European countries where I can pay with Euros” and the examples: France, Germany, and Spain, the task is to find other countries where Euros are accepted [102]. The approach presented starts by using a search engine to retrieve pages that are similar to the query.

Then, those pages (entities) are further ranked using three factors: the similarity between a page’s text with the query, the number of common categories between the page and the example entities, and the number of links the page has to the top N pages returned by the search engine (N was set to 20). The latter factor is also weighted by the number of links each page in the top N has to the example entities.

The targeted scenario for Vercoustre et al. [102] is similar to the MDN D2C query type given that the query terms and the supplied examples provide context to retrieve related entities (pages). However, diagrams provide richer context in terms of the number of provided elements (entities) and the structure among them (the number of supplied examples in the proposed task is two or three). In addition, given that the query is written in natural language, it may contain irrelevant entities, which can be misleading (e.g., a query asking for countries where Euros are not accepted). Such context is not possible in diagrams given that diagrams are manually prepared and mapped. In addition, Vercoustre et al. [102] used only a small subset of the graph of Wikipedia compared to MDN. For instance, links among the example entities and most of the links among the top N pages are not considered.

9.4 Exploratory Search Tools For Wikipedia

Multiple research projects focused on providing exploratory search support for the Wikipedia collection. One of the early attempts was the Wikipedia Faceted Search (WFS), a work of Hahn et al. [55]. The interface of this project allows users to use facets of the entities (pages) in the search results for further exploration. For example, assume that a user is looking for rivers that have a specific length range and that flow into the Rhine river. A keyword search for ‘rivers’ in Wikipedia returns 12,432 pages. Given these pages, the project interface displays common facets as shown in Figure 9.2. In that display, a user can refine the search by typing ‘Rhine’ in the ‘has

mouth on' facet and the desired length range in the 'length' facet. Wikipedia Faceted Search relies on structured knowledge extracted from Wikipedia using the DBpedia Extraction Framework [24].

The screenshot shows the DBpedia Faceted Search interface. At the top, there is a search bar (labeled 1) with the text "enter search terms..." and a "Search" button. Below the search bar, there are links for "About DBpedia", "About neofonie", and "Contact". The main interface is divided into several sections. On the left, there is a sidebar (labeled 2) with facets for "item type", "has mouth at", "length (m)", "watershed (km²)", and "name". Each facet has a search input and a list of options. The "item type" facet shows "River (26)", "Place (26)", and "Body Of Water (26)". The "has mouth at" facet shows "Rhine (26)", "Ingelheim am Rhein (1)", and "Lahnstein (1)". The "length (m)" facet shows a range from "50000" to "90000" with "2" results. The "watershed (km²)" facet shows a range from "827" to "28286" with "1" result. The "name" facet shows "Pfinz (1)" and "Rench (1)". In the center, there is a "Your Filters" section (labeled 3) with "Reset Filters" and a list of filters: "item type River", "has mouth at Rhine", and "length (m) 50000 and up". On the right, there is a "Results 1 to 6 of 26" section (labeled 4) showing a list of search results. Each result includes a thumbnail image, a title, and a description. The results shown are "Main", "Moselle River", "Neckar", and "Ruhr (river)".

Figure 9.2: The Wikipedia Faceted Search [55].

A similar faceted search interface is Swipe by Atzori and Zaniolo [14] that provides a search-by-example interface (similar to Query By Example in databases). Similar to the previous project, Swipe relies on the DBpedia knowledge base. Swipe uses infoboxes as an entry form for search keywords. For instance, a user looking for a specific type of entities such as movies can start their search by navigating to the

article of a random movie. Swipe modifies the HTML code of the article's infobox to allow the entry of search keywords. Then, the user can enter his or her search keywords such as a director's name and a movie's length. The system uses these keywords to generate a SPARQL query to retrieve the required data from the DBpedia knowledge base.

Facetedpedia by Li et al. [70] is another faceted search system for Wikipedia. In Facetedpedia, facets are drawn from the Wikipedia category structure. After performing a keyword search, the system finds attribute articles, which are articles referenced by the articles in the search results. For example, if the search is about 'Computer Scientists', the attribute articles may include the country of origin of the scientists in the search results or universities where those scientists worked at or graduated from. The system navigates the category structure of Wikipedia starting from the attribute articles to find common categories (with multiple paths to attribute articles). Those categories with their immediate subcategories are shown as facets (two-level hierarchical facets). In the previous example, facets such as 'Association of American Universities' and 'Software companies of the United States' are displayed. Figure 9.3 shows the displayed facets [6].

One similarity between MDN (in the I&A-Model experiment) and all projects discussed in this section is that the search process starts at a set of pages (i.e., items). MDN collects items in the TU set; The WFS and Facetedpedia used search results; and Swipe used an example entity that represents other similar entities. Then, all systems used the fact that the starting item pages are linked to some attribute pages that may represent interesting facets. The links between item pages and attribute pages used by Swipe and the WFS are extracted from infoboxes. MDN and Facetedpedia use links from the entire body of articles. Another difference is that MDN and

Facets	Selected Categories	
Association of American Universities > [22]	Wikipedia Articles	
Carnegie Mellon University . [3] California Institute of Technology . [1] University of California, Los Angeles . [1] {Stanford University} . [5] {University of California, Berkeley} . [3] [SEE MORE]	(50 articles selected)	
Software companies of the United States > [13]	John McCarthy (computer scientist) John McCarthy (born September 4, 1927, in Boston, Massachusetts), is an American <i>computer scientist</i> and cognitive scientist who received the Turing Award in ... http://en.wikipedia.org/wiki/John_McCarthy_(computer_scientist)	Edsger W. Dijkstra Among his contributions shortest path-algorithm, algorithm; Reverse Polish http://en.wikipedia.org/wiki/Edsger_W._Dijkstra
Adobe Systems . [2] Apple Inc. . [3] {PARC (company)} . [5] {Kurzweil Educational Systems} . [1]	Donald Knuth /kəˈnuːθ/ kə-NOOTH; born January 10, 1938) is a <i>computer scientist</i> and Professor ... In addition to his writings on <i>computer science</i> , Knuth, a Lutheran, ... http://en.wikipedia.org/wiki/Donald_Knuth	Edward Felten Edward William Felten (I professor of <i>computer science</i> Princeton University. On named ... http://en.wikipedia.org/wiki/Edward_Felten
Computer hardware companies > [10]	Andrew S. Tanenbaum Andrew Stuart "Andy" Tanenbaum (sometimes referred to by the handle ast) (born March 16, 1944) is a professor of <i>computer science</i> at the Vrije Universiteit, ... http://en.wikipedia.org/wiki/Andrew_S._Tanenbaum	Alan Turing He was highly influential <i>computer science</i> , provided says 'Father of computing' http://en.wikipedia.org/wiki/Alan_Turing
Motherboard companies . [2] Sun Microsystems . [2] Networking hardware companies . [5] {IBM} . [2]	Computer storage companies . [2] Apple Inc. . [3] Home computer hardware companies . [3] {Cray} . [1]	Alan Kay Alan Curtis Kay (born March 17, 1941) is an American <i>computer scientist</i> , known for object-oriented programming. http://en.wikipedia.org/wiki/Alan_Kay
Companies established in the 1980s > [6]	John Backus John Warner Backus (December 3, 1924 ♦ March 17, 2007) was an American <i>computer scientist</i> . He directed the team that invented the first widely used ... http://en.wikipedia.org/wiki/John_Backus	Fred Brooks Fred M. Brooks (born August 16, 1923) is an American <i>computer scientist</i> and author of the book <i>The Mythical Man-Month</i> . http://en.wikipedia.org/wiki/Fred_Brooks
Companies established in 1982 . [4] Companies established in 1989 . [1]		
Visitor attractions in the United States by city > [6]		
Visitor attractions in Silicon Valley . [5] Visitor attractions in Washington, D.C. . [1]		
20th century books > [8]		
1980s books . [3] 1970s books . [1]		

Figure 9.3: Facetedpedia user interface after searching for ‘computer scientist’ articles [6].

Facetedpedia used only facets whose values are links to other pages (e.g., an ingredient of a candy bar or the alma mater of a computer scientist). In addition to these facets, Swipe and the WFS also included facets whose values are numeric or alphabetic (e.g., a river or a movie length). In terms of the links used, all discussed faceted-search projects used only outgoing links from item pages to attribute pages. In MDN, we extended the usage of the hyperlink structure by also considering incoming links to item pages (and hence reciprocal links). We also included other hyperlinks such as links among item pages.

The use of induced visualizations is another approach for providing exploratory search interfaces for the Wikipedia collection. Bekkerman and Donin [21] create a two-level graph visualization of Wikipedia categories. Pang et al. [85] provide another visualization for the Wikipedia category structure using cartographic visualizations [96] shown in Figure 9.4. The visualization presents categories as areas in the map with related categories placed in close proximity. Subcategories are represented as

internal territories inside the areas of their parent categories. The size of a category area was used to represent the number of subcategories and articles in the category while the color of the area represents only the number of articles. As discussed in Section 9.1, these overviews are dynamically induced visualizations. In MDN, we extend the choices of Wikipedia overviews by superimposing any type of diagrams.

Figure 9.4: A visualization of Wikipedia categories [23]

9.5 Word Relatedness and Named Entity Disambiguation

Word relatedness (WR) and named-entity disambiguation (NED) are two Natural Language Processing (NLP) applications related to MDN. Techniques for these applications are relevant to computing MDN D2C queries. However, there are some differences, as we discuss shortly in this section.

In NED, the task is to disambiguate a target mention (named entity) in a given text (context) and link the mention to the appropriate article in Wikipedia. For example, given the sentence ‘The Willamette river passes through downtown Portland’, the task of an NED system is to disambiguate and link the word ‘Portland’ to ‘Portland, Oregon’ and not ‘Portland, Maine’. In WR, a system measures the relatedness

between two entities returning a high score if they are closely related. For example, we expect the relatedness score between the pair ‘Willamette’ and ‘Oregon’ to be higher than the pair ‘Willamette’ and ‘California’. Some WR systems expect as an input a pair of Wikipedia articles representing the two entities (i.e., no disambiguation is required). Other systems expect a pair of mentions where each mention may have different meanings and therefore connect to different articles in Wikipedia (e.g., the mention ‘Willamette’ may refer to the Willamette river or the Willamette Week newspaper).

Multiple works targeting WR and NED rely on graph-based methods. Some of these efforts focused on the Wikipedia hyperlink graph utilizing the large coverage of the corpus. Milne and Witten [106] measured the relatedness between two Wikipedia articles by looking at the number of incoming edges that they share. (The method focused only on direct neighbors.) The approach is based on the idea that related articles are likely to be referenced by the same set of pages. Jeh and Widom [62] presented SimRank, a graph algorithm that utilized nodes beyond direct neighbors. SimRank recursively computed relatedness between two nodes using the relatedness of their neighbors. The base case is the relatedness of a node and itself, which is given a maximum score. Ponzetto and Strube [86] tested multiple methods for WR on the Wikipedia category structure and reported that a normalized path-length measure (from Leacock and Chodorow [69]) performs the best. The measure is based on computing the shortest path between categories related to the compared articles in addition to considering the depth of the category structure. For NED, Guo et al. [54] disambiguated a target mention by counting the direct links between the articles connected to the target mention and articles connected to other mentions in the context.

For methods based on the Wikipedia hyperlink graph, Agirre et al. [11] reported the state-of-the-art performance in both WR and NED. Their methods used the PPR algorithm on a graph of Wikipedia that discarded categories and non-reciprocal links. For WR, the method expected two mentions as input, with each mention associated with multiple Wikipedia articles (possible meanings of the mention). The PPR algorithm is computed for each mention using its associated Wikipedia articles as a base set (teleport vector). Then, the relatedness between the two mentions is computed as the cosine similarity of their PageRank vectors. For NED, the system takes as an input a target mention with its context (a 101-word excerpt with the target mention in the middle). The system finds Wikipedia articles related to the target mention and all other mentions in the context (each mention may connect to multiple articles). The Wikipedia articles are used as a base set for computing the PPR algorithm. The Wikipedia article with the highest PageRank score is suggested as the entity most relevant to the target mention. We further discuss the configuration of the algorithm and the graph used in Agirre et al. in the next section, when we talk about related work that used the PPR algorithm.

The techniques used in NED and WR projects can be useful in computing MDN D2C queries, such as the use of the PPR algorithm and the Wikipedia hyperlink graph. However, there are some differences between these two applications and MDN D2C queries that require separate research to focus on MDN-specific settings. For example, one may suggest the use of word relatedness to compute MDN D2C queries. A possible approach is to compute the relatedness between a selected element in a diagram and every Wikipedia page, then rank those pages by their relatedness score. To use the context provided in the diagram, relatedness between each diagram element and every Wikipedia page has to be computed too. Such an approach is

clearly expensive for both experiments and production. Considering the methods presented by Agirre et al., the approach used with NED is much cheaper and closer to MDN as we explain next.

Compared to WR, NED is much closer to MDN D2C queries given that NED includes context (text that include mentions related to the target mention), which is similar to the context provided by the diagram (elements related to the selected element). However, diagrams organize elements in different structures compared to the mentions surrounding a target mention (the structure of the mentions is only similar to path graphs discussed in Section 3.3). Furthermore, NED links each mention in the context to Wikipedia articles that represent all possible meanings of the mention including irrelevant ones. In MDN however we expect manual connections to link elements in the diagram to only the related pages in Wikipedia.

9.6 Superimposed Information

The idea of superimposing diagrams builds on the work of superimposed information (SI) [74]. Maier and Delcambre [74] described SI as data placed over existing data; the top level is called the *superimposed layer* and the lower level is called the *base layer*. Examples of SI in print and digital formats are numerous including annotations in books, bookmarks in web browsers, and organized lists of links to web pages [74]. As an example of SI, Maier and Delcambre introduced structured maps [40], which can include a superimposed layer with information modeled using the entity-relationship model. For instance, in the art domain an SI layer can include artist and painting entities; these entities in the SI layer can link to artist’s information and painting’s picture and reviews in an art collection.

In the area of digital libraries, the following work superimposed specific maps or diagrams over information resources. Cañas et al. [31] presented CMAPTOOLS, a

tool for creating concept maps. A concept map [83] is a simple view of concepts in a domain represented as rectangles or ovals with relationships between concepts represented using arcs. CMAPTOOLS allows authors to link concepts in a concept map to related information resources as well as other concept maps. Butcher [29] used strand maps [13], representing ordering of educational learning objectives, as an interface to access digital library documents. Bowers et al. [26] superimposed Entity-Relationship diagrams over a collection of documents. In MDN, we do not rely on a specific type of diagram or map. We also support additional query features, such as C2D and D2D queries. Furthermore, we access and rank collection documents not directly connected to the diagrams or maps.

Another form of superimposed information that provides navigational support is *guided tours* [101]. In a guided tour, a user or a content author can create a sequence of multiple stops (i.e., URLs to web pages or collection documents) to guide future users' navigation. The tour can also include annotations to introduce the resources in the tour or provide interpretations. Trigg [101] used guided tours in NoteCards, a system that supported formulation and management of information. Furuta et al. [48] used guided tours (they called them Walden's paths) to provide guided navigation in the web for grade-school students. Yuan and White [109] presented *trailblazing*, a process that allows users to create a sequence of URLs to guide novice users in web search. Fernando et al. [47] studied the process of creating guided tours for a cultural heritage collection.

In the guided-tours projects discussed above, the research focused on the creation and usage of guided tours. In MDN, and specifically in this dissertation, we focus on the incorporation of already-created diagrams in navigation (i.e., diagrams that were not created with the specific intent to help users in navigation). In addition, guided

tours provided a sorted list of resources with a specific theme or story (e.g., finding causes and treatment of a disease or describing a cultural artifact). On the other hand, we use diagrams in MDN to provide an overall overview of a domain where users can start anywhere in the diagram and explore in any direction. Furthermore, in MDN we introduce the D2D and C2D query types as well as the ranking and retrieval of related resources in the collection.

One aspect that was studied in guided-tours research projects is the types of resources selected by users and the nature of the annotations or descriptions included with the tours [47, 48, 109]. Yuan and White [109] studied the differences between experts and novices in creating trails (tours) in the medical domain. The researchers observed that trails created by domain experts were better. While novices focused only on the details, experts provided background before going into the specifics. Fernando et al. [47] noticed that paths created by users lacked background information. Therefore, they compared two automatic methods to add background to users' tours. The best method was the extraction of the first n sentences from a Wikipedia article that best represents the tour topic. Furuta et al. [48] found that some paths contained richer annotations than the others. Teachers, for example, included questions in their paths to help students learn the subject. Given that we focused on available diagrams and we mapped diagrams to Wikipedia ourselves, we have not studied these aspects of using diagrams in navigation. These findings can guide future work for MDN, such as studying expert involvement in creating and mapping diagrams or enabling users to annotate diagrams to provide additional background.

9.7 The Personalized PageRank Algorithm

The Personalized PageRank (PPR) algorithm has been used in different applications that utilized different graphs. In this section, we focus on related work that

used PPR in applications similar to MDN, where given some nodes of interest in a graph, the goal is to find related nodes. Since we reported a preference for a high damping factor for MDN D2C queries, which was different from related work using a similar graph, we focus on how this parameter was tuned in other research projects. We also describe different graph configurations used in these projects.

The closest to our work are projects that used PPR with the Wikipedia graph. Agirre et al. [11] used PPR for word relatedness and named entity disambiguation (described in Section 9.5). They reported their best performance with a Wikipedia graph that included only articles and reciprocal links (categories and non-reciprocal links are removed). In MDN, we showed that keeping low weight non-reciprocal links is better than removing them completely. Note that we used a small graph in our experiments that included a subset of the Wikipedia nodes and links. Therefore, our findings only provide indications that can be studied in more detail for other applications that use the full graph of Wikipedia.

In a similar project, Yeh et al. [108] used PPR on the Wikipedia graph for measuring word relatedness and text similarity. Their best results were achieved using an approach that builds on Explicit Semantic Analysis (ESA) [49]. ESA models an input text in a vector space where each dimension represents a Wikipedia article. To map an input text to a weighted vector, ESA computes the similarity between the text and each Wikipedia article. Given that ESA is based only on text similarity, Yeh et al. used the ESA vector as a teleport vector for PPR to benefit from the Wikipedia graph. Relatedness between two texts is simply computed as the cosine similarity of their PageRank vectors.

In a tuning session for the damping factor, Yeh et al. showed that increasing the damping factor had a positive impact on performance for the overall graph. However,

when using a pruned Wikipedia graph, the increase of the damping factor had a negative impact on performance. The pruned graph and a PPR with a low damping factor (around 0.15) achieved the best performance. The study concluded that the original Wikipedia graph was noisy. Therefore, allowing the random surfer to visit Wikipedia pages far from the teleport vector is better for the application when using a cleaned (pruned) version of the Wikipedia graph. We do not provide details of the graph pruning method given that the approach of focusing on reciprocal links presented by Agirre et al. (described earlier) provided better performance.

The insights provided by Yeh et al. and Agirre et al. indicate that, unlike MDN D2C queries, where best performance is achieved with high damping factor, text similarity and word relatedness perform the best with low damping factor values. Unlike the mentioned NLP applications, it seems that diagrams provide richer contexts (base set or teleport vector) that positively influence the performance (i.e., the random surfer should teleport to those contexts more frequently).

Lu et al. [73] built user profiles to rank tweets in Twitter. Part of a user's profile is simply a weighted vector of Wikipedia articles representing the user's interest. Similar to Yeh et al., the vector is built using ESA to map tweets in a user timeline to Wikipedia concepts. The ESA vector is then used as a teleport vector for running PPR. To rank tweets for a specific user, a weighted vector is created for each tweet (using ESA also). The tweets are then ranked by the score of the cosine similarity between their vectors and the user-profile vector. The approach used a high damping factor $d=0.85$, which was selected based on experience. The Wikipedia graph used was not processed or cleaned. The approach also considered similarity of a user's network (e.g., followed and following users) and users who interacted with each tweet.

The previous application showed similarity to MDN in terms of using a high

damping factor value. Using the Wikipedia graph without any pruning or cleaning may have influenced this choice (i.e., keep the random surfer close to the base set to avoid the noise in the graph). However, a bigger factor may have been the richness of the base set used in PPR. A user profile was based on 1000 tweets, which increases the chance that most concepts the user is interested in are mentioned in these tweets. Therefore, the Wikipedia vector already represents most user interests. Allowing the random surfer to travel far from this rich vector was not useful.

For collections other than Wikipedia, Balmin et al. [18] provided some insight regarding the tuning of the damping factor. They presented ObjectRank, a ranking algorithm based on PPR to enhance keyword search in a collection or a database modeled as a graph (e.g., the computer science bibliography DBLP). For a single-keyword search, the nodes containing the keyword are used as a base set for the PPR computation to find other highly related nodes that do not explicitly contain the keyword. The researchers used a weighted directed graph that matches a predefined schema, which can be configured to capture the semantics of the collection. For example, in one DBLP schema, 70% of the authority of a paper is distributed among the papers it cites while 10% is transferred to the paper’s venue and the remaining 20% is distributed among the paper’s authors. The researchers experimented with different damping factors but did not report a specific setting that delivers consistent performance. They mentioned that for keywords related to small (more specific) domains, a higher damping factor is preferred. Even though we used a different graph in MDN, the authors’ observation matches our findings to some extent. Related pages in our D2C-query experiments were mostly close to the diagram used (mostly in L2), leading PPR to perform better with high damping factor values

9.8 The Energy-Spreading Algorithm

The energy-spreading (or activation-spreading [35]) technique has been used in different applications in the area of information retrieval. Demovic et al. [41] used the technique for movie recommendation. They used a graph that included different types of nodes, such as movies and actors, where energy is spread starting from nodes of interest to the user. Movies that receive the largest amount of energy are recommended. Rodriguez and Bollen [90] used energy spreading in a co-authorship graph to determine the best peer-reviewers for submissions to a conference. The algorithm starts from author nodes whose papers are referenced by a conference submission. The amount of energy sent from an author to another depends on the number of papers the two authors have co-authored (represented as the edge weight). In the area of web-search personalization, Sieg et al. [94] used energy spreading in ontological user profiles to identify topics of interest to the user. Using a term vector that represents the current interests of a user (e.g., terms extracted from user's search queries), initial concepts in the user's ontological profile are identified and scored. Scores of those concepts and other related ones are updated using the energy-spreading technique. The user profiles are then used to personalize web search results.

The main difference between the energy-spreading technique used in the GES algorithm and the research projects listed above is the sensitivity to node degrees. In the described projects, energy is sent by splitting a node's energy among its neighboring nodes. Nodes also aggregate the energy they receive. In both sending and receiving, node degree influences the amount of energy sent or received. High-degree nodes, for example, can accumulate more energy. In GES, we limit the effect of node degrees by sending the same amount of energy to all neighboring nodes (instead of splitting energy). We also use the SAS function that considers only the top N received

energies reducing the advantage of having a high degree.

The energy-spreading technique was also used with the graph of Wikipedia. Kapani pathi et al. [66] used the energy-spreading technique in the Wikipedia category hierarchy to identify topics in Twitter posts (i.e., find user interests). Starting from articles related to a user's Twitter posts, energy is sent upward in the Wikipedia category hierarchy to find the most relevant categories (topics). Instead of splitting energy, a source node (article or subcategory) sends its energy upward to all target nodes (parent categories). Target nodes sum the received energies, leading to large categories having a large advantage. To mitigate this problem, the researchers used the fact that large categories mostly exist in the middle levels of the Wikipedia category hierarchy. Therefore, the activation function applied a penalty based on the level of each category. This method is not applicable to MDN, since we do not have classes of nodes based on node degrees.

A similar work that used energy spreading in the Wikipedia category hierarchy is topic identification in documents by Schönhofen [93]. He mapped words in a document to related articles in Wikipedia. Similar to the previous work, the articles send their energy (without splitting) to related categories. Also, categories sum the received energies leading to the dominance of categories related to large domains such as Films and Actors. To cope with this problem, the approach used a text-based measure to penalize categories whose vocabulary is large (where vocabulary is the number of distinct words in the titles of articles connected to the category). Such a technique is not applicable to MDN, given that so far we only use the hyperlink structure of Wikipedia, as well as that we work with both categories and articles.

9.9 Summary

In this chapter, we described various approaches and solutions for exploratory-search scenarios. We discussed related work that used the Wikipedia collection, particularly those that utilized the Wikipedia hyperlink graph. We also discussed related work that used techniques we used in MDN, which include superimposed information and both the PPR and energy-spreading algorithms.

To the best of our knowledge, we are the first to study the use of diagrams in providing exploratory search interfaces for collections (especially Wikipedia). We are also the first to study the use of the PPR and energy-spreading algorithms on graphs that include diagram elements and Wikipedia pages.

10 Conclusions, General Observations, and Future Work

In this chapter, we summarize and conclude our research as well as present possible future-work directions. In Section 10.1, we start by presenting an overall overview of the problem and the suggested solution. We present results and findings that are specific to MDN. In Section 10.2, we present general findings and observations that can be utilized in other applications. Finally, in Section 10.3, we present directions for future work.

10.1 Conclusions

In large document collections, traditional information retrieval tools such as keyword searching and hyperlink browsing may not be sufficient for some information-seeking scenarios. For example, domain novices may start their information seeking with the disadvantage of not knowing which concepts (or documents) exist in the domain, which concepts fulfil their information needs, and which concepts are interesting for general exploration. Therefore, those novices struggle to find the appropriate search keywords or choose the right hyperlink to click. This problem, called Exploratory Search [75], attracted Information Retrieval (IR) and Computer-Human Interaction (CHI) research communities to build new styles of interactive search interfaces.

We presented our solution, Multiple Diagram Navigation (MDN), that provides support for domain novices by presenting multiple graphical overviews of a collection

of documents. While related work uses induced visualizations to provide overviews of collections, these approaches support limited overview options restricted by the available visualization tools. MDN, on the other hand, relies on superimposed diagrams, which allows content authors to attach any type of overview (diagram) to their collections. End users can navigate among diagrams and collection documents using the three types of MDN navigational queries: Diagram to Content (D2C), Content to Diagram (C2D), and Diagram to Diagram (D2D).

We started this research by presenting Basic MDN (BMDN), which is a simple version of the system where diagrams are connected to a limited number of collection documents. As an interactive navigational user interface, it was important to evaluate how effective the interface is in exploratory search scenarios. We conducted a user study where subjects perceived MDN as useful and easy to use. They also endorsed the effectiveness of the three types of queries in MDN.

Encouraged by the results of the user study, we decided to work on the biggest limitation BMDN had, which is accessing only the subset of a collection documents that are manually connected to the diagrams. We extended MDN to access an entire collection¹ by using both internal collection connections (e.g., Wikipedia hyperlinks) and internal diagram connections. We called the new system, Extended MDN (EMDN). We started working on EMDN by analysing the structure of 37 diagrams from different domains. We introduced two diagram models: the Items and Attributes (I&A) Model and the Universal Model. Both diagram models are based on the graph model to represent diagram concepts and internal connections.

In EMDN, we mostly focused on studying D2C-query ranking in the Wikipedia collection. Wikipedia is a likely destination for exploratory searchers. In addition,

¹We assume every document in the collection has a path to at least one diagram element.

we benefited from a large body of research on ranking Wikipedia pages for various applications and using different algorithms. The D2C-query type is the most challenging in MDN since we have to rank a large collection of pages compared to only ranking elements in a diagram such as in C2D queries.

We studied the use of two graph ranking algorithms, Personalized PageRank (PPR) and Greedy Energy Spreading (GES). Given the novelty of our application, we wanted to find out the optimal settings for PPR. For example, compared to other applications of PPR, MDN provided rich and manually prepared context (diagrams). We also hypothesized that PPR may be influenced heavily by the structure of the Wikipedia collection, which might override the influence of the diagrams used. Therefore, we designed GES to maximize the influence of diagrams on the ranking process. Our goal was to find the best algorithm and settings that yield the best ranking for D2C queries (and later for C2D queries).

We conducted three investigations regarding ranking in EMDN. In the first investigation, we studied D2C-query ranking in I&A-Model diagrams with manually labeled top-ten pages returned by the different settings of both PPR and GES. In the second investigation, we also studied D2C-query ranking, but in Universal-Model diagrams, where we turned to an automatic evaluation method. The last investigation focused on studying different approaches to rank diagram elements in C2D queries. We used Universal-Model diagrams for this task.

One of the important findings that our experiments revealed is the positive influence diagrams have on the D2C-ranking process. Assuming a user is interested in a diagram concept (element) and also the viewpoint or the aspect that the diagram presents, the diagram itself provides an added value that helps in finding related pages in the collection. This finding was supported by the first two investigations

and with evidence from both PPR and GES. Increasing diagram influence on ranking, which also decreased the Wikipedia influence, increased the performance of MDN D2C ranking.

If one diagram changes the ranking in a positive way, what happens if we have two diagrams that cover the same exact domain? That was a question that we could study with I&A-Model diagrams, given the ability of those diagrams to easily present different aspects of the same domain. When studying ten pairs of diagrams in the I&A-Model investigation, we found out that those pairs provided different rankings. As a result, two D2C queries with the same selected element but issued from different diagrams would return mostly different pages in the top ten.

When comparing PPR and GES in D2C and C2D queries, we found out that GES was slightly better in the first investigation – it increased the relevance of about 1.5 pages in the top ten. With Universal-Model diagrams, GES outperformed PPR when including pages connected to the diagrams in the ranked lists. However, when removing those pages, the two algorithms showed similar performance. In C2D-query ranking, GES was slightly better when considering the ranking of all diagram elements. However, when we focused on the ranking of the most related element in the diagram, GES outperformed PPR. We conclude that GES is generally slightly better than PPR for MDN D2C and C2D queries. However, in some cases, such as with large Universal-Model diagrams that cover most of their respective domains, PPR can provide a useful ranking of Wikipedia pages. We think that further MDN research and implementations can still use both algorithms as ranking-engine choices to bring more insight.

Our last investigation regarding C2D queries revealed that it is possible to use D2C-ranked lists to compute the rankings for C2D queries. A D2C-based approach

that used GES D2C-ranked lists was better overall than a native one that used the PPR algorithm and started from a C2D selected page. Even though the native approach was slightly better when considering the ranking of all diagram elements, the D2C-based approach outperformed the native one by a large margin when we considered only the rank of the most related element in the diagram. What this finding means for MDN is that we can run D2C queries offline and use their ranked lists to compute both D2C and C2D queries online. This approach enables MDN to provide fast response times in query processing, which is necessary for an interactive user interface.

10.2 General Observations

Some of the findings of our research apply beyond MDN. Therefore, we cover those findings in this section and discuss their effect on MDN. The effect on MDN may also apply for other applications such as those using the Wikipedia graph or the Personalized PageRank.

One of the factors that we focused on during our tuning of PPR and GES is the treatment of reciprocal and non-reciprocal links in Wikipedia. Agirre et al. [11] introduced the technique of using only reciprocal links in Wikipedia. Wikipedia articles are heavily hyperlinked. Therefore, hyperlinks do not always represent strong relatedness between two articles. However, a strong indication that two articles are indeed related is when they link to each other (i.e., they have a reciprocal link).

We did observe the importance of reciprocal links in our D2C-ranking experiments. However, we chose a different approach that retained non-reciprocal links but gave them lower weights. This approach achieved better performance than keeping only reciprocal links or assigning the same weight for both types of links in two D2C-query investigations. Furthermore, this behavior was observed in both PPR and GES. For

example, in D2C ranking with Universal-Model diagrams, both algorithms worked the best when giving non-reciprocal links about half the weight of reciprocal links. We think that other applications that use the graph of Wikipedia should examine this parameter. A related parameter that we studied is the inclusion of categories and article-template hyperlinks. Agirre et al. [11] reported their best performance when they excluded categories from the Wikipedia graph. In terms of templates, we noticed how templates can reverse the influence of a diagram on the ranking of D2C queries in our investigation regarding I&A-Model diagrams. Therefore, we studied the effect of including or excluding categories and article-template hyperlinks in more detail when we worked with Universal-Model diagrams. Our results indicate a considerable influence of these parameters on both GES and PPR.

Similar to the findings of Agirre et al. [11], our results show that PPR did not work well when including categories in the graph. A possible reason is that categories tend to have large degrees. In addition, all category hyperlinks are reciprocal (adding a link to a category in an article automatically adds the article in that category). Therefore, as PPR is sensitive to node degrees, categories may have accumulated larger scores that did not represent their true relevance to a diagram concept selected in a D2C query. However, adding article-template hyperlinks seemed to help PPR mitigate the existence of categories. In fact, the best graph configuration for PPR was the one that included categories and article-template hyperlinks.

For GES, the influence of categories and article-template hyperlinks was the opposite of that in PPR. GES performed better when including categories. The lower sensitivity GES has to node degrees (compared to PPR) may have yielded this outcome. On the other hand, GES performed the worst when including article-template hyperlinks.

Overall, we think that the parameter of including or excluding categories and article-template hyperlinks are influential for algorithms that use the Wikipedia graph such as measuring word relatedness in Natural Language Processing (NLP) and generating recommendations for social networks. Even though we did not work with the full Wikipedia graph, we think testing this parameter for future research is worthwhile.

One of the important observations we had in our experiments with the PPR algorithm is the high damping factor values (0.65 to 0.85) that the algorithm preferred in MDN D2C queries. This observation was confirmed in two investigations regarding D2C queries. The value 0.15 for damping factor was the one used in the original PageRank algorithm [27]. It was also the value used in related work [11, 108] that utilized the Wikipedia graph. However, in MDN, it seems that the richness of the context (diagrams) was an important reason for increasing the damping factor. Given that the diagrams connected to a large number of related pages, other related pages were also likely to be close (most related pages had an at least one two-edge path to a diagram element). However, we noticed that in this case PPR tended to rank pages in the context (pages connected to the diagram in MDN) higher than other pages in the collection. PPR did not seem to utilize the structure in diagrams that gave context pages shorter distances to the selected diagram element. We conclude that richness of diagrams, which we used as context for PPR, can decrease the distance to target related pages. In this case, high damping factor values may yield the best performance for PPR. However, the rank of pages in the context should be examined carefully. We note that in the C2D experiment where we ran PPR from a Wikipedia page with no diagram context, the algorithm performed the best with damping factor equal to 0.15.

The last observation we cover in this section is the high greediness we ended up using in GES. While the method we used to compute PPR processed the whole graph, we were able to achieve a similar or better ranking in GES by propagating energy from a relatively small number of Wikipedia pages (e.g., only 21 pages on average in Universal-Model diagrams). We showed that propagating energy from more pages either does not increase or even hurts the performance. We think that this greedy technique can be investigated in other energy-spreading (and possibly other) ranking algorithms, especially those that work on the Wikipedia graph.

10.3 Future Work

In this section, we discuss possible directions of future work. Given the novelty of MDN as a navigation system, there are various features that need to be studied and possibly added to the system. We discuss some of those features that we think are worth investigating.

10.3.1 Extending Diagram Navigation to Collections Other than Wikipedia

One direction of future work is to study the process of extending the use of MDN to other collections such as class material, digital libraries, and online shopping websites. One way to achieve this goal is by manually connecting a collection to MDN diagrams. In this case, further studies are needed to tune MDN ranking algorithms for those collections.

Another alternative that allows MDN to access different collections is by mapping these collections to Wikipedia or a related knowledge base such as DBpedia (i.e., using Wikipedia or a knowledge base as an intermediate layer between diagrams and other collections). An important step towards this goal is finding appropriate and efficient algorithms to map such collections to Wikipedia. Similar mapping methods,

such as entity linking and Wikification [78] and Explicit Semantic Analysis [49] can be studied and possibly extended to achieve this goal. One advantage of using Wikipedia as a middle layer is to allow MDN to access collections that have limited (or lack) internal connections such as for a repository of class lecture notes or news articles collected from different sources. Also, we can benefit from the knowledge we collected regarding ranking in the Wikipedia collection.

10.3.2 Comparing Diagram Navigation with Other Exploratory Search Tools.

Given the number of available choices for exploratory search tools, a possible re-search direction is to study in what scenarios each tool is more helpful. For example, a user study can compare different exploratory search tasks and suggest some use cases for different tools, such as MDN, a visualization such as a tag cloud, and a recommendation engine. It would be interesting to know which tools different users prefer to use. For example, do users prefer a dynamic overview created by a visualization tool or a static one in MDN (possibly created by a domain expert and rated by other users)? Also, do users prefer a ranked list provided by a recommendation engine based on their navigation history or a one based on their selection on a diagram?

10.3.3 Automating the Task of Connecting Diagrams to Wikipedia.

The current scenario for MDN starts with manually creating connections between diagram elements (concepts) and related Wikipedia pages and also between elements in a diagram. A future work direction can investigate whether this manual work can be automated (or semi-automated) using tools such as Optical Character Recognition (OCR) and Named Entity Disambiguation (NED). OCR can be used to recognize the text in diagrams saved in image formats. We do not know whether OCR tools can (or

can be extended) to also recognize connections (lines) between concepts in diagrams. NED tools are used to disambiguate entities in document text and connect them to related articles in Wikipedia. One approach can use NED to connect concepts (entities) mentioned in diagrams' text to the right articles in Wikipedia. A major difference between document text and diagrams is that diagram concepts are usually composed of short sentences or single words. In addition, while NED uses adjacent words as context for a target entity in a given text, it can use close and connected concepts in a diagram.

10.3.4 Multiple Selected Elements or Pages in D2C and C2D Queries

In this study, we only studied single-element D2C queries and also single-page C2D queries. A possible feature to add to MDN is supporting D2C queries that include multiple diagram elements and C2D queries with multiple Wikipedia pages. One question that needs to be answered is whether we can compute multi-node queries using single-node queries. For example, a D2C query that includes two diagram elements can average the ranks or scores of Wikipedia pages when each element is selected alone in a D2C query. We need to know how close this approach is to computing two-element query from scratch (such as by propagating energy from the two elements together in GES or including the two elements in the base set in PPR).

10.3.5 Studying C2D Queries in More Detail

As we mentioned in Chapter 8, we only compared approaches to compute the ranking of C2D queries. An extension to that line of work can use a user-centered approach to evaluate the accuracy of C2D queries in more detail. In addition, an important factor that can be studied is how C2D ranking should be displayed. For example, we showed that highlighting the top few elements in the ranking is better

supported with the current ranking approach. We need to know how many elements that we need to highlight. If we choose to show a heat map instead (proportionally highlight diagram elements), we need to know how diagram-element colors or different brightness levels should be used.

10.3.6 Measuring the Quality of Diagrams for MDN Ranking

In Chapters 7 and 8, we mentioned how having a good context (a number of semantically related elements close to the selected element) in D2C queries can help in increasing the performance of D2C ranking. We also talked about diagrams that contain different clusters of elements and how that can help increase the performance of C2D queries. We think that such properties of diagrams can be measured. As a result, a user (content author) can be notified if, for example, a diagram requires more connections to Wikipedia to help in the ranking process.

10.3.7 Computing the Ranking of D2D Queries

When we studied ranking for EMDN, we chose to focus on D2C and C2D ranking. One direction of future work is to study ranking for D2D queries. In Chapter 8, we showed that the C2D-query ranking can actually be computed using D2C-query ranked lists. Similarly, we think that the ranking of D2D queries may be computed using a composition of D2C and C2D queries. For example, given a selected element e in a diagram $D1$ we find the most related Wikipedia page (rank #1) when e is selected in a D2C query. Then, we issue a C2D query from this page to another diagram $D2$. The approach can also use the top k pages ranked in the D2C query and then issue multiple C2D queries to diagram $D2$. However, in this case, we need to aggregate the results of the C2D queries. One way to study the performance of the composition of D2C and C2D queries is to compare the result of this approach to manual connections

of related elements in pairs of diagrams (e.g., manually connecting related elements in the diagrams $D1$ and $D2$).

Bibliography

- [1] Car makes, www.carmagazine.co.uk
- [2] Civil war timeline, <https://pistonclasico.com/wall/civil-war-project-civil-war-timeline-info>
- [3] Delicious coffee, <https://hubpages.com/food/coffee-maker-buying-guide#>
- [4] Periodic table, https://en.wikipedia.org/wiki/Periodic_table
- [5] World history chart, <https://www.hyperhistory.com/>
- [6] Facetedpedia (2012), <https://idir.uta.edu/facetedpedia/>
- [7] Data platforms map (2016), <https://451research.com/state-of-the-database-landscape>
- [8] New registered users, monthly (2018), <https://stats.wikimedia.org/v2/#/en.wikipedia.org/contributing/new-registered-users/normal|bar|2-Year|~total>
- [9] Total page views, monthly (2018), <https://stats.wikimedia.org/v2/#/en.wikipedia.org/reading/total-page-views/normal|bar|2-Year|~total>
- [10] Wikipedia:statistics (2018), <https://en.wikipedia.org/wiki/Wikipedia:Statistics>
- [11] Agirre, E., Barrena, A., Soroa, A.: Studying the Wikipedia hyperlink graph for relatedness and disambiguation. arXiv preprint arXiv:1503.01655 (2015)
- [12] Alonso, O., Baeza-Yates, R., Gertz, M.: Exploratory search using timelines. In: SIGCHI 2007 Workshop on Exploratory Search and HCI Workshop. pp. 1–4. No. 1 (2007)
- [13] American Association for the Advancement of Science.: Atlas of Science Literacy: Project 2061 (2001)
- [14] Atzori, M., Zaniolo, C.: Swipe: searching Wikipedia by example. In: Proceedings of the 21st International Conference on World Wide Web. pp. 309–312. ACM (2012)

- [15] Avrachenkov, K., Litvak, N., Nemirovsky, D., Osipova, N.: Monte Carlo methods in PageRank computation: When one iteration is sufficient. *SIAM Journal on Numerical Analysis* 45(2), 890–904 (2007)
- [16] Bahmani, B., Chowdhury, A., Goel, A.: Fast incremental and personalized PageRank. *Proceedings of the VLDB Endowment* 4(3), 173–184 (2010)
- [17] Bairi, R.B., Carman, M., Ramakrishnan, G.: On the evolution of Wikipedia: Dynamics of categories and articles. In: *Ninth International AAAI Conference on Web and Social Media* (2015)
- [18] Balmin, A., Hristidis, V., Papakonstantinou, Y.: Objectrank: Authority-based keyword search in databases. In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases-Volume 30*. pp. 564–575. *VLDB Endowment* (2004)
- [19] Balog, K., Serdyukov, P., De Vries, A.P.: Overview of the TREC 2010 Entity track. Tech. rep., Norwegian University of Science and Technology Trondheim (2010)
- [20] Baluja, S., Seth, R., Sivakumar, D., Jing, Y., Yagnik, J., Kumar, S., Ravichandran, D., Aly, M.: Video suggestion and discovery for YouTube: taking random walks through the view graph. In: *Proceedings of the 17th International Conference on World Wide Web*. pp. 895–904. *ACM* (2008)
- [21] Bekkerman, R., Donin, O.: Visualizing Wikipedia for interactive exploration. In: *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics*. pp. 11–17. *ACM* (2017)
- [22] Benotman, H.: Drupal module for multiple diagram navigation (2015), <https://github.com/benotman/mdn>
- [23] Biuk-Aghai, R.P., Pang, C.I., Si, Y.W.: Visualizing large-scale human collaboration in Wikipedia. *Future Generation Computer Systems* 31, 120–133 (2014)
- [24] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web* 7(3), 154–165 (2009)
- [25] Blanco, R., Cambazoglu, B.B., Mika, P., Torzec, N.: Entity recommendations in web search. In: *International Semantic Web Conference*. pp. 33–48. *Springer* (2013)
- [26] Bowers, S., Delcambre, L., Maier, D.: Superimposed schematics: Introducing er structure for in-situ information selections. In: *International Conference on Conceptual Modeling*. pp. 90–104. *Springer* (2002)
- [27] Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems* 30(1-7), 107–117 (1998)

- [28] Brinkley, M.: Anatomy of today's threats and how to stop them—an architect's view of threats and McAfee protection (2013), https://intel.activeevents.com/sf13/connect/fileDownload/session/6B15D993C3C1AA0DF1507C65FC7110ED/SF13_SECS001_100.pdf
- [29] Butcher, K., Bhushan, S., Sumner, T.: Multimedia displays for conceptual discovery: information seeking with strand maps. *Multimedia Systems* 11(3), 236–248 (2006), <http://dx.doi.org/10.1007/s00530-005-0004-y>
- [30] CableTV.com: Remote control: Who owns what you watch on TV (2011), <https://www.cabletv.com/blog/infographic-who-owns-what-you-watch-on-tv/>
- [31] Cañas, A.J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., Gomez, G., Arroyo, M., Carvajal, R.: Cmaptools: A knowledge modeling and sharing environment. In: *Concept maps: Theory, Methodology, Technology. Proceedings of the First International Conference on Concept Mapping*. vol. 1, pp. 125–133 (2004)
- [32] Cantus: South American community of nations, https://upload.wikimedia.org/wikipedia/commons/e/e0/UNESCO_World_Heritage.svg
- [33] Caryl-Sue: September 9, 2007: Wikipedia becomes largest encyclopedia (2013), <https://www.nationalgeographic.org/thisday/sep9/wikipedia-becomes-largest-encyclopedia/>
- [34] Chandrasekaran, S.: Becoming a data scientist – curriculum via Metromap (2013), <http://nirvacana.com/thoughts/2013/07/08/becoming-a-data-scientist/>
- [35] Crestani, F.: Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review* 11(6), 453–482 (1997)
- [36] Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Q.* 13(3), 319–340 (Sep 1989), <http://dx.doi.org/10.2307/249008>
- [37] Davis, M.: Culinary herbs, http://botanicalposters.com/index.php?main_page=index&cPath=
- [38] De Vries, A., Thom, J., Vercoustre, A., Craswell, N., Lalmas, M.: Inex 2007 entity ranking track guidelines. In: *INEX 2007 Workshop Pre-Proceedings*. vol. 10 (2007)
- [39] Dear Doctor, I.: Minerals, <https://www.deardocor.com/articles/vitamins-and-dietary-supplements/>
- [40] Delcambre, L.M., Maier, D., Reddy, R., Anderson, L.: Structured maps: Modeling explicit semantics over a universe of information. *International Journal on Digital Libraries* 1(1), 20–35 (1997)

- [41] Demovic, L., Fritscher, E., Kriz, J., Kuzmik, O., Proksa, O., Vandlikova, D., Zelenik, D., Bielikova, M.: Movie recommendation based on graph traversal algorithms. In: Database and Expert Systems Applications (DEXA), 2013 24th International Workshop on. pp. 152–156. IEEE (2013)
- [42] Dork, M., Carpendale, S., Collins, C., Williamson, C.: Visgets: Coordinated visualizations for web-based information exploration and discovery. *IEEE Transactions on Visualization and Computer Graphics* 14(6), 1205–1212 (2008)
- [43] elenabsl: vitamins, https://www.123rf.com/photo_58290076_stock-vector-vitamin-food-sources-and-functions-rainbow-wheel-chart-with-food-icons-healthy-eating-and-healthcare.html?fromid=aG5TbjBqMXpWVFZ30G9Cd0xVZHNJdz09
- [44] Encyclopædia Britannica, I.: American civil war battles., <http://kids.britannica.com/elementary/art-87023/Most-of-the-major-battles-of-the-American-Civil-War>
- [45] EuroTalk.com: World language families, <https://eurotalk.com/blog/tag/language-family/>
- [46] exploringnature.org: Animal kingdom, www.exploringnature.org
- [47] Fernando, S., Goodale, P., Clough, P., Stevenson, M., Hall, M., Agirre, E.: Generating paths through cultural heritage collections. In: Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities. pp. 1–10 (2013)
- [48] Furuta, R., Shipman III, F.M., Marshall, C.C., Brenner, D., Hsieh, H.w.: Hypertext paths and the World-Wide Web: experiences with Walden’s paths. In: Proceedings of the Eighth ACM Conference on Hypertext. pp. 167–176. ACM (1997)
- [49] Gabrilovich, E., Markovitch, S., et al.: Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In: International Joint Conferences on Artificial Intelligence. vol. 7, pp. 1606–1611 (2007)
- [50] Gallardo, K.: Landforms, <https://www.thinglink.com/scene/709032717307609088>
- [51] Gouws, S., Rooyen, G., Engelbrecht, H.A.: Measuring conceptual similarity by spreading activation over Wikipedia’s hyperlink structure. In: Proceedings of the 2nd Workshop on The People’s Web Meets NLP: Collaboratively Constructed Semantic Resources. pp. 46–54 (2010)
- [52] Greggs, F.: Pathways in nursing, <https://visual.ly/community/infographic/health/pathways-nursing>
- [53] Group, W.S.W.: Scalable vector graphics (SVG) 2 (2018), <https://www.w3.org/TR/SVG2/>

- [54] Guo, Y., Che, W., Liu, T., Li, S.: A graph-based method for entity linking. In: Proceedings of 5th International Joint Conference on Natural Language Processing. pp. 1010–1018 (2011)
- [55] Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürge, M., Düwiger, H., Scheel, U.: Faceted Wikipedia search. In: International Conference on Business Information Systems. pp. 1–11. Springer (2010)
- [56] Hall, M., Clough, P.: Exploring large digital library collections using a map-based visualisation. In: Research and Advanced Technology for Digital Libraries, pp. 216–227. Springer (2013)
- [57] Hannon, J.: The geological history of earth, <https://carnegiemuseums.org/carnegie-magazine/winter-2016/earth-in-the-age-of-humans/>
- [58] Haveliwala, T.H.: Topic-sensitive PageRank. In: Proceedings of the 11th International Conference on World Wide Web. pp. 517–526. ACM (2002)
- [59] Herzog, A., Shahmehri, N., Duma, C.: An ontology of information security. International Journal of Information Security and Privacy 1(4), 1–23 (2007)
- [60] Iván, G., Grolmusz, V.: When the web meets the cell: using personalized PageRank for analyzing protein interaction networks. Bioinformatics 27(3), 405–407 (2010)
- [61] Javed, W., Elmqvist, N.: Exploring the design space of composite visualization. In: Pacific Visualization Symposium (PacificVis), 2012 IEEE. pp. 1–8 (Feb 2012)
- [62] Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 538–543. ACM (2002)
- [63] Jiang, B.: Ranking spaces for predicting human movement in an urban environment. International Journal of Geographical Information Science 23(7), 823–837 (2009)
- [64] Joki: A handful of companies control the food industry, <http://convergencealimentaire.info/>
- [65] Kaleagasi, B.: Space exploration - timeline overview, <https://bartukaleagasi.files.wordpress.com/2017/01/space-technology-under-nasa-and-spacex.pdf>
- [66] Kapanipathi, P., Jain, P., Venkataramani, C., Sheth, A.: User interests identification on Twitter using a hierarchical knowledge base. In: European Semantic Web Conference. pp. 99–113. Springer (2014)
- [67] Lab, P.C.: Cheese wheel., www.popchartlab.com

- [68] Lab, P.C.: The delectable kaleidoscope of candy bars (2011), <https://visual.ly/community/infographic/food/delectable-kaleidoscope-candy-bars>
- [69] Leacock, C., Chodorow, M.: Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database* 49(2), 265–283 (1998)
- [70] Li, C., Yan, N., Roy, S.B., Lisham, L., Das, G.: Facetedpedia: dynamic generation of query-dependent faceted interfaces for Wikipedia. In: *Proceedings of the 19th international conference on World wide web*. pp. 651–660. ACM (2010)
- [71] LLC, B.L.R.G.: How to identify rocks, www.researchgeologist.com
- [72] Lopez, A.L.: In graphics: a world of languages - and how many speak them (2015), <https://www.scmp.com/infographics/article/1810040/infographic-world-languages>
- [73] Lu, C., Lam, W., Zhang, Y.: Twitter user modeling and tweets recommendation based on Wikipedia concept graph. In: *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*. pp. 33–38 (2012)
- [74] Maier, D., Delcambre, L.M.: Superimposed information for the internet. In: *WebDB (Informal Proceedings)*. pp. 1–9 (1999)
- [75] Marchionini, G.: Exploratory search: From finding to understanding. *Commun. ACM* 49(4), 41–46 (Apr 2006), <http://doi.acm.org/10.1145/1121949.1121979>
- [76] Massey, B., Team: Google Summer of Code wrap-up: Portland State University (2015), <https://opensource.googleblog.com/2015/10/google-summer-of-code-wrap-up-portland.html>
- [77] Mendel, C.: Portland, <http://carolmendelmaps.com/trays/index.html#Oregon>
- [78] Mihalcea, R., Csomai, A.: Wikify!: linking documents to encyclopedic knowledge. In: *Proceedings of The Sixteenth ACM Conference on Information and Knowledge Management*. pp. 233–242. ACM (2007)
- [79] Miliaraki, I., Blanco, R., Lalmas, M.: From Selena Gomez to Marlon Brando: Understanding explorative entity search. In: *Proceedings of the 24th International Conference on World Wide Web*. pp. 765–775. International World Wide Web Conferences Steering Committee (2015)
- [80] Millen, D.R., Yang, M., Whittaker, S., Feinberg, J.: Social bookmarking and exploratory search. In: *European Conferences on Computer-Supported Cooperative Work 2007*, pp. 21–40. Springer (2007)
- [81] NordNordWest: UNESCO world heritage, https://upload.wikimedia.org/wikipedia/commons/e/e0/UNESCO_World_Heritage.svg

- [82] North, C., Shneiderman, B.: Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In: Proceedings of the Working Conference on Advanced Visual Interfaces. pp. 128–135. AVI '00, ACM, New York, NY, USA (2000), <http://doi.acm.org/10.1145/345513.345282>
- [83] Novak, J.: Clarify with concept maps. *Science Teacher* 58(7), 44–49 (1991)
- [84] Page, L., Brin, S., Motwani, R., Winograd, T., et al.: The PageRank citation ranking: Bringing order to the web (1998)
- [85] Pang, C.I., Biuk-Aghai, R.P.: Wikipedia world map: method and application of map-like wiki visualization. In: Proceedings of the 7th International Symposium on Wikis and Open Collaboration. pp. 124–133. ACM (2011)
- [86] Ponzetto, S.P., Strube, M.: Knowledge derived from Wikipedia for computing semantic relatedness. *Journal of Artificial Intelligence Research* 30, 181–212 (2007)
- [87] Pound, J., Mika, P., Zaragoza, H.: Ad-hoc object retrieval in the web of data. In: Proceedings of the 19th International Conference on World Wide Web. pp. 771–780. ACM (2010)
- [88] Rank, S.: World War Two 1939-1945, <https://www.historyonthenet.com/>
- [89] Rauber, A., Bina, H.: A metaphor graphics based representation of digital libraries on the world wide web: using the Libviewer to make metadata visible. In: Tenth International Workshop on Database and Expert Systems Applications. pp. 286–290. IEEE (1999)
- [90] Rodriguez, M.A., Bollen, J.: An algorithm to determine peer-reviewers. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management. pp. 319–328. ACM (2008)
- [91] Rose, J., Aguayo, D.: Portland bridges (2014), https://www.oregonlive.com/commuting/index.ssf/2014/04/with_trimet_ready_to_name_tran.html
- [92] Sarrafzadeh, B., Vechtomova, O., Jokic, V.: Exploring knowledge graphs for exploratory search. In: Proceedings of the 5th Information Interaction in Context Symposium. pp. 135–144. ACM (2014)
- [93] Schönhofen, P.: Identifying document topics using the Wikipedia category network. *Web Intelligence and Agent Systems: An International Journal* 7(2), 195–207 (2009)
- [94] Sieg, A., Mobasher, B., Burke, R.: Web search personalization with ontological user profiles. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management. pp. 525–534. ACM (2007)

- [95] Singhal, A.: Introducing the knowledge graph: things, not strings. Official Google blog 5 (2012)
- [96] Skupin, A.: The world of geography: Visualizing a knowledge domain with cartographic means. *Proceedings of the National Academy of Sciences* 101 (suppl 1), 5274–5278 (2004)
- [97] SoftwareGuild: programming languages through the years (2015), <https://www.thesoftwareguild.com/blog/history-of-programming-languages/>
- [98] tasteatlas: Most popular cheeses in the world, <https://www.tasteatlas.com/cheese>
- [99] Tay, D.: Interesting systems and our amazing body (2013), <http://www.tayxiongsheng.com/>
- [100] Thudt, A., Hinrichs, U., Carpendale, S.: The bohemian bookshelf: Supporting serendipitous book discoveries through information visualization. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. pp. 1461–1470 (2012)
- [101] Trigg, R.H.: Guided tours and tabletops: tools for communicating in a hypertext environment. *ACM Transactions on Information Systems* 6(4), 398–414 (1988)
- [102] Vercoustre, A.M., Thom, J.A., Pehcevski, J.: Entity ranking in Wikipedia. In: *Proceedings of the 2008 ACM symposium on Applied computing*. pp. 1101–1106. ACM (2008)
- [103] Weaver, C.: Building highly-coordinated visualizations in improvise. In: *IEEE Symposium on Information Visualization, 2004. INFOVIS 2004*. pp. 159–166 (2004)
- [104] White, R.W., Roth, R.A.: Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services* 1(1), 1–98 (2009)
- [105] Wildemuth, B.M., Freund, L.: Assigning search tasks designed to elicit exploratory search behaviors. In: *Proceedings of the Symposium on Human-Computer Interaction and Information Retrieval*. pp. 4:1–4:10. ACM (2012)
- [106] Witten, D.M.I.H.: An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. *Wikipedia and Artificial Intelligence: An Evolving Synergy* p. 25 (2008)
- [107] Wu, K., Storck, I., Fischer, S.: Javascript tools, <https://www.codefellows.org/blog/a-list-of-foundational-javascript-tools/>

- [108] Yeh, E., Ramage, D., Manning, C.D., Agirre, E., Soroa, A.: Wikiwalk: random walks on Wikipedia for semantic relatedness. In: Proceedings of the 2009 Workshop on Graph-Based Methods for Natural Language Processing. pp. 41–49. Association for Computational Linguistics (2009)
- [109] Yuan, X., White, R.: Building the trail best traveled: effects of domain knowledge on web search trailblazing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 1795–1804. ACM (2012)

Appendix A: Exploratory Tasks

1. The first exploratory task given to subjects in the user study (Section 2.4) involved comparing malicious ads and identity theft; and investigating their relationships with viruses and attachments. The task included the following description.

“Your relative is concerned about multiple ads displayed on her computer. She thinks they are from viruses and they might steal her personal information. She was wondering if the attachments she has received recently from a friend has anything to do with her concerns. You decided to help your relative by learning about computer security threats so you can clarify things to her and also suggest important topics for her to read. Try to explore the subject so you can inform and maybe guide your relative to useful resources.”
2. The second task involved comparing Phishing attacks and SQL Injection. The task included the following description.

“Two CS students were arguing about whether a phishing attack could lead to SQL injection attack. You would like to join the discussion, but you need some solid background about computer security threats. You have to decide which argument you will support or maybe form your own argument. Try to get prepared.”

Appendix B: Labeling Criteria

Table 1: The criteria used by two CS students in labeling the top-ten pages in the I&A-Model investigation (see Section 6.1 for more details). The criteria are checked in order using an if-elseif semantics.

Domain	Diagram	Label	Criterion
Candy bars	Company	1	Candy bar whose manufacturer is the same as the selected candy bar.
		2	Candy bar, which has a variety manufactured by the same company as the selected candy bar. The page contains varieties made by the same company that manufactures the selected candy bar. A candy bar previously owned by the same company that manufactures the selected candy bar.
		3	A candy bar manufactured by a different company. The page is not a candy bar.
Candy bars	Ingredients	1	The page is a candy bar, which contains most of the ingredients in the selected candy bar.
		2	The page is a candy bar, which contains few of the ingredients in the selected candy bar.
		3	The page is a candy bar, which contains none of the ingredients in the selected candy bar. The page is not a candy bar.

Domain	Diagram	Label	Criterion
Car models	Make	1	The selected car and the ranked car have the same make.
		2	The makes of the selected car and the ranked car have the same parent company.
		3	The selected car and the ranked car have different makes of different parent companies.
Car models	Size	1	The selected car and the ranked car have the same size.
		2	The selected car and the ranked car have close sizes.
		3	The selected car and the ranked car have different sizes.
Cheese types	Animal Milk	1	A cheese made with the same milk used to make the selected cheese.
		2	A page related to the selected cheese or the milk used to make it.
		3	A cheese made by a different milk or an irrelevant page.
Cheese types	Country	1	A cheese that is originated in the same country as the selected cheese.
		2	A page is slightly related to the selected cheese.
		3	Not a cheese or an irrelevant page.
Civil war battles	State	1	A battle that is located in the same state as the selected battle.
		2	A battle that is located in a neighboring state.
		3	A battle that is located in other states. The page is not a battle.
Civil war battles	timeline	1	The selected and ranked battles are in the same year. Battles in different years but are few days apart are considered similar too.
		2	The years of the selected and ranked battles are one or two apart.
		3	The years of the selected and ranked battles are three or more apart.

Domain	Diagram	Label	Criterion
Database platforms	DB features	1	The page is a database platform that supports most of the DB features supported by the selected database platform.
		2	The page is a database platform that supports few of the DB features supported by the selected database platform.
		3	The page is a database platform that supports none of the DB features supported by the selected database platform. The page is not a database platform.
Database platforms	DB model	1	The page is a database platform that supports most of the data models supported by the selected database platform.
		2	The page is a database platform that supports few of the data models supported by the selected database platform.
		3	The page is a database platform that supports none of the data models supported by the selected database platform. The page is not a database platform.
World languages	Country	1	A language that is spoken in the same country as the selected language (spoken by a large population, spoken as a foreign language, or a recognized minority language). A place where the selected language is an official language.
		2	A language spoken by only a small group of immigrants in the country of the selected language.
		3	A language that is not spoken in the country of the selected language. The page is irrelevant.
World languages	Language family	1	A language that is in the same tree and branch as the selected language.
		2	A language that is in the same tree, but a different branch as the selected language.
		3	A language that is in a different tree from the selected language's tree.

Domain	Diagram	Label	Criterion
Animal species	Animal kingdom	1	The ranked animal and the selected animal have the same phylum and class.
		2	The label was not used.
		3	The ranked animal and the selected animal have different phyla or different classes.
Animal species	Extinction	1	The ranked animal and the selected animal have the same extinction classification.
		2	The ranked animal have (or had) the extinction classification of the selected animal at some time or in some locations.
		3	The ranked animal and the selected animal have different extinction classifications.
Landforms	Country	1	The ranked landform and the selected landform are located in the same country.
		2	The ranked landform is close to the border of the selected landform's country.
		3	The ranked landform and the selected landform are located in different countries.
Landforms	Landform	1	The ranked landform and the selected landform have the same type.
		2	Label was not used.
		3	The ranked landform and the selected landform have different types.
Vegetables and Fruits	Minerals		Note: If the page is a fruit or a vegetable, check the daily value of the minerals associated with the selected fruit/vegetable in the diagram. A fruit/vegetable is considered a high source of a mineral if it contains 8% or more of the daily value, a medium source if it has 4% to 7% of the daily value, or a low source if it has 3% or less of the daily value.

Domain	Diagram	Label	Criterion
Vegetables and Fruits	Minerals	1	Fruit/vegetable is at least a high source of some minerals and a medium source of the rest of the minerals found in the selected fruit/vegetable.
		2	Fruit/vegetable is at least a high source of one mineral found in the selected fruit/vegetable. Fruit/vegetable is a medium source of most minerals found in the selected fruit/vegetable.
		3	Fruit/vegetable is a low source of most minerals found in the selected fruit/vegetable. The page is not a fruit nor a vegetable.
Vegetables and Fruits	Vitamins	1	Apply the same rule used for label one in the minerals diagram (substitute minerals with vitamins).
		2	Apply the same rule used for label two in the minerals diagram (substitute minerals with vitamins).
		3	Apply the same rule used for label three in the minerals diagram (substitute minerals with vitamins).
World heritage sites (WHS)	Civilization	1	The ranked WHS site and the selected WHS site are created by the same civilization.
		2	The civilization of the selected WHS site influenced the architecture or the art of the ranked WHS site.
		3	The civilization of the selected WHS site is not related to the ranked WHS site.
World heritage sites	Country	1	The ranked WHS site and selected WHS site are located in the same country.
		2	This label was not used in this diagram.
		3	The ranked WHS site and the selected WHS site are located in different countries.

Appendix C: Diagrams Used in the Experiments

I&A-Model Diagrams Used in the Investigation in Chapter 6

Note: In some diagrams used for the I&A-Model experiment, items (especially those selected in D2C queries) are not included in the diagram. We asked the students who labeled the top-ten pages to assume the items selected in the D2C queries were included in those diagrams, and connected to the related attributes. Also, in the I&A-Model experiment, we sometimes found a diagram and create another to have a pair of diagrams with matching items (to show different aspects of a domain). However, in many cases the diagram we create is similar to an existing one. We refer to some of those existing diagrams. We sent a request to use a diagram in experiments and publications for all diagrams whose owner could be identified.

Figure 1: Candy bars by company. (The diagram was created by us. A similar existing diagram can be found at [64].)

Figure 2: Candy bars' ingredients [68].

Figure 3: Car makes [1].

Figure 4: an excerpt from the car sizes diagram (source is unknown). A full diagram can be found at <http://www.car-brand-names.com/types-of-cars/>.

Figure 5: Cheese by type of milk [67].

Figure 6: Cheese by country (diagram was created by us. A similar existing diagram can be found at [98]).

Figure 7: An excerpt from a diagram about the civil war battles by state [44]. The full diagram can be found at <http://kids.britannica.com/elementary/art-87023/Most-of-the-major-battles-of-the-American-Civil-War>

Figure 8: A timeline of the American civil-war battles (diagram was created by us. A similar diagram can found at [2]).

Figure 9: An excerpt from the database platforms diagram [7]. We considered the diagram to cover two aspects of the domain, which are database models and database features. The full diagram can be found at <https://451research.com/state-of-the-database-landscape>

Figure 10: World languages organized by country (diagram was created by us. A similar diagram can be found at [72]).

Figure 11: World languages organized by language families [45].

Figure 12: Animal species organized in the animal kingdom [46].

Figure 13: An excerpt from a diagram that shows animal species organized in extinction categories (source of the diagram is unknown). A full diagram can be found at <https://howtoconserve.org/2015/09/18/endangered-species/>

Figure 14: Landforms organized by country (diagram was created by us).

Figure 15: Landform types. (The diagram was created by us. A similar diagram can be found at [50].) Landform images are free images downloaded from PixaBay.com.

Figure 16: Vegetables and fruits organized by vitamins [43].

Mineral Name	Major Functions	Deficiency Effects	Toxicity Effects	Food Sources
Calcium	Makes up bone and teeth; muscle contraction/relaxation; blood pressure; clotting; nerve function	Children – stunted growth Adults – bone loss (osteoporosis)	Diarrhea, interference with absorption of other minerals	Dairy, fish with bones, tofu, greens, legumes, fortified foods
Chromium	Helps insulin move glucose (sugar) from blood into cells	Abnormal glucose metabolism	Possible muscle degeneration	Meat, whole grains, vegetable oils
Fluoride	Helps make bones and teeth stronger, helps teeth resist decay	Susceptibility to tooth decay	Fluorosis, discolored teeth, nausea, chest pain	Fluoridated water, seafood, tea
Iodine	A component of thyroid hormone – helps regulate growth, development, metabolism	Goiter, cretinism	Low thyroid activity, enlarged thyroid	Iodized salt, seafood, plants grown in iodine-rich soil
Iron	Part of hemoglobin –	Anemia, weakness,	Fatigue, infection,	Red meats, fish, poultry,

Figure 17: An excerpt from a diagram that includes minerals with their food sources [39]. A full diagram can be found at <https://www.deardoctor.com/articles/vitamins-and-dietary-supplements/>

	Book Text: Antiquity	Graphics from the World History Chart	blue words = links
America	Peru : Chavin and Paracas cultures	Mexico : Olmec culture at its peak	
Italy	Villanova culture by Italic peoples (Iron Age)		
Europe	Etruscan civilization in Italy	Legendary reign of se	
	Late Bronze Age	Druids in Gaul and Britain	Ce
Greece	Dorian migration	Iliad & Odyssey composed	Greek C o
	Greek settlements in Ionia (Miletus)	Growth of Greek city states (Polis)	Rise of S
		Olympic Games	Messenian War
			Repu
	Semitic-speaking Arameans from Arabia rule over Mesopotamia	NEW ASSYRIAN EMP	
Asia Minor	Phoenicia (cap. Tyre)	Control of Mediterranean trade	Assyrian Conquest
	Hebrews King David unites Judah and Israel	Kdm Israel (cap. Samaria) Elijah and Elisha	ar
		Judah (cap. Jerusalem)	Isaiah appears
Egypt	The New Kdm	The declining kdm split into Middle Egypt ruled by priests at Thebes and into Lower Egypt dominated by Lybians	Egypt rul by Nubi
	-1000	-900	-800
India	Rig Veda evolves (from ca. 1500 BC)	Caste system evolves	Epic Age
		Aryan expansion into Ganges area	Upanishade
		The two Epics the 'Mahabharata' and the 'Ramayana' relate to events of the epic age	Brahmanas o
China	► to text Chou	Western C H O U (cap. Hao)	A Barbarian attack overwhelmed the old capital Hao and
			Ea

Figure 18: A civilization timeline [5]. Only part of the timeline is shown. A full diagram can be found at <https://www.hyperhistory.com/>

Figure 19: A map of the UNESCO World heritage sites [81].

Universal-Model Diagrams used in the Investigation in Chapter 7

We list the diagrams used in our experiments in Chapter 7. We sent a request to use a diagram in experiments and publications for all diagrams whose owner could be identified.

Figure 20: An excerpt of the JavaScript libraries diagram [107]. A full diagram can be found at <https://www.codefellows.org/blog/a-list-of-foundational-javascript-tools>

Figure 21: Portland Map [77].

Figure 22: The Attack Graph [28]

Figure 23: An excerpt from the Coffee diagram [3]. A full diagram can be found at <https://hubpages.com/food/coffee-maker-buying-guide#>

Figure 24: An excerpt from the Portland bridges diagram [91]. A full diagram can be found at https://www.oregonlive.com/commuting/index.ssf/2014/04/with-trimet-ready_to_name_tran.html

Figure 25: An excerpt from the nursing career flowchart [52]. A full diagram can be found at <https://visual.ly/community/infographic/health/pathways-nursing>

Figure 26: Programming Languages [97]

Figure 27: Data Science Tools [34]

Figure 28: Culinary Herbs. (The diagram was created by us. A similar existing diagram can be found at [37].) Herb images are free images downloaded from Pixabay.com.

World War Two 1939 - 1945		
1 st September	Hitler invaded Poland	1939
3 rd September	Britain and France declared war on Germany	1939
May	Hitler invaded Denmark, Norway, Holland and Belgium	1940
26 th May	British forces pushed back to the beach at Dunkirk	1940
22 nd June	France Surrendered	1940
10 th July – 31 st Oct	Battle of Britain	1940
22 nd September	Tripartite Pact allied Germany, Italy and Japan	1940
December	Italians defeated in North Africa	1940
22 nd June	Hitler invaded Russia – Operation Barbarossa	1941
7 th December	Japanese attacked Pearl Harbor	1941
8 th December	Britain and US declared war on Japan	1941
4 th – 6 th June	Battle of Midway	1942
23 rd October	Battle of El Alamein	1942
21 st August	Battle of Stalingrad began	1942
12 th May	Axis forces surrendered North Africa	1943
July	Allies invaded Sicily	1943
3 rd September	Italy surrendered	1943
November	Allies held Tehran Conference	1943
6 th June	Operation Overlord – D-Day	1944
25 th August	Paris liberated	1944
December	Battle of the Bulge	1944
28 th April	Mussolini captured and executed	1945
30 th April	Hitler committed suicide	1945
4 th May	Germany surrendered	1945
6 th + 9 th August	Atomic bombs dropped on Hiroshima and Nagasaki	1945
14 th August	Japan surrendered	1945
<small>World War Two Timeline © HODIN Press Limited 2019. All Rights Reserved. www.hodinpress.com</small>		

Figure 29: WW2 Timeline [88]

Figure 30: An excerpt from the database platforms diagram [7]. The full diagram can be found at <https://451research.com/state-of-the-database-landscape>

Figure 31: Candy Bar Ingredients [68]

Figure 32: South America Map [32]

Figure 33: Body Systems [99]

Figure 34: An excerpt from the geological eras diagram [57]. A full diagram can be found at <https://carnegiemuseums.org/carnegie-magazine/winter-2016/earth-in-the-age-of-humans/>

HOW TO IDENTIFY ROCKS

IGNEOUS INTRUSIVE

ROCKS are made when molten rock cools while still underground.

SEDIMENTARY ROCKS are made when pieces of rocks settle from the water or when minerals are deposited by plants, animals, and other chemical processes.

Source: Bill Langer, U.S. Geological Survey.

<http://www.bedrockquarry.com/images/identify-rocks.jpg>

Figure 35: Rock Types [71]

Figure 36: Space Discovery Timeline [65]

Q WHO OWNS THE BIG TV NETWORKS?

Figure 37: TV Networks [30]

Figure 39: Cheese Types [67]