Visual Index Tree and Boolean Creator: Final User Interface Design Project

Group 1: Caitlin Peraria, Jade Pritz, & Karin T. Haynie

Department of Library Science, Clarion University

LS 542: Information Seeking Behavior in User Interface Design

Dr. YooJin Ha

May 6, 2022

Visual Index Tree and Boolean Creator: Final User Interface Design Project

Undergraduate students in science and health science programs are required to conduct literature searches to support and/or disapprove a chosen topic. Properly utilizing library resources becomes an essential skill, but many first and second-year undergraduate students are not proficient in conducting literature searches, a critical component of their education. Most student users are unaware that academic search systems use an index or controlled vocabulary to categorize materials. Students instinctively utilize a "Google-like" search strategy (natural language, colloquial terms) that makes sense to them, but become frustrated with the lack of results returned. This is due to incompatibilities between indexed systems and natural language.

Since most academic libraries use the Library of Congress subject headings (LCSH) for searching, we are utilizing its index. A hypothetical search query was created: "babies with heart defects". Group One's user interface (UI) design would handle such a query by breaking it into keyword chunks, then translating them into index terms to assist in term expansion and contraction within the search. This would essentially result in relevant Boolean queries. The goal of Group One's UI design project is to make searching easier for students by transparently translating their "Google-like searches" into the "proper language" of academic search systems, utilizing index and Boolean queries.

Related Work

In 1977, Bates conducted an experiment with college students to determine whether a searcher's familiarity with the subject they are exploring has any bearing on the recall and precision of intended search results. Juniors and seniors in two different majors were asked to list what terms they would use to get information on specific queries. One of the queries was within

their field. The other query occurred in the opposite major's field. Bates (1977) wished to see how often and closely the search words chosen by students actually matched terms used in the catalog's index (i.e., the students would get a returned result to their query). The catalog utilized by Bates was rather primitive compared to today. The catalog consisted of books physically present in the library with a limited number of subject terms per item in the catalog. The concept of subject familiarity vs. index stands. Points of varying levels were given to exact matches (4), partial matches (referring to the number of words that did have a match within a phrase) (2-3), the root of a term present (1), or zero matches (0). At best, students achieved 21.5% in usable queries, and 61% of vague queries (Bates, 1977). The experiment was conducted a second time with library science students as the test group. Factors included assumption of intimate knowledge of index organization and little knowledge of the subjects they were searching. These factors are the opposite of the two previous student groups studied. The library science student group achieved 35% on usable matches and 64% on vague matches. Knowing the catalog index resulted in substantially better search results than having high knowledge about the subject itself (Bates, 1977).

The current method of query searching is through an academic library database's controlled vocabulary (CV), using the limitation of the Library of Congress Subject Headings (LCSH). As the LCSH "cover all knowledge domains and projects" (Julien, Tirilly, Leide & Guastavino, 2012), this makes it a prime candidate for exploring a hierarchical structure. Issues addressed include that user must browse each CV individually, hiding the web of interrelated concepts from the user's view. Related concepts are defined by the author as 'a hierarchical relationship' (or the relationship between a broader term and a narrower term) or 'an associative relationship' (or the relationship between two related terms).

Julien et al. (2012) attempt to construct a tree-like structure for the science and engineering collections, noting “the most connected group was associated with the domain of Science; consequently, this domain is chosen for this study since it offers the most complex structure”. They go on to match each CV term with an authority record, either exactly or including the removal of one subdivision or more. They were left with 18,318 CV terms that matched their specifications, some of which were orphan terms (terms with no broader heading). This brought up a few issues within their research. LCSH immediately balks at a tree-like structure, violating the rule that each branch on the tree can only have one root. CV terms can have multiple roots. The authors attempted to reduce this problem by duplicating branches but found that the effort expanded the tree to unmanageable sizes. The authors also had to create a common root on their own as no such capability exists within the LCSH.

Julien et al. (2012) state that while a tree structure is possible (at least within the Science and Engineering collections), it also brings with it multiple problems that could impact its usability. Mostly it was highly redundant, riddled with problems that would impact the tree structure (i.e. no root heading, multiple relationships per CV term, and some terms being a root and a branch of another term), and needlessly complex.

Hjørland (2015) discusses many problems users face when using Boolean-oriented search systems. The first flaw is that a Boolean search is highly dependent on human expertise of indexes and phrasing syntax. This is a problem since many of the most popular and useful academic databases (i.e. MEDLINE, PsychInfo) are most thoroughly and accurately searched via indexed terms in Boolean phrasing. To make the situation worse within search systems, some fields must be populated with index terms, while others are more reliant on natural language. A mix of the two field types can be commonly found on the same query form. The query forms

rarely indicate which type of language should be used in designated fields, and an index (if needed) is not easily accessible. Boolean logic is also terrible at title searching because it cannot interpret figures of speech or metaphors, especially if quotation marks are not used. Boolean logic is highly literal.

Hjørland (2015) also acknowledges problems brought up in his literature review of Boolean logic. Specifically, construction of Boolean query phrasing is not clear or consistent across systems. The phrase "AND" can make searches too narrow. The phrase "OR" can be too broad. Using the phrase "AND/OR" is counterintuitive to those not acquainted with Boolean logic. Proper and consistent use of applicable indexes must be utilized in most systems as "Boolean logic treats each term equally" instead of weighted by any importance or relevance with results rarely ranked by applicability (p. 1566).

According to O'Neill et al. (2014), using "authorities [when setting up a database] can significantly improve searching by increasing the number of access points," (p. 6). Regardless of this benefit, in practice authorities are rarely utilized by common users. Even vast universal vocabularies such as the LCSH have not appeared to help. Users also rarely employ Boolean logic in searches--a key component in many indexed search systems. O'Neill et al. (2014) have found, "from the perspective of the typical end-user, intuitiveness and ease of use are more important than functionality [multiple entry points, etc.]. Advanced features, while often demanded by experienced searchers, [can be] an impediment to end users." (p. 8). It is a shame that indexing does not help common users because index hierarchy enhances a user's ability to explore a subject by also linking to broader and related terms (O'Neill et al., 2014).

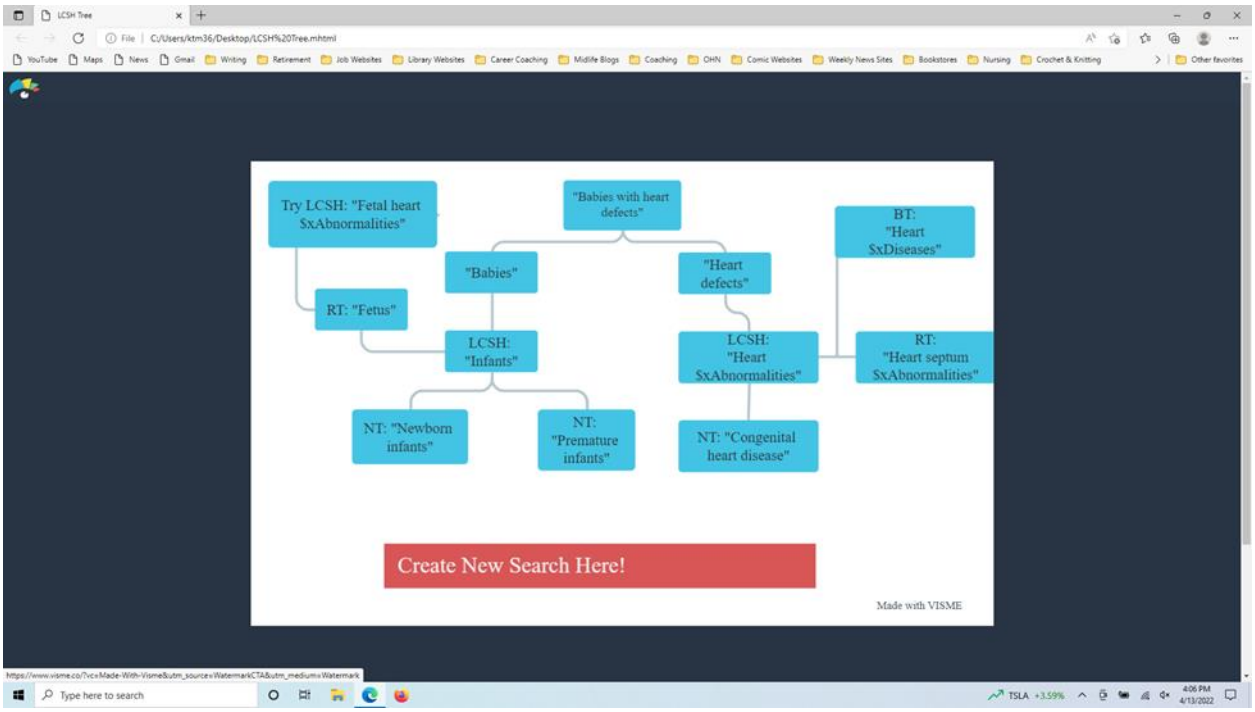
Group One's search system interface attempts to alleviate these concerns. Bates's (1977) quandary regarding the need for users to understand the catalog's index is being addressed by

utilizing visual index term trees. Julien et al. (2012) show that there have been efforts in making a tree structure for the LCSH and illuminate the problems that would likely to crop up, allowing Group One to try to circumvent those. Hjørland's emphasis (2015) on user need to grasp Boolean phrasing by is being considered by creating a simple "include/do-not-include" color coding system that creates a semantically proper Boolean phrase for the user. O'Neill et al. (2014) feel that the benefits of advanced search features should not outweigh ease of use; to avoid that burden, Group one is inserting a "Friendly Search Wizard," to reduce user confusion. By addressing these concerns with a user interface design that is friendly and easier to use, students should experience less anxiety and frustration when conducting a search query in an academic library database.

Justinmind is a user interface prototype design program that can help students visualize their design and evaluate it. Justinmind is a suitable application design platform that functions well in the online class environment since it can produce prototypes compatible with different screen sizes and devices (Justinmind, 2022). Rahman, Yusof, Zin, and Zukarnain (2021) used Justinmind as a prototype and wireframe design tool in their software development classes. Students evaluating Justinmind found it to be an enjoyable design tool that allowed them an opportunity to produce and evaluate a prototype design of their choosing. Students who utilized Justinmind collaborated in an innovative and exciting environment that increased their social presence in an online course (Rahman, Yusof, Zin, & Zukarnain, 2021). Justinmind's user experience allows the user to conduct a cognitive walkthrough of the design. Cognitive walkthrough is a usability evaluation strategy "in which one or more evaluators work through a task scheme and ask questions from the user's perspective" (Dewi, Dantes, & Indrawan, 2020, p. 12024). Thus managing prototype designs that meet the representative user's perspective.

Method

First the user interface breaks down the query into key terms. Then aligns each term with its closest index term. All that is listed under/above that term is in a "phylogenetic tree"-like layout, giving users the ability to "see" what direction the search could go. The user would then select the terms by selecting a color designated for OR and one designated for AND (forgot to add another for "NOT" on the diagram). Clicking each term would open a mini tree under it for more specific terms. A similar tree would also be created for "babies" (from related LCSH section) in this search example. This tree ultimately states (heart OR cardiac OR cardiogenic) AND (abnormalities OR congenital OR diseases). The simple visual flowchart below for just the term "heart" in a query such as "babies with heart defects":



When utilizing a subject index tree, the user starts with a search query of key terms: "Babies with heart defects"

The library database used these keywords: "Babies", "Heart defects"

Keyword : “Babies”

Controlled term: "Infants"

Narrower terms: "Newborn infants"; "Premature infants"; "Infant boys"; "Infant girls"

Keyword: "Heart defects"

Controlled term: "Heart $x Diseases"

Narrower terms:

INFORMATION FOR: Heart--Diseases.

Please note: Broader Terms are not currently available

Select a Link Below to Continue...
Authority Record
Narrower Term: Arrhythmia.
Narrower Term: Carcinoid heart disease
Narrower Term: Cardiac arrest.
Narrower Term: Cardiac tamponade.
Narrower Term: Cor pulmonale
Narrower Term: Coronary heart disease.
Narrower Term: Fatty heart
Narrower Term: Heart block
Narrower Term: Heart--Dilatation
Narrower Term: Heart diseases in pregnancy.
Narrower Term: Heart diseases in women.
Narrower Term: Heart failure.
Narrower Term: Heart--Fibrosis
Narrower Term: Heart--Hypertrophy.
Narrower Term: Heart--Infections
Narrower Term: Heart--Metabolism--Disorders
Narrower Term: Heart--Necrosis
Narrower Term: Heart--Parasites
Narrower Term: Heart--Rupture
Narrower Term: Nonatherosclerotic myocardial ischemia
Narrower Term: Postpericardiotomy syndrome

Related terms: "Heart atrium $x Diseases"; "Heart $x Blood-vessels $x Diseases"; "Heart $x Blood-vessels $x Abnormalities"; "Heart septum $x Abnormalities"; "Heart valves $x Abnormalities"; "Heart valves $x Diseases"; "Myocardium $x Diseases"; "Pericardium $x Diseases"; "Endocardium $x Diseases"

Controlled term: "Heart $x Abnormalities"

Narrower term: "Congenital heart disease"

Controlled term: "Fetal heart $x Abnormalities"

Narrower term: "Fetal heart $x Blood-vessels $x Abnormalities"

Justinmind is a user interface prototype design program that can help students visualize their design and evaluate it. Justinmind is a suitable application design platform that functions well in the online class environment since it can produce prototypes compatible with different screen sizes and devices (Justinmind, 2022). Rahman, Yusof, Zin, and Zukarnain (2021) used Justinmind as a prototype and wireframe design tool in their software development classes. Students evaluating Justinmind found it to be an enjoyable design tool that allowed them an opportunity to produce and evaluate a prototype design of their choosing. Students who utilized Justinmind collaborated in an innovative and exciting environment that increased their social presence in an online course (Rahman, Yusof, Zin, & Zukarnain, 2021). Justinmind's user experience allows the user to conduct a cognitive walkthrough of the design. Cognitive walkthrough is a usability evaluation strategy "in which one or more evaluators work through a task scheme and ask questions from the user's perspective" (Dewi, Dantes, & Indrawan, 2020, p. 12024). Thus managing prototype designs that meet the representative user's perspective.

Results

Based on external evaluations using the Nielsen's 10 Usability Heuristics for User Interface Design (2020), the revised prototype met/violated the following heuristics principles:

Heuristic Principle	Met the Principle	Violated the Principle

Visibility of system status	X(2)	X
Match between system and the real world	X (3)	
User control and freedom		X (3)
Consistency and standards	X (2)	X
Error prevention	X	X (2)
Recognition rather than recall	X (2)	X
Flexibility and efficiency of use	X (3)	
Aesthetic and minimalist design	X (3)	
Help users recognize, diagnose, and recover from errors	X (2)	X
Help and documentation	X	X (2)

Note: M. Schachte, E. Palmer, and S. Conway were the external evaluators of this revised prototype. “X (2)” refers to multiple external evaluators agreeing for that specific heuristic principle.

Discussion

Group One was assigned three evaluators from other groups to assess the revised visual index tree prototype. The following heuristic criteria were 100% favored by the evaluators: "Match between system and the real world"; "flexibility and efficiency of use" (of the revised prototype); and the appeal of the "aesthetic and minimalist design". No additional changes were recommended for these criteria; however, Group One will revisit these criteria areas to improve the prototype design after final revisions are made.

For the "user control and freedom" heuristic criterion, 100% of the evaluators indicated that Group One violated this criterion. One of the evaluators indicated that the revised prototype was "confusing and difficult to understand" in that it didn't allow the user a choice to go backward to a prior search. Another evaluator suggested that the user be given the ability to move forward and backward when needed throughout the search. The third evaluator recommended an "undo" button to allow the user to go backward if their search failed. This criterion area will be addressed in the next version of this prototype.

For two of the heuristic criteria--"error prevention" and "help & documentation"--67% of the evaluators indicated that Group One violated these criteria. These evaluators recommended that as part of fixing the "user control and freedom" criterion, that an additional button to allow the user to move back and forth within the search query would also fix these criteria. Adding error messages to the screen where a search fails, would allow the user to move backward to redo part or all of the search. These criteria will also be addressed in the next version of this prototype.

Conclusions

The tree structure presents the user with something more clearly understandable than the traditional index, allowing them to not only tailor their search to the limiters but to narrow down more easily to a specific topic. As the search translates their query into the controlled vocabulary of the LCSH, Group One's project also subconsciously trains the user to use controlled vocabulary, without making finding that vocabulary needlessly difficult. For the prototype at least, the tree also linked together controlled vocabulary that was not originally linked together in the LCSH, showing that there is a great deal left to do on that system to make it truly work as a tree structure.

The Boolean creation side of the prototype makes Boolean searching more understandable to the user. While there are guides on the Internet to how to use the Boolean limiters, it's easy to forget when the user is stuck with unfamiliar language and a database that doesn't present the limiters in a user-friendly way. It's easy to drop the limiters if your initial search brings nothing up either because the user is not using the right words or is using the wrong Boolean limiters in the wrong places. Introducing a tool that not only supplies users with the correct language but also creates Boolean limiters on its own will save users a great deal of stress and frustration than using a traditional database.

Group One is aware that the revised prototype needs more work, but are proud of our first attempt at creating a UI. The external evaluator comments will be helpful in improving this prototype to make searching topics easier and less anxiety provoking for students using an academic library database. We covered several aspects very well. The facets we did not cover sufficiently are easily fixable (ex: adding the unanimously desired back button, clearer directions). Of course, when adding more LCSH terms, we may run into unforeseen problems with the hierarchy system and how it will present on screen.

References

Bates, M. J. (1977). Factors affecting subject catalog search success. *Journal of the American Society for Information Science*, 28 (3), 161–169. https://doi.org/10.1002/asi.4630280304

Dewi, P. W. S., Dantes, G. R. and Indrawan, G. (2020). User experience evaluation of e-report application using cognitive walkthrough (cw), heuristic evaluation (he) and user experience questionnaire (ueq). *Journal of Physics: Conference Series*. IOP Publishing, p. 12024.

Hjørland, B. (2015). Classical databases and knowledge organization: A case for Boolean retrieval and human decision-making during searches. *Journal of the Association for Information Science & Technology*, 66(8), 1559–1575. https://doi.org/10.1002/asi.23250

Julien, C.-A., Tirilly, P., Leide, J. E., & Guastavino, C. (2012). Constructing a true LCSH tree of a science and engineering collection. *Journal of the American Society for Information Science & Technology*, 63(12), 2405–2418. https://doi-org.proxy-clarion.klnpa.org/10.1002/asi.22749

Justinmind. (2022). Justinmind: All-in-one prototyping tool for web and mobile apps, Justinmind. Retrieved April 14, 2022 from https://www.justinmind.com/

Neilsen, J. (2020 November 15). *10 usability heuristics for user interface design*. Nielsen Norman Group. Retrieved April 26, 2022, from https://www.nngroup.com/articles/ten-usability-heuristics/

O'Neill, E. T., Bennett, R., & Kammerer, K. (2014). Using Authorities to Improve Subject Searches. *Cataloging & Classification Quarterly*, 52(1), 6–19. https://doi.org/10.1080/01639374.2013.850018

Rahman, N.A., Yusof, R., Zin, N.A.M., & Zukarnain, Z.A. (2021). Developing prototype in online class using Justinmind: A COVID-19 story. *Journal of Computing Research and Innovation*, 6 (2), 84-94. https://doi.org/10.24191/jcrinn.v6i2.199