Mind-Map Based User Modeling and Research Paper Recommender Systems

JOERAN BEEL, Otto-von-Guericke University
STEFAN LANGER, Docear
GEORGIA M. KAPITSAKI, University of Cyprus
BELA GIPP, University of California, Berkeley

Mind-maps have not received much attention in the user modeling and recommender system community, although they contain lots of information that could be valuable for user modeling and recommender systems. For this paper, we explored the effectiveness of standard user modeling approaches applied to mind-maps, and developed novel user modeling approaches that consider the unique characteristics of mind-maps. We applied and evaluated the approaches with our mind-mapping and reference management software Docear. Docear displayed 431,112 research paper recommendations, based on 4,701 users' mind-maps, from March 2013 to August 2014. The evaluation shows that standard user modeling approaches are reasonably effective when applied to mind-maps, with click-through rates (CTR) between 1.05% and 4.12%. However, when adjusting user modeling to the unique characteristics of mind-maps, a higher CTR of 9.82% could be achieved. The adjustments included, among others, a restriction of the user model size to 35 terms. These terms were extracted from the users' most recently moved nodes within the past 90 days. Nodes were weighted based on their depths and the number of siblings. The terms of the nodes were weighted with a novel weighting scheme (TF-IDuF) that might also be relevant for recommender systems in general. Overall, our results reinforced our astonishment that mind-maps are being disregarded by the user modeling and recommender system community, and that currently no mind-mapping tools feature any recommender systems. Our research shows that mind-map specific user modeling has a high potential. We hope to initiate a discussion that encourages researchers to do research in this field and that encourages developers to integrate recommender systems to their mind-mapping tools.

Categories and Subject Descriptors: H.3.3 Information Search and Retrieval: Information Storage and Retrieval

General Terms: Management, Measurement, Performance, Human Factors

Additional Key Words and Phrases: Recommender systems, user modeling, mind-maps

1. INTRODUCTION

Mind-mapping is a technique to record and organize information, and to develop new ideas [Holland et al. 2004]. As such mind-maps are used for tasks like brainstorming, knowledge management, note taking, presenting, project planning, decision making, and career planning [Frey 2010]. Originally, mind-mapping is done with pen and paper, but since the 1980's more than hundred software tools evolved to aid users in creating mind-maps [Gee 2012]. These tools are used by an estimated two million users who create around five millions mind-maps every year [Beel, Langer, Genzmehr, et al. 2014]. Around 300,000 mind-maps are publicly available in so-called “mind-map galleries”, i.e. websites on which users can upload their mind-maps and other users can browse and download these mind-maps.

Mind-maps received significant attention in various research fields. In human computer interaction, Faste and Lin [2012] evaluated the effectiveness of mind-mapping tools and developed a framework for collaboration based on mind-maps. In the field of document engineering and text mining, Kudelic et al. [2012] created mind-maps from texts automatically, and Bia et al. [2010] utilized mind-maps to model semi-structured documents, i.e. XML files and the corresponding DTDs, schemas, and XML instances. In the field of education, Jamieson [2012] researched how graph analysis techniques could be used with mind-maps to quantify the

1 http://www.mindmeister.com/public
2 https://www.mindomo.com/mind-maps
3 http://www.xmind.net/share/
learning of students, and Somers et al. [2014] used mind-maps to research how knowledgeable business school students are.

In the field of user modeling and recommender systems research, mind-maps received no attention yet. However, mind-maps typically contain information that reflects the interests, knowledge, and information needs of the mind-maps’ authors. In addition, the content of mind-maps is comparable to the content of emails, web pages, and research articles [Beel and Langer 2011]. Hence, we believe that mind-maps should be equally well suited for user modeling as are emails [Ha et al. 2012], web-pages [Göksedef and Gündüz-Ögüdücü 2010], and research articles [Zarrinkalam and Kahun 2013].

Mind-maps are similar to outlines and consist of three elements, namely nodes, connections, and visual clues. To begin mind-mapping, users create a root node that represents the central concept that the users are interested in [Davies 2011]. To detail the central concept, users create child-nodes that are connected to the root node. To detail the child-nodes, users create child-nodes for the child-nodes, and so on. Nodes typically contain a few terms, and may link websites, or files on the users’ hard drives. In some cases, mind-mapping tools offer additional features, for instance to add images, notes, or formulas. Some example mind-maps are shown in Figure 1, Figure 2, and Figure 3. We created the mind-maps with our mind-mapping and reference management software Docear [Beel et al. 2011].

The mind-map in Figure 1 was created to manage academic PDF files and references. We created categories reflecting our research interests (“Academic Search Engines”), sub-categories (“Google Scholar”), and sorted PDFs to the (sub-) categories. Docear imports annotations (comments, highlighted text, and bookmarks) that we made in the PDFs, so we can manage the annotations in the mind-map. A click on a
PDF icon opens the linked PDF file. Docear also extracts metadata for the PDF files (e.g. title and journal name), and displays the metadata when the mouse is moved over a PDF icon. A circle at the end of a node indicates that the node has child nodes, which are currently hidden ("folded"). Figure 2 shows a mind-map that we created to organize academic conferences and journals (red arrows indicate a hyperlink to a web-page). Figure 3 shows a mind-map that we created for career planning. Other popular tasks that mind-maps are used for include brainstorming, knowledge management, note taking, presentation, project planning, and decision making [Frey 2010].

In practice, only two companies utilized mind-maps for user modeling, more precisely for personalized advertisement. MindMeister extracted terms from the node that a user last edited or created and used these terms as user model. The user model, i.e. the terms, were sent to Amazon’s Web Service as search query. Amazon returned book recommendations that matched the query, and MindMeister displayed the...
recommendations in a window next to the mind-map (Figure 4). For instance, if the author of the mind-map in Figure 4 had created or edited the node "Content Based Recommender", these words had been sent to Amazon. Mindomo applied a similar concept, only that Google AdSense instead of Amazon was used. Both companies have since abandoned their user modeling systems, although they still actively maintain their mind-mapping tools in general. In an email, Mindomo explained that “people were not really interested” in the advertisement [Beel, Langer, Genzmehr, et al. 2014]. We were surprised about Mindomo’s statement, because we expected mind-maps to be an effective source for user modeling.

Figure 4: Personalized advertisement from Amazon (right) based on a mind-map (left) in MindMeister

To explore the effectiveness of mind-map-based user modeling, we re-implemented MindMeister’s approach, and used it in our mind-mapping and reference management software Docear to recommend research papers [Beel, Langer, Genzmehr, et al. 2014]. Instead of using Amazon’s Web Service or Google AdSense, we built our own corpus of recommendation candidates and used Apache Lucene to match recommendation candidates with the user models. In Docear, MindMeister’s approach, i.e. utilizing the terms from the last edited or created node, achieved click-through rates (CTR) between 0.2% and around 1%. Compared to other recommender systems [Liu et al. 2010; Mönnich and Spiering 2008; Said et al. 2013], such a CTR is disappointing, which might explain why Mindomo and MindMeister abandoned their recommender systems.

MindMeister’s user modeling approach is one of three content-based filtering (CBF) approaches that we consider rather obvious to use with mind-maps. The second rather obvious approach is to build user models based on all terms contained in the users’ current mind-map. The third approach is to utilize all terms from all mind-maps a user ever created. We implemented the second and third approach as well [Beel, Langer, Genzmehr, et al. 2014]. With Docear, both approaches achieved CTRs of around 6%, which is a reasonable performance and significantly better than the first approach. We were surprised that rather similar user modeling approaches differed in their effectiveness by a factor of six. Apparently, small differences in the algorithms – such as whether to utilize terms from a single node or from the entire mind-map – have a significant impact on user modeling performance. The question
arises if, and how, the effectiveness can be further increased by adjusting the user modeling approach to the unique characteristics of mind-maps.

The characteristics of mind-maps are manifold. First, mind-maps are often personal, while most other document types such as emails and research papers are meant to be seen by at least one other person. As such, we assume that mind-maps’ content might not be as well formulated as content in other documents. For instance, there might be more abbreviations or grammar errors because users do worry about others who might not be able to understand the content of the mind-maps. Second, mind-maps might be less “standardized”. While the structure, e.g., of research articles is rather standard (title, abstract, headings, body text, etc.), mind-maps are used for various tasks. We would assume that mind-maps used for project planning differ in structure and content from a mind-map used to plan a vacation. This might lead to challenges when it comes to selecting and weighting certain features, e.g. terms, of a mind-map. While it is rather obvious that terms in a title of a research paper are more meaningful than words in the paper’s appendix, such obviousness does not exist for different nodes in mind-maps. Third, mind-maps evolve over time and user-modeling applications typically have access to the revisions of a mind-map. Consequently, a mind-map based user modeling system might consider how mind-maps have evolved over time. In contrast, other user modeling applications typically get access to content when the items are finally published, and cannot consider the evolution. However, the evolution of mind-maps is likely very dissimilar. A brainstorming mind-map might have a life span of a few hours. A mind-map for planning the next vacation probably has a life span of a few weeks. A mind-map for managing literature might be used over several years. This might lead to challenges when the evolution of the mind-map should be considered by a user modeling system. Further differences relate to the formatting and layout options in mind-maps since mind-mapping tools usually offer various formatting options, or the hiding of nodes.

Given the characteristics of mind-maps, we hypothesize that standard user modeling approaches will not result in optimal effectiveness when applied to mind-maps. Therefore, our main research goal is to identify variables that influence the effectiveness of user modeling approaches using mind-maps. Identifying these variables allows the development of a mind-map specific user modeling approach that should perform significantly better than the trivial user modeling approaches that we already implemented. From such a mind-map specific user modeling approach, millions of mind-mapping users could benefit [Beel, Langer, Genzmehr, et al. 2014]. As we conduct our research in the context of research paper recommender systems, some of our findings may also be useful for the research paper recommender system community.

In the remainder of this paper, we first give an overview of related work on recommender system and user modeling research. We show which user modeling approaches exist, and discuss their suitability for mind-maps. In the methodology section, we explain which approaches Docear uses to create user models, and how their effectiveness is evaluated. In the results section, we present comparisons of different user modeling approaches, and show that our mind-map specific user modeling approach is more than twice as effective as standard user modeling approaches.

2. RELATED WORK

Content based filtering (CBF) is one of the most widely used and researched recommendation approaches [Lops et al. 2011]. The central component of CBF is the
user modeling process, in which the interests of users are inferred from the items that users interacted with. ‘Items’ are usually textual, for instance emails [Paik et al. 2001], webpages [AlMurtadha et al. 2011], or research articles [Lee et al. 2013; Vellino 2013]. ‘Interaction’ may be established through actions such as buying [Huang et al. 2002], authoring [Zhou et al. 2008], downloading [Jack et al. 2012], or citing [Caragea et al. 2013] an item. Since mind-maps are also mostly textual, they should be generally suitable for CBF, and the most relevant ‘interaction’ that we see between users and mind-maps would probably be authorship. Some of the most important questions relating to CBF are:

(1) Which items to examine and how to weight them? Often, recommender systems examine all items of a user, for instance all emails authored, or all books bought. This is not necessarily sensible, because preferences of users can change over time. This issue is called concept drift or “preference stability” [Burke and Ramezani 2011]. Concept drift is domain dependent [Burke and Ramezani 2011]. In the case of Docear, and research paper recommender systems in general, we assume that concept drift is rather slow. Researchers typically do not change their preferences much and often work for many years in the same research field. Sugiyama and Kan are among the few who briefly researched this issue [Sugiyama and Kan 2010]. They found that among the papers authored by a user, only those authored in the past three years should be considered for user modeling. Analyzing older papers decreased effectiveness.

(2) Which features to utilize and how to weight them? Typical features utilized by recommender systems are terms. In the field of research paper recommender systems, citations [Bollacker et al. 1998] and authors [Zarrinkalam and Kahani 2013] have been used in addition to terms. To weight features, TF-IDF is typically applied. Giles et al. also applied TF-IDF to citations and called this approach CC-IDF [Bollacker et al. 1998]. Feature weighting might also differ by the document fields where a feature occurs. For instance, terms occurring in the title are often weighted stronger than terms occurring in the abstract [Nascimento et al. 2011]. In case of mind-maps, terms are the most obvious feature to use, although hyperlinks or citations could also be used, if they are available in a mind-map, which is not always the case [Beel and Langer 2011].

(3) How to match user models and recommendation candidates? Matching user models and recommendation candidates is a crucial task for recommender systems. However, matching mind-map based user models with e.g. research papers should not differ from matching user models based on other items. We will therefore ignore this aspect in the remainder, and apply standard matching methods such as cosine similarity in the vector space model.

Collaborative Filtering (CF) is a commonly used alternative to CBF. However, for mind-map specific recommender systems, CF seems less interesting. First of all, classic CF is domain independent, and as such there cannot be any mind-map specific CF, in particular since we do not want to recommend mind-maps but research papers. Second, implicit CF could be mind-map specific, because special techniques might be needed to infer ratings from the items linked in the users’ mind-maps. However, mind-map users tend not to link many items in their mind-maps. Of MindMeister’s (public) mind-maps 75.27% contain no links to webpages [Beel and Langer 2011]. Only 8% contain more than ten links. Because the webpages are about all kinds of topics, sparsity would be too high to apply implicit CF. Also for Docear, implicit CF seems inappropriate. Although many of Docear users cite/link a significant number of articles/PDF files, there are only a few thousands of active Docear users, but millions
of academic articles. We found that of 616,635 papers that were linked in the user's mind-maps, 224 papers (0.036%) were linked by two different users, three papers (0.00049%) were linked by three different users, and no paper was linked by more than three different users. This means, 99.96% of the papers were only linked by a single user. In other words, barely any users have some PDFs in common and if they do, they have at maximum three in common. This makes the application of (implicit) collaborative filtering infeasible. We also investigated the possibility of applying content-boosted CF [Balabanovi and Shoham 1997; Melville et al. 2002], but sparsity was still very high. In addition, there is a technical problem. The user's papers are not uploaded to our servers, and we have no information where the papers are available for download. As such, even if sparsity was low, we could not recommend the papers in a sensible way.

Stereotyping is another option to give recommendations to mind-mapping users. It is one of the earliest user modeling and recommendation approaches and was introduced in 1979 by Elaine Rich [Rich 1979]. In stereotyping, one assumes that users will like what their peer group is enjoying. For instance, the travel agency Orbitz observed that Mac users tended to book pricier hotels than PC users [Mattioli 2012]. Therefore, Orbitz assigned their visitors to the stereotype “Mac User” or “PC user”, and favored either cheaper or pricier hotels in the search results lists. According to Orbitz, all parties benefited from this approach. Users received more relevant hotel search results, and Orbitz received higher commissions. For mind-mapping users, stereotyping might be an appropriate recommendation approach – for instance, books or seminars about mind-mapping could be recommended.

There are many additional aspects that are important for recommender systems, for instance trust, presentation, and robustness [Ricci et al. 2011]. However, these aspects are likely not influenced by the decision whether to utilize mind-maps or other items. Thus, we will ignore these aspects in this paper, and focus on the unique characteristics of mind-maps and their effects on user modeling and recommender systems.

3. METHODOLOGY

In a brainstorming session, we identified 28 variables that might affect the effectiveness of user modeling based on mind-maps. Due to time restrictions, we decided to implement and evaluate only a few variables that we considered most promising, and for which an evaluation with Docear was feasible. A variable that we considered not feasible was the “position of a node”. In most mind-mapping tools, users can arrange their nodes freely. We would assume that depending on the position of a node, its importance differs, and hence the weighting for user modeling should differ. For instance, a node far away from the root node could be weighted differently than a node in close proximity to the root node. However, Docear positions nodes automatically, which is why we could not analyze the influence of freely positioned nodes.

The variables we focused on included, the number of mind-maps to analyze, the number of nodes to utilize, the size of the user model, whether to use only visible nodes, and different weighting schemes including standard schemes like TF-IDF but also mind-map-specific weighting schemes based, for example, on the number of children a node has.

* In the analysis, we ignored papers that Docear automatically adds to user's mind-maps as part of demo-projects.
From March 2013 to August 2014, Docear’s recommender system delivered 45,232 recommendation sets, with 431,112 recommendations to 4,701 users. Recommendations were created on Docear’s web server. To enable communication between the Docear Desktop software and the server, a RESTful Web Service was implemented. Recommendations were displayed to users every five days upon the start-up of Docear. In addition, users could request recommendations manually at any time. Recommendations were displayed as a list of ten research papers, for which the titles were shown. A click on a recommendation opened the paper in the user’s web browser. All papers are freely available as full-text PDF files. The publication corpus used for recommendations included around 1.8 million documents from various languages, and various research fields. For more details on the recommender system please refer to [Beel, Langer and Gipp 2014].

Each set of recommendations typically contained ten research papers, and was created with a randomly assembled algorithm. That means, whenever recommendations were requested, a random value was chosen for each variable. For instance, one algorithm might have utilized visible nodes from the 2 most recently modified mind-maps, weighted the terms of these nodes with **TF-IDF**, and stored the 25 highest weighted terms in the user model. Another algorithm might have used the 250 last modified nodes (visible and invisible) among all mind-maps, weighted the citations of these nodes with **TF-only**, and stored the 5 highest weighted citations in the user model.

To measure effectiveness and identify the optimal values for the variables, we compared click-through rates (CTR). For instance, to evaluate whether a user model size of ten or 100 terms was more effective, CTR of all algorithms with a user model size of ten was compared to CTR of all algorithms with a user model size of 100. CTR is a common evaluation metric that describes the ratio of delivered recommendations by those that were clicked. CTR is similar to the ‘precision’ metric, which is commonly used in offline evaluations of recommender systems, and which measures how many recommendations in a list we were accurate. There is a lively discussion about online evaluations, offline evaluations, and different evaluation metrics such as CTR. We do not discuss this issue here but refer to a recent publication, in which we showed that online evaluations are preferable over offline evaluations, and that CTR seems to be the most sensible metric for our purpose [Beel and Langer 2014a]. In that publication we also explain why showing only the title of the recommended papers is sufficient for our evaluation, instead of showing further information such as author name and publication year.

After finding the optimal values for each variable, we combined the optimal values in a single algorithm and compared this algorithm against four baselines to analyze whether the mind-map specific user modeling performed better than the baselines. One baseline was the stereotype approach. For this approach, we generalized that all Docear users are researchers, and that all researchers would be interested in books and research papers about academic writing. Hence, when the stereotype approach was randomly chosen, a list of ten papers relating to academic writing was recommended. These papers were manually selected by the Docear team. The second, third and fourth baseline were those CBF variations that are rather obvious: a) the approach of MindMeister, in which only terms of the most recently modified node are analyzed for the user model (‘modified’ means created, edited or moved); b) all terms of the user’s current mind-map are used for the user model; c) all terms of all mind-maps that the user ever created are utilized for the user model.
Our methodology has a limitation. Determining the optimal values for each variable separately ignores potential dependencies. For instance, only because a user model size of 100 terms is most effective on average, and analyzing 500 nodes is most effective on average, does not mean that analyzing 500 nodes and having a user model size of 100 terms is the optimal combination. In the ideal case, we would have evaluated all possible variations to find the single best variation. However, for some variables, there are up to 1,000 possible values, and combining all these variables and values leads to millions of possible variations. Evaluating this many variations was not feasible for us. The second best option would have been a multivariate statistical analysis to identify the impact of the single variables. However, also for such an analysis we did not have enough data. Therefore, our methodology was the third best option. It will not lead to a single optimal combination of variables, but as our result shows, our methodology leads to a significantly better algorithm than the baselines, and the results help understanding the factors that affect effectiveness in mind-map based user modeling.

We analyzed the effects of the variables for both CBF based on citations and CBF based on terms, and expected that the optimal values for the variables would differ for terms and citations. A “citation” is a reference or link in a mind-map to a research paper. In Figure 1, nodes with a PDF icon, link to a PDF file, typically a research article. If such a link exists, this is seen as a citation for the linked research article. A citation is also made when a user added bibliographic data, such as title and author to a node. Articles with the same title are assumed to be identical. This assumption might not always be true, but it should be effective for most cases, and we had not the resources to implement a sophisticated document disambiguation approach.

Matching user models with recommendation candidates was the same for both terms and citations. The user model, consisting of terms or citation IDs, was sent to Lucene. Lucene returned those research papers that were most relevant for the terms or citations (relevance was calculated with Lucene’s default algorithm). For more details refer to [Beel, Langer and Gipp 2014].

For the term-based CBF variations, all reported differences are statistically significant (p<0.05), if not reported otherwise in the text. Significance was calculated with a two tailed $t$-test and $\chi^2$ test where appropriate. Results for citation based CBF are mostly not statistically significant, because the approach was implemented only a few months ago, and not all users cite research papers in their mind-maps. Therefore, insufficient citation-based recommendations were delivered to produce significant results. Consequently, the focus of this paper lies on the term-based CBF variations. We also report runtimes in the charts for informative reasons, but do not discuss the data in this paper. It should be noted that runtimes could significantly differ with different implementations, or on different hardware. Overall, runtimes are rather long. This is caused by recording many statistics and running some other services on the recommendation server [Beel, Langer and Gipp 2014].

4. RESULTS

4.1 Mind-Map & Node Selection

4.1.1 Mind-Map Selection

When utilizing mind-maps for user modeling, one central question is which mind-maps to analyze, and which parts of the mind-maps to analyze. We experimented with a few variables to answer this question.
We hypothesized that analyzing all mind-maps of a user is not the most effective approach. If too many, or too old mind-maps are analyzed, this could introduce noise in the user model. To test this hypothesis, Docear’s recommender system randomly used the 𝑥 most recently modified mind-maps, regardless of when they were modified. An initial analysis shows that there is a slight tendency that CTR increases, the more mind-maps are utilized (Figure 5). When only a user’s most recent mind-map is utilized, CTR is 4.52% on average. Utilizing eight or nine mind-maps resulted in the highest CTR (6.82%). However, these results might be misleading since the analysis is based on recommendations for all users including those who created only few mind-maps. For users who only created e.g. two mind-maps, it would not be possible to analyze the eight or nine most recently created mind-maps. Therefore, we did the same analysis for users who created at least eight mind-maps (Figure 6). In this analysis, no statistically significant difference could be found for the number of utilized mind-maps. Judging by these numbers, it seems that the number of the most recently modified mind-maps is not an effective variable to optimize the user modeling process.

4.1.2 Node Selection

As an alternative to using the 𝑥 most recently modified mind-maps, Docear analyzed the 𝑥 most recently modified nodes. For example, if 𝑥=50, the terms (or citations) that are contained in the 50 most recently modified nodes were used for user modeling. The intention is that users might be working in different sections of several mind-
maps, and only those actively edited sections are relevant for user modeling. The analysis shows that the more nodes are used, the higher CTR becomes (Figure 7). While CTR is 3.66% on average when 1 to 9 nodes are used, CTR increases to 6.60% when 1,000 and more nodes are used. Interestingly, it is the opposite for citations: the more nodes with citations are used, the lower CTR becomes.

![Figure 7: CTR by the number of nodes to analyze (over all users)](image)

However, these results, again, might be misleading since not all users have created e.g. thousand nodes. Consequently, the CTR for e.g. one to nine nodes includes recommendations for all users, but the CTR for analyzing 1,000 or more nodes only considers recommendations to users who created at least 1,000 nodes. Therefore, we performed the previous analysis again, but for users who have created at least 1,000 nodes (Figure 8). This time, a saturation appears. When the 50 to 99 most recently created nodes are used, CTR is highest (7.50%). When more nodes are analyzed, CTR decreases. We did the same analysis for other user groups, and results were always the same – using only the 50 to 99 most recently modified nodes led to the highest CTRs on average. With regard to citations the results slightly change. For users with 1,000 or more nodes, using the most recent 10 to 49 citations is most effective (8.36% vs. 7.32% for using 1 to 9 citations).

![Figure 8: CTR by the number of nodes to analyze (for users with 1,000+ nodes)](image)

Selecting a fix number of nodes might not be the most effective criteria. The most recent, for example, 100 nodes could include nodes that were modified some years ago. Such nodes would probably not represent a user’s current information needs any
more. Therefore, Docear's recommender system randomly used all nodes that were modified within the past \( x \) days (Figure 9). When the recommender system utilized only those nodes that were modified on the current day, CTR was 3.81\% on average\(^5\). When nodes from the last two or three days were utilized, CTR increased to 5.52\%. CTR was highest, when nodes modified during the past 61 to 120 days were used (7.72\%), and remained high when nodes of the past 121 to 180 days were used. When nodes were used that were modified more than 180 days ago, CTR began to decrease. Apparently, the interests of Docear's users change after a few months.

Another variable we tested was the node modification type (Figure 10). The recommender system chose randomly, whether to utilize only nodes that were newly created, nodes that were moved, nodes that were edited, or nodes with any type of modification (created, edited, or moved). Utilizing moved nodes only, resulted in the highest CTR (7.40\%) on average, while the other modification types achieved CTRs around 5\%.

\(^5\) The analysis was done only for users being registered since at least 360 days
We find this interesting, because this result indicates that the evolution of a mind-map might be important for user modeling, and certain actions (e.g. moving nodes) indicate a higher importance of certain nodes. We have only experimented with a binary selection scheme, i.e. selecting or not selecting a node based on actions such as editing or moving nodes. It might also be interesting to weight nodes based on the user's action. For instance, the user modeling process could utilize all nodes but weight moved nodes stronger than newly created nodes.

Most mind-mapping tools allow folding a node, i.e. to hide its children. In Docear, this is indicated by a circle at the end of node (Figure 1). We hypothesized that nodes that are hidden, are currently not relevant for describing the user's information needs. Therefore, Docear's recommender system randomly chose whether to use only visible nodes, invisible nodes, or all nodes. When using visible nodes only, CTR increased from 6.00% (analyzing all nodes) to 7.61% (Figure 11). Using only invisible nodes led to a CTR of 4.89% on average. This indicates once more that by selecting a few meaningful nodes, a better effectiveness can be achieved than by examining simply all nodes.

![Figure 11: Node visibility as selection criteria (at least 100 nodes analyzed)](image)

**4.1.3 Node Extension**

We hypothesized that the relation among nodes is important. This means the terms of a node might be more meaningful when the node's context is known with regard to the neighboring nodes. The most common relationships in mind-maps are parents, children, and siblings. For instance, in Figure 1, the author's information needs seem rather vague when looking only at one node, e.g., “Google Scholar indexed invisible text”. In combination with the (grand) parent “(Academic) Search Engine Spam” the author's interests become clearer. Therefore, we experimented with extending the original selection of nodes. After the system chose the relevant nodes to examine with one of the previously introduced methods, the recommender system randomly chose whether to add siblings, parents, or children to the original selection.

Adding siblings resulted in a CTR of 5.76% compared to 5.12% for not adding siblings (Figure 12). Adding parent-nodes decreased CTR to 5.38% from 5.50% for not adding parent-nodes. Adding children increased CTR from 5.27% to 5.61%. All differences are small but statistically significant. In addition, when the recommender system combined all factors, i.e. adding siblings and children but ignoring parents, CTR was 6.18% on average, which was a significant improvement compared to not extending nodes (4.84%). One might suspect that extending the original node selection was only more effective because the extension caused more nodes to be used and the more nodes are used, the higher CTR becomes in general (see section 4.1.2).
However, this suspicion is not correct. For instance, when 100 to 499 nodes were selected, and no children or siblings were added, CTR was 5.15% on average. When, 10 to 50 nodes were selected and after adding children and siblings 100 to 499 nodes were used, CTR was 5.45%. This shows that selecting a few recently modified nodes, and extending them with their siblings and children, is more effective than selecting simply more nodes based only on the modification date.

4.2 Node & Feature Weighting

Often, user modeling applications weight features (e.g. terms) that occur in a certain document field (e.g. in title) stronger than features occurring in other document fields (e.g. the body text). Mind-maps have no fields for title, abstract, headings, or body text. Instead, mind-maps have nodes. Nodes have a certain depth, which is their distance from the root node. We hypothesized that the depth of a node might indicate the importance of the node. For instance, in Figure 1, the node “Scopus” has a depth of 2, and we would assume that the term “Scopus” describes the user’s interests with a different accuracy than the node “Academic Search Engines” that has a depth of 1.

To test the hypothesis, Docear’s recommender system randomly chose whether to weight terms of a node stronger or weaker, depending on its depth. If the nodes were to be weighted stronger the deeper they were, the weight of a node (1 by default) was multiplied with a) the absolute depth of the node; b) the natural logarithm of the depth; c) the logarithm to base 10 of the depth; or d) the square root of the depth. If the resulting weight was lower than 1, e.g., for ln (2), then the weight was set to 1. If nodes were to receive less weight the deeper they were, then the original weight of 1 was multiplied with the reciprocal of the metrics a) – d). If the resulting weight was larger than 1, e.g., for ln(2), the weight was set to 1. In the following charts, we provide CTR for the mentioned metrics. However, the differences among the metrics are not statistically significant. Hence, we concentrate on comparing the overall CTR, i.e. the CTR of weighting nodes stronger or weaker the deeper they are regardless of the particular metric.

The results show that when nodes are weighted stronger the deeper they are in a mind-map, CTR increases (Figure 13). Weighting them stronger, led to a CTR of 5.62% on average, while weighting them weaker led to a CTR of 5.15% on average.
We also experimented with other metrics that are based on the number of children, the number of siblings, and the number of words contained in a node. It appears that weighting nodes stronger the more children a node has, increases CTR (Figure 14). Weighting them stronger led to a CTR of 5.17% on average, while weighting them weaker led to a CTR of 5.01%. However, the difference was statistically not significant. Weighting based on the number of siblings had a significant effect (Figure 15). Weighting nodes stronger the more siblings they have led to a CTR of 5.41%, compared to 5.01% for weighting them weaker. Weighting nodes based on the number of terms they contained led to no significant differences (Figure 16).

After the individual weights are calculated, the weights need to be combined into a single node weighting score. We experimented with four different schemes to combine the scores. The most effective scheme was using the sum of all individual scores (6.38%). Using only the maximum score (max), multiplying the scores (product) or using the average score (avg) led to CTRs slightly above 5% (Figure 17).
After nodes are weighted, the nodes’ features inherit the weight of the nodes and are additionally weighted with a randomly chosen weighting scheme. This means, if a node has a weight of 8, then all terms (or citation) of that node receive a weight of 8, and this weight was multiplied with one of the following weighting schemes: plain frequency ($TF$-Only), $TF-IDF$, and a novel metric that we call $TF-IDuF$. $TF-IDuF$ is similar to $TF-IDF$ but based on the inverse document frequency in the user’s document corpus, instead of the standard document corpus. Hence, $TF-IDuF$, weights a term stronger the more often it occurs in the user’s mind-maps (or nodes) that are

---

$^6$ $TF-IDuF$ was only calculated for terms, not yet for citations.
currently selected for user modeling, but the less mind-maps (or nodes) of the user contain this term. The rationale is that when users being using a term frequently that they did not use frequently before, this term is of particular importance for the user model. In addition, if users are using terms for a longer time, they probably have already received recommendations for that term.

Figure 18: CTR of different weighting schemes

TF-IDF is commonly found to be more effective than TF-only [Manning et al. 2009]. Our analysis confirms this well-known finding – TF-IDF outperformed TF-only for terms with a CTR of 5.07% vs. 4.13% (Figure 18). However, to the best of our knowledge, it has not been empirically shown that TD-IDF is superior to TF-only when applied to citations, i.e. CC-IDF [Bollacker et al. 1998]. CC-IDF is frequently applied by research paper recommender systems to weight citations [Beel and Langer 2014b]. Hence, empirical evidence for the effectiveness of CC-IDF for citations is important. In Docear’s recommender system, CC-IDF led to lower CTRs (5.80%) than TF-only (6.07%) when applied to citations. We find this result surprising and can only speculate about the reason. One explanation might be the following: The rationale behind IDF for weighting terms is that terms occurring in many documents of the corpus (e.g. the, and, he, she, etc.), do not describe the content of the documents well. This rational seems plausible to us. However, the rationale does not necessarily applies for citations. Citations occurring in many documents of the corpus might still describe the citing document well, maybe even better than little cited papers. For instance, this paper cites, among others, Bollacker et al. [1998] and Said et al. [2013]. Bollacker et al. [1998] write about research paper recommender systems and their paper received more than 300 citations according to Google Scholar. This means, many papers in the corpus contain a citation to Bollacker et al. [1998]. Said et al. [2013] write about news recommendations and received only four citations. The paper from Bollacker et al. [1998] is certainly more relevant to describe our paper than Said et al. [2013]. Hence, CC-IDF would have led to suboptimal results when weighting the two papers. Of course, this is only one example, and there might be other examples in which CC-IDF were to prefer over TF-only. Further research is necessary to explore this issue.

Our novel metric TF-IDuF was slightly less effective than TF-IDF for terms but more effective than TF-only (Figure 18). When we repeated the analysis for those user-modeling processes that analyzed at least 500 nodes, TF-IDuF became slightly more effective than TF-IDF (Figure 19). This shows that the use of terms within a user’s “personal corpus” may be an important measure about a term’s relevancy, in particular when the personal corpus is large. We believe that further research is
necessary to explore the potential of TF-IDuF. Probably, TF-IDuF is particularly interesting when there is no access to the global corpus, and hence TF-IDF cannot be calculated. A combination of TF-IDF and TF-IDuF might also be interesting.

![Figure 19: CTR of different weighting schemes when 500 or more nodes were analyzed](image)

### 4.3 User Model Size

Just because utilizing e.g. the 50 most recently moved nodes is most effective, does not mean that necessarily all features of these nodes need to be stored in the user model. Therefore, Docear’s recommender system randomly selected to store only the x highest weighted features in the user model. For user modeling based on at least 50 nodes, CTR is highest (8.39%) for user models containing the 26 to 50 highest weighted terms (Figure 20). User models containing less, or more, terms achieve significant lower CTRs. For instance, user models with one to five terms have a CTR of 2.19% on average. User models that contained more than 500 terms have a CTR of 4.84% on average. Interestingly, CTR for citations continuously decreases the more citations a user model contains. Consequently, a user model size between 26 and 50 seems most sensible for terms, and a user model size of ten or less for citations.

![Figure 20: CTR by user model size (feature weight not stored)](image)

The previous analysis was based on un-weighted lists of terms and citations, i.e. the user model contained only a list of the features without any weight information.

---

The comparatively high CTR for user models with 501 and more citations is statistically not significant.
If the weights were stored in the user model, and used for the matching process, the picture changes (Figure 21). In this case, CTR has a peak for user models containing between 251 and 500 terms (8.13%). Interestingly, this CTR is similar to the maximum CTR for the optimal non-weighted user model size (8.81% for 26 to 50 terms). We find this surprising because we expected weighted lists to be more effective than un-weighted lists. The results for weighted citations are also surprising – CTR varies and shows no clear trend. We have no explanation for the results and believe that further research is necessary.\(^8\)

![CTR by user model size (feature weight stored)](image)

4.4 Additional observations

During our research, we made a few observations that do not necessarily relate to mind-map specific user modeling, but that might be interesting for the general recommender system community.

![CTR based on the recommendation’s original rank](image)

To provide greater recommendation variety to users, the ten final recommendations were randomly chosen from the top 50 recommendation candidates. This increases variety but decreases CTR on average (Figure 22). Those recommendations that were originally among the top 10 candidates, achieved a CTR

\(^8\) We double-checked all data and the source code, and are quite confident that there are no flaws in the user modeling process.

ACM Transactions on xxxxxxx, Vol. xx, No. xx, Article xx, Publication date: Month YYYY
of 5.81% on average. For lower ranked recommendations, CTR continuously decreased down to 4.58% for recommendations that were among the top 41 to 50 candidates. For citation-based recommendations, the trend was similar.

Before Docear displays the finally chosen recommendations, the recommendations are shuffled. This allowed us to analyze the effect that the display-rank of a recommendation has on CTR (Figure 23). Those recommendations ranked at position 1 had the highest CTR on average (6.77%), while recommendations shown in the middle of the list had the lowest CTR (4.40% on position 5). For the last positions, CTR again increased a little bit (5.34% for position 10). This means, just the recommendation rank made a difference in CTR of up to 50% (6.77% vs. 4.40%). The discovery that the recommendation rank affects CTR is not new, but, to the best of our knowledge, it has not been empirically quantified in the domain of research paper recommender systems.

![CTR based on the rank at which a recommendation was displayed](image)

We observed that when Lucene returned less than 1,000 recommendation candidates for a particular user model, the CTR tended to be lower as if 1,000 or more candidates were returned, in particular for term-based citations (Figure 24). For 1,000 or more recommendation candidates, CTR was 5.13% on average. In contrast, when 50 to 99 candidates were returned, CTR was 1.82% on average. This means, the overall CTR could be increased when user models that lead to less than 1,000 recommendation candidates are discarded, and a new user model is created instead. We also observed that CTR was comparatively high, when Lucene returned less than ten recommendation candidates. While this might seem surprising on first glance, there is a plausible explanation. If less than ten candidates are returned, then less than ten recommendations can be displayed. However, the less recommendations are shown in general, the higher the CTR tends to be. For citation-based recommendations, the maximum number of recommendation candidates never was above 1,000.
Throughout the paper, citation-based recommendations seemed to be more effective than term-based recommendations: Based on all delivered recommendations, citation-based recommendations had an average CTR of 6.11%, while term-based recommendations had an average CTR of 5.10% (Figure 25). However, this comparison is misleading. Citation-based recommendations are only possible if users have at least one citation in their mind-maps. Consequently, novel users, without citations in their mind-maps receive only term-based recommendations. These users tend to have lower CTRs than users with more comprehensive mind-maps. In addition, citation-based user modeling often returns less than ten recommendation candidates. Therefore, we made a “fair” comparison and compared citation- and term-based user models for which at least 10 recommendation candidates are returned; the original rank of the recommendation candidates was between one and ten; and the users had made at least one citation in their mind-map. In this comparison, term-based recommendations outperform citation-based recommendations with a CTR of 6.57% vs. 5.25%.

4.5 Mind-Maps vs. Other Items
Our main goal was to build a recommender system that considers the unique characteristics of mind-maps. However, we were also interested to explore how user modeling based on mind-maps compares to user modeling based on other items, research papers in particular. Therefore, we evaluated seven approaches that utilized
(a) terms of nodes in the users’ mind-maps (b) terms from the titles of the users’ PDF files (c) terms from the titles of the user’s citations (d) terms from the titles of the PDFs and citations, (e) terms from the nodes of mind-maps, and titles of citations, f) terms from the nodes of mind-maps, and titles of PDF files g) terms from all mentioned sources.

The highest CTR was achieved for utilizing terms from the cited paper titles (7.14%). Utilizing terms from the titles of all PDFs that users had linked in their mind-maps led to a CTR of 5.63% on average. Utilizing terms from the mind-maps’ nodes led to a CTR of 5.21% on average (Figure 26). The other approaches also achieved CTRs around 5% and 6%. We would not conclude that titles of citations are generally more effective than nodes in mind-maps. The approaches were all rather simple. With appropriate enhancements, all approaches probably could perform more effectively (in the next section we show that a mind-map specific user modeling approach is twice as effective as the simple node approach). However, the analysis indicates that mind-maps are in the same league for user modeling as are research papers since CTRs are comparable. This encourages us to believe that developers of mind-mapping tools should integrate recommender systems in their tools, and that these recommender systems will achieve similar performances as recommender systems in other domains.

5. DOCEAR’S MIND-MAP SPECIFIC USER MODELING APPROACH

The evaluation made clear that numerous variables influence the effectiveness of mind-map based user modeling. We combined the optimal values of these variables in a single algorithm as follows: The algorithm used the 75 most recently moved nodes from the past 90 days that were visible. If less than 75 moved and visible nodes were available, then the up to 75 most recently modified nodes from the past 90 days were used instead. The selected nodes were extended by all their children and siblings. All nodes were weighted based on the nodes’ depth and number of siblings (we used the ln weighting and summed the individual scores). The terms of these nodes were additionally weighted with TF-IDuF (stop-words were removed). The 35 highest weighted terms were stored in the user model as un-weighted list. This user model was used for the matching process with the recommendation candidates.

We compared our algorithm against four baselines. The four baselines were the stereotype approach (cf. section 2), and the three “obvious” CBF approaches (using terms from a single node, from a single mind-map, or from all mind-maps; see section 1).
Among the baselines, using all terms from all mind-maps (CTR of 3.96%) and from a single mind-map (CTR of 4.12%) performed alike (Figure 27). Using terms from only one node – the approach that MindMeister applied – resulted in the lowest CTR (1.05%). Stereotype recommendations performed reasonable with a CTR of 3.52%. Overall, CTR of the baselines tends to be lower than in our previous evaluation [Beel, Langer, Genzmehr, et al. 2014]. However, since our previous evaluation, we added several new variables, and some might have decreased CTR on average. The reasonable effectiveness of stereotype recommendations seems surprising, considering how rarely used this approach is in the recommendation community. Nevertheless, the result is plausible. Most of Docear’s users are researchers and therefore they should be interested in books about academic writing, and hence click the corresponding recommendations.

Docear’s mind-map specific user modeling algorithm significantly outperformed all baselines and achieved a CTR of 9.82% (Figure 27). This is more than twice as high as the best performing baseline and nearly 10 times as high as MindMeister’s approach, the only approach that had been applied in practice thus far. Because we experimented only with a few variables, and the experiments were of relative basic nature, we are convinced that more research could further increase the effectiveness.

6. DISCUSSION & SUMMARY

We explored the effectiveness of user modeling based on mind-maps. Our goal was to learn whether this is a promising field of research, and whether users could benefit from mind-map specific user modeling systems. We examined how effective standard user modeling approaches are when applied to mind-maps, and how to enhance these approaches by taking into account the characteristics of mind-maps. We implemented a mind-map based research paper recommender system, and integrated it into our mind-mapping software Docear. The recommender system displayed 431,112 recommendations to 4,701 users from March 2013 to August 2014, and recommendations were created with several variations of content-based filtering (CBF), of which some considered different characteristics of mind-maps.

The evaluation of the different user modeling approaches revealed the following results.

First, standard user modeling approaches can be reasonably effective when applied to mind-maps. However, the effectiveness varied depending on which standard approach was used. When user models were based on all terms of users’
mind-maps, the click-through rate (CTR) was around 4%. When only terms from the most recently modified node were used, CTR was 1.05%. These results led us to the conclusion that user modeling based on mind-maps is not trivial, and minor differences in the approaches lead to significant differences in the effectiveness of the user modeling.

Second, user modeling based on mind-maps can achieve significantly higher CTRs when the characteristics of mind-maps are considered. Based on our research, the following variables should be considered: a) the number of analyzed nodes. It seems that the terms of the most recently modified 50 to 99 nodes are sufficient to describe the users’ information needs. Using more, or less, nodes decreased the average CTR b) Time restrictions were important. It seems that utilizing nodes that were created more than four month ago, decreased CTR. c) CTR increased when only nodes were used that were recently moved by a user, instead of using nodes that were created or edited. d) Using only nodes that are visible in the mind-map also increased effectiveness compared to using both visible and invisible nodes. e) Extending the originally selected nodes by adding siblings and children increased average CTR slightly but statistically significantly. This indicates that the full meaning of nodes becomes only clear when their neighbor nodes are considered. f) We also found that weighting nodes, and their terms, based on node depth and the number of siblings increased CTR. The deeper a node, and the more siblings it has, the more relevant are its terms to describe the users’ information needs. The separate weights should be combined by their sum. g) The final user model should contain the highest weighted 26 to 50 terms, if the user model is stored as un-weighted list. If weights are stored, it seems that larger user models are sensible. However, more research is needed to clarify this.

Third, when the variables were combined in their favorable way, this mind-map specific user modeling approach outperformed standard user modeling approaches applied to mind-maps by a factor of over 2 (CTR of 9.82% vs. 4.12%). Compared to the approach that was applied in practice by MindMeister (using only the last modified node), our approach increased effectiveness by a factor of nearly 10 (CTR of 9.82% vs. 1.05%).

Fourth, there are several results beyond mind-map specific user modeling. We introduced a new weighting scheme (TF-IDuF) that seems equally effective as TF-IDF, and might be combined with TF-IDF. We were first who empirically compared CC-IDF, i.e. TF-IDF applied to citations, with plain citation frequency (TF-only). The results indicate that CC-IDF might be less effective than a simple TF-only measure. However, more research is needed to clarify this. In the domain of research paper recommender systems, the finding that a user model size should be between 26 and 50 terms is also novel, as well as the finding that researchers’ interests shift after about four month might also prove useful for other research paper recommender systems. To the best of our knowledge, it has also not been shown that the recommendation rank can affect CTR by up to 50%

Our research has a few limitations. So far, the values for the variables are only rough suggestions. For instance, the finding that the optimal user model size is between 26 and 50 terms is still rather vague. There are also more potential variables that we have not yet analyzed. For instance, the evolution of mind-maps over time might enhance the effectiveness of mind-map specific user modeling. We could imagine that weighting nodes by the intensity of use (e.g. how often a node was edited, opened, or moved) might provide valuable information. We also advocate research on the differences of content and the structure of mind-maps that were
created for different purposes, such as brainstorming or literature management. This might provide valuable insights on the characteristics of mind-maps. More research is needed to explore dependencies among the variables. This requires more advanced statistical analyses of the variables. This, however, requires research in large-scale recommender systems with significantly more users than Docear has. It should also be noted that our research was based only on Docear, which is a unique mind-mapping software, because it focuses on researchers. Additional research with other mind-mapping tools seems desirable. This is particularly true because most mind-mapping tools focus on certain groups of users and it would be interesting to explore whether there is one mind-map specific user modeling approach that suits all mind-mapping applications, or whether each application needs to apply a different approach. Most of our results with regard to citations were statistically not significant. It would also be interesting to research in more detail how citations, or hyperlinks, could be exploited. In addition, we only evaluated the algorithms with CTR. For future research, user studies might also be desirable to evaluate the algorithms.

Overall, the results of our research reinforced our astonishment that mind-maps are being disregarded by the user modeling and recommender system community. We believe that this paper showed the potential of mind-map specific user modeling, and we hope that it initiates a discussion, which encourages other researchers to do research in this field. We believe there is a lot of interesting work to do that could further increase the effectiveness of mind-map specific user modeling. We also hope that our results encourage developers of mind-mapping tools to integrate recommender systems in their software. The results of our paper will help to implement a decently effective user modeling approach. This would benefit the mind-mapping users or, in case of personalized advertisement, generate revenues that are presumably higher than those that could be achieved with the user modeling approaches of MindMeister and Mindomo.

**FINAL NOTE**

All data that we used for our analysis is publicly available [Beel, Langer and Gipp 2014]. The data allows replicating our calculations, and performing new analyses beyond the results that we presented in this paper. To foster further research on mind-map specific user modeling, we invite other scientists to join us, and cooperate on the development and research of Docear’s recommender system.

**REFERENCES**


ACM Transactions on xxxxxxxx, Vol. xx, No. x, Article x, Publication date: Month YYYY


