

# Exploring the Potential of User Modeling based on Mind Maps

Joeran Beel<sup>1</sup>, Stefan Langer<sup>1</sup>, Georgia Kapitsaki<sup>2</sup>, Corinna Breitingner<sup>1</sup>, Bela Gipp<sup>3</sup>

<sup>1</sup> Docear, Magdeburg, Germany  
{beel | langer | breitingner}@docear.org

<sup>2</sup> University of Cyprus, Department of Computer Science, Nicosia, Cyprus  
gkapi@cs.ucy.ac.cy

<sup>3</sup> University of Konstanz, Department of Computer and Information Science, Konstanz, Germany  
bela.gipp@uni-konstanz.de

**Abstract.** Mind maps have not received much attention in the user modeling and recommender system community, although mind maps contain rich information that could be valuable for user-modeling and recommender systems. In this paper, we explored the effectiveness of standard user-modeling approaches applied to mind maps. Additionally, we develop novel user modeling approaches that consider the unique characteristics of mind maps. The approaches are applied and evaluated using our mind mapping and reference-management software Docear. Docear displayed 430,893 research paper recommendations, based on 4,700 user 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 1.16% and 3.92%. However, when adjusting user modeling to the unique characteristics of mind maps, a higher CTR of 7.20% could be achieved. A user study confirmed the high effectiveness of the mind map specific approach with an average rating of 3.23 (out of 5), compared to a rating of 2.53 for the best baseline. Our research shows that mind map-specific user modeling has a high potential, and we hope that our results initiate a discussion that encourages researchers to do research in this field and developers to integrate recommender systems to their mind mapping tools.

**Keywords:** mind map, user modeling, recommender systems

## 1 Introduction

Mind mapping is a visual technique to record and organize information, and to develop new ideas. As such, mind-maps are used for tasks like brainstorming, knowledge management, note taking, presenting, project planning, decision-making, and career planning. Originally, mind mapping was performed using pen and paper, but since the 1980s, more than one hundred software tools for aiding users in creating mind maps have evolved. These tools are used by an estimated two million users who create around five millions mind-maps every year [1]. Mind-maps received attention in various research fields. They have been used to implement a lambda calculator [2], to filter search results from Google [3], to present software requirements [4], to research how knowledgeable business school students are, and there are numerous studies about the effectiveness of mind-maps as a learning tool [5]. In the field of document engineering and text mining, mind maps were created automatically from texts [6], and have been utilized to model XML files [7].

In the field of user-modeling and recommender-systems research, mind maps have thus far received no attention. However, mind-maps typically contain structured information that reflects the interests, knowledge, and information needs of their users. In this regard, the content of mind-maps is comparable to the content of emails, web pages, and research articles [1]. Hence, we believe that mind maps should be equally well suited for user modeling as are other documents, such as emails [8], web-pages [9], and research articles [10].

We assume that the reader is familiar with the idea behind mind mapping, and provide only a brief example of a mind-map, which we created using the mind-mapping and reference-management software *Docear* (Figure 1). For more information about mind maps, please refer to [11]. The mind map in Figure 1 was created to manage academic PDFs and references. We created categories reflecting our research interests (“Academic Search Engines”), subcategories (“Google Scholar”), and sorted PDFs by category and subcategory. Docear imported annotations (comments, highlighted text, and bookmarks) made in the PDFs, and clicking a PDF icon opens the linked PDF file. Docear also extracts metadata from PDF files (e.g. title and journal name), and displays metadata when the mouse hovers over a PDF icon. A circle at the end of a node indicates that the node has child nodes, which are collapsed. Clicking the circle unfolds the node, i.e. make its child nodes visible.

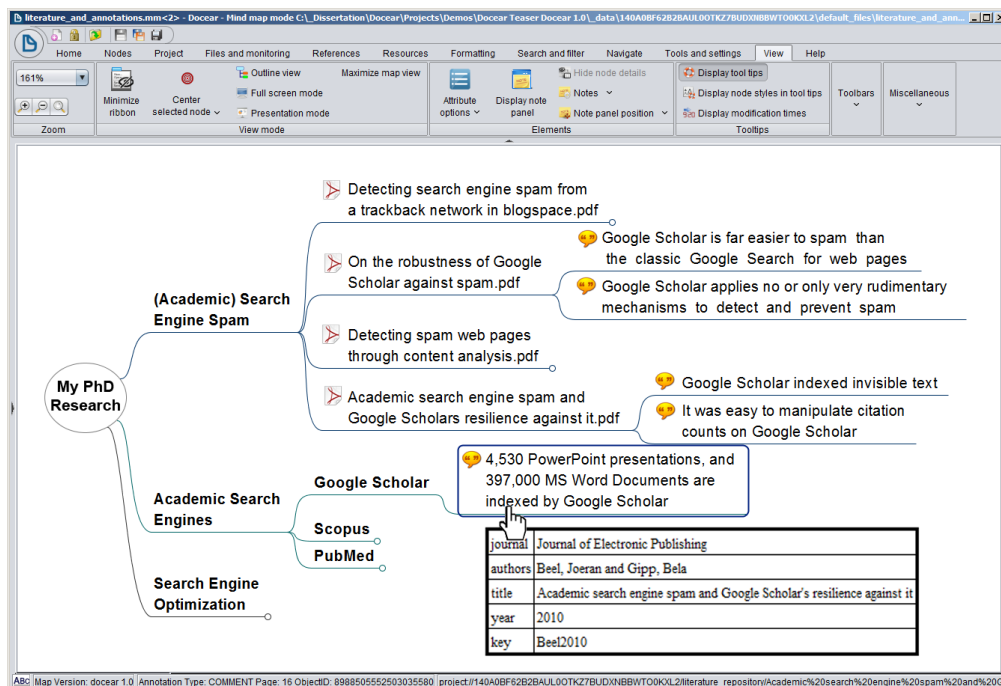


Figure 1: Screenshot of a mind-map in Docear

As mentioned, we are the first to research the potential of mind-maps for user modeling and recommender systems. However, in industry the companies *MindMeister* and *Mindomo* already have utilized mind-maps for user modeling, more precisely for personalized advertisements. MindMeister extracted terms from the node that a user created or edited most recently, and used these terms as the user model. MindMeister sent this user model, i.e. the terms, to Amazon’s Web Service as a search query. Amazon returned book recommendations that matched the query, and MindMeister displayed the recommendations in a window next to the mind-map.

Mindomo applied a similar concept in which Google AdSense was used instead of Amazon. Both companies have since abandoned their user modeling systems, although they still actively maintain their mind mapping tools. In an email, Mindomo explained that “people were not really interested” in the advertisements [1]. We were surprised about Mindomo’s statement, because we expected mind-maps to be an effective source for user modeling.

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 [1]. Instead of using Amazon’s Web Service or Google AdSense, we built our own corpus of recommendation candidates and used Apache Lucene to match candidates with user models. MindMeister’s approach, i.e. utilizing the terms from the most recently edited or created node, achieved click-through rates (CTR) between 0.2% and around 1%. Compared to other recommender systems [12], 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 approach is to

build user models based on terms contained in the users' *current* mind map. The third approach is to utilize terms from *all* mind maps a user ever created. We implemented the second and third approach as well [1]. 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 further be increased by adjusting the user modeling approach to the unique characteristics of mind-maps.

Therefore, our main research goal is to identify variables that influence the effectiveness of user-modeling approaches based on 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 [1].

## 2 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 implemented and evaluated only a few variables that we considered most promising, and for which an evaluation with Docear was feasible. 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 mind map-specific weighting schemes.

From March 2013 to August 2014, Docear's recommender system delivered 45,208 recommendation sets, with 430,893 recommendations to 4,700 users. 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. 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 [13].

Each set of recommendations 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, and stored the 25 highest weighted terms in the user model. Another algorithm might have used the 250 most recently modified nodes (*visible and invisible*) among all mind-maps, weighted the *citations* of these nodes, 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), which describe the ratio of delivered recommendations by those that were clicked. 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. There is a lively discussion about the meaningfulness of CTR and online evaluations and its alternatives, i.e. offline evaluations and user studies. 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 [14]. 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 this mind-map-specific user modeling performed better than the baselines. Effectiveness was again measured with CTR but we additionally conducted a user study based with 182 users who rated 491 recommendation sets on a five-star scale.

One baseline was the stereotype approach. Stereotyping is one of the earliest user modeling and recommendation classes [15]. In stereotyping, one assumes that users will like what their peer group is enjoying. For Docear we generalized that all Docear users are researchers, and that all researchers would be interested in literature about academic writing. Hence, when the stereotype approach was 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 baselines were those CBF variations that are rather obvious and that we already used in our initial study [1]: 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, since 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.

For the term-based CBF variations, all reported differences are statistically significant ( $p < 0.05$ ), if not reported otherwise. 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.

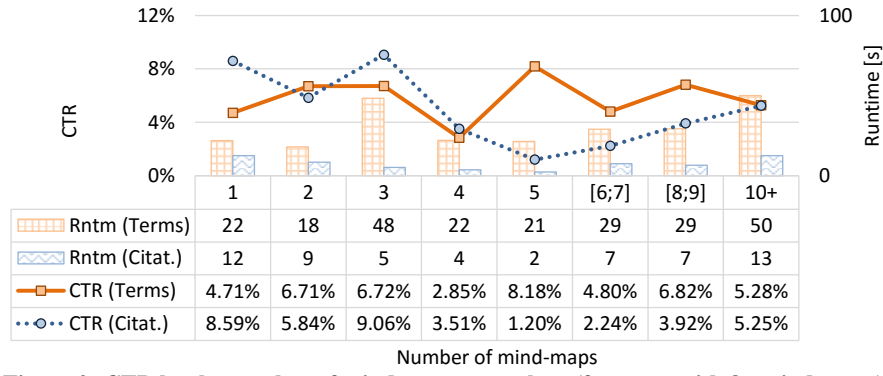
**Please note:** Most of the data that we used for our analysis is publicly available [13]. The data should allow replicating our calculations, and performing new analyses beyond the results that we presented in this paper. To foster further research on mind map-specific user modeling, we invite other scientists to join us, and cooperate on the development and research of Docear’s recommender system.

### 3 Results

#### 3.1 Mind-Map & Node Selection

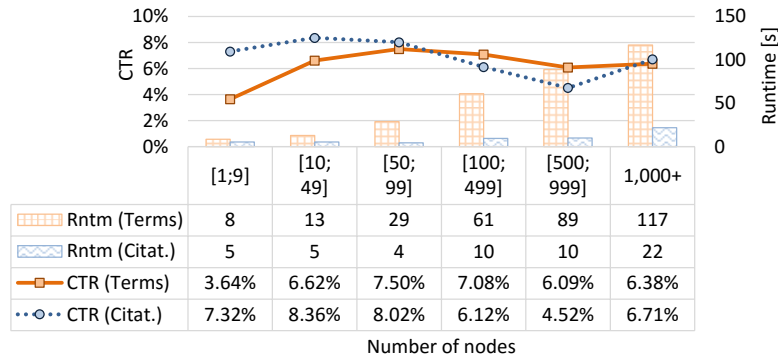
When utilizing mind maps for user modeling, one central question is *which mind maps* to analyze, and *which parts* of the mind maps to analyze. We experimented with a few variables to answer this question.

We hypothesized that analyzing all mind maps of a user is not the most effective approach. If the mind maps analyzed are too many or too old, this could introduce noise into 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. For users who created at least eight mind maps, no statistically significant difference could be found for the number of utilized mind maps (Figure 2). 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.



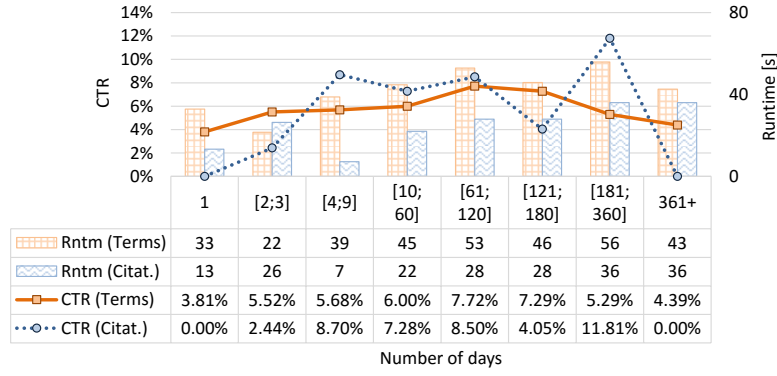
**Figure 2: CTR by the number of mind-maps to analyze (for users with 8+ mind-maps)**

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. For users who have created at least 1,000 nodes, saturation appears (Figure 3). 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).



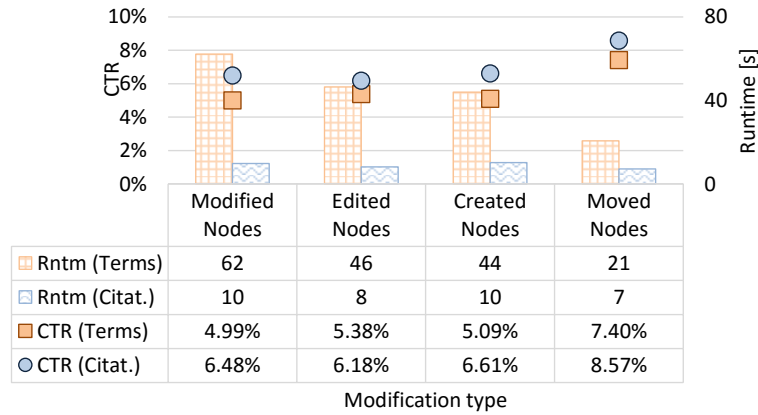
**Figure 3: CTR by the number of nodes to analyze (for users with 1,000+ nodes)**

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 4). When the recommender system utilized only those nodes that were modified on the current day, CTR was 3.81% on average for users who used Docear for at least 360 days. 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.



**Figure 4: CTR for nodes analyzed in the past  $x$  days (for users being registered 360+ days)**

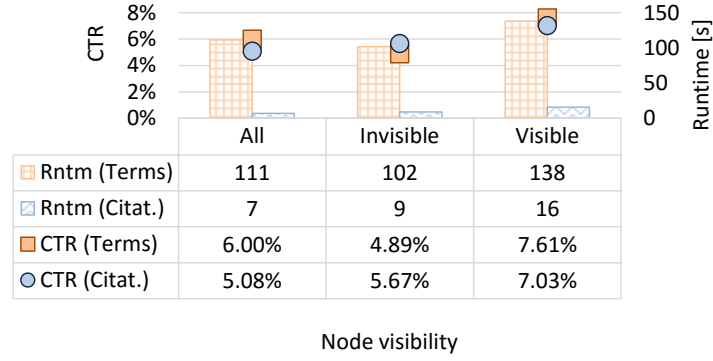
Another variable we tested was the node modification type (Figure 5). The recommender system chose randomly, whether to utilize only nodes that were newly *created*, nodes that were *moved*, nodes that were *edited*, or nodes with any type of *modification* (created, edited, or moved). Utilizing moved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



**Figure 5: Node modification type**

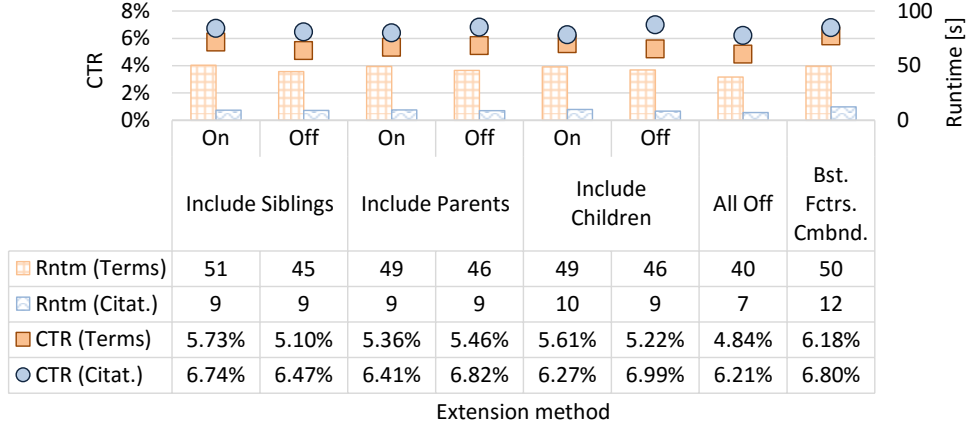
We find this interesting, because this result indicates that the evolution of a mind-map might be important for user modeling, and certain actions (e.g. moving nodes) indicate a higher importance of certain nodes. We have only experimented with a binary selection scheme, i.e. selecting or not selecting 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 need. 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 6). 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 6: Node visibility as selection criteria (at least 100 nodes analyzed)**

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.



**Figure 7: Extending the original node selection**

Adding siblings resulted in a CTR of 5.73% compared to 5.10% for not adding siblings (Figure 7). Adding parent-nodes decreased CTR to 5.36% compared to 5.46% for not adding parent-nodes. Adding children increased CTR from 5.22% 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. 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 indicates 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.

### 3.2 Node 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 fields (e.g. the body text). Mind maps have no fields for title, abstract, headings, or body text. Instead, mind maps have nodes, which have a certain *depth*, i.e. 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 this term describes the user’s interests with a different accuracy than the node “Academic Search Engines” that has a depth of 1.

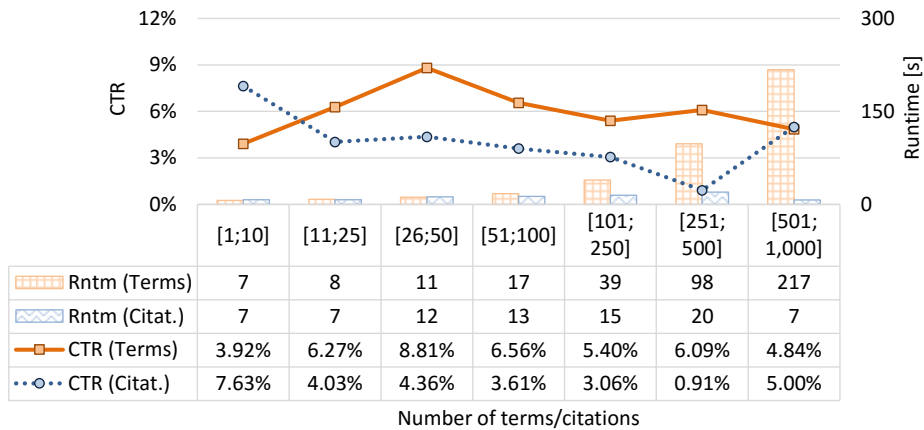
To test the hypothesis, Docear’s recommender system randomly chose whether to weight terms of a node stronger or weaker, depending on its depth. If the nodes were to be weighted stronger the deeper they were, the weight of a node (1 by default) was multiplied with **a)** the absolute depth of the node; **b)** the natural logarithm of the depth; **c)** the common logarithm 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.

Results show that when nodes are weighted stronger the deeper they are in a mind map, CTR increases. Weighting them stronger, led to a CTR of 5.61% on average, while weighting them weaker led to a CTR of 5.12% 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. Weighting them stronger led to a CTR of 5.17% on average, while weighting them weaker led to a CTR of 4.97%. However, the difference was statistically not significant. Weighting based on the number of siblings had a significant effect. Weighting nodes stronger the more siblings they have led to a CTR of 5.40%, compared to 5.01% for weighting them weaker. Weighting nodes based on the number of terms they contained led to no significant differences.

After the individual weights are calculated, the weights need to be combined into a single node weighting score. We experimented with four different schemes for combining the scores. The most effective scheme was using the *sum* of all individual scores (CTR = 6.38%). Using only the maximum score (*max*), multiplying the scores (*product*) or using the average score (*avg*) led to CTRs slightly above 5%.

### 3.3 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 (8.81%) for user models containing the 26 to 50 highest weighted terms (Figure 8). User models containing less, or more, terms achieve significant lower CTRs. For instance, user models with one to ten terms have a CTR of 3.92% 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<sup>1</sup>. 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.

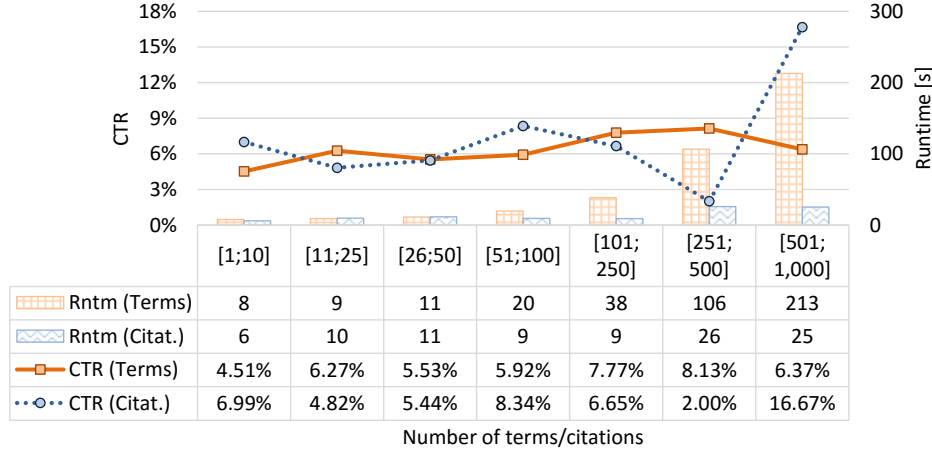


<sup>1</sup> The high CTR for user models with 501 and more citations is statistically not significant.



**Figure 8: CTR by user model size (feature weight not stored)**

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. If the weights were stored in the user model, and used for the matching process, the picture changes (Figure 9). 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 this result and believe that further research is necessary<sup>2</sup>.



**Figure 9: CTR by user model size (feature weight stored)**

### 3.4 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 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, and the three “obvious” CBF approaches (using terms from a single node, from a single mind map, or from all mind maps).

Among the baselines, using all terms from all mind maps and from a single mind map performed alike in terms of CTR (Figure 10) and ratings. Using terms from only one node – the approach that MindMeister applied – resulted in the lowest CTR (1.16%) and ratings (1.63). Stereotype recommendations performed comparably reasonable with a CTR of 3.08% and a rating of 1.97 on average. Overall, CTR of the baselines tends to be lower than in our previous evaluation [1]. However, since our previous evaluation, we added several new variables, and some might have decreased CTR on average. In addition, Docear’s Web Spider was not running in the past couple of months. This means, no new recommendation candidates were added to the corpus. Hence, long-time users probably often received recommendations they had received previously, which decreases average CTR. The reasonable effectiveness of stereotype recommendations might seem 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

<sup>2</sup> We double-checked all data and the source code, and are quite confident that there are no flaws in the user modeling process.

click the corresponding recommendations. Even though the ratings are not very high, we believe that further research about applying stereotype recommendations might be promising.

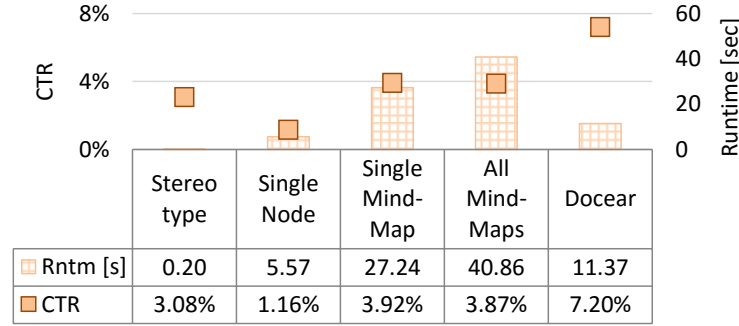


Figure 10: Docear's mind map-specific approach vs. Baselines (CTR and Runtime)

Docear's mind map-specific user modeling algorithm significantly outperformed all baselines and achieved a CTR of 7.20% on average (Figure 10). This is nearly twice as high as the best performing baseline and six times as high as MindMeister's approach, the only approach that had been applied in practice thus far. User ratings also show a significantly higher effectiveness for Docear's approach (3.23) than for the best performing baseline (2.53). 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.

## 4 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 430,893 recommendations to 4,700 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 1.16%. 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 months 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 and 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, the more siblings it has, and possibly the more children 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 nearly two (CTR of 7.20% vs. 3.92%). 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 six (CTR of 7.20% vs. 1.16%).

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. Hence, more research is required to specify the optimal values of the variables. There are also more potential variables that we have not yet analyzed but which might be promising. 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, which in turn requires research using large-scale recommender systems that have significantly more users than Docear. It should also be noted that our research was based only on Docear, which is a unique mind mapping software, because it focuses on academic users. 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 using 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 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 into their software. The results of our paper will help with implementing a suitable user modeling approach. This would benefit 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.

## References

- [1] J. Beel, S. Langer, M. Genzmehr, and B. Gipp, "Utilizing Mind-Maps for Information Retrieval and User Modelling," in *Proceedings of the 22nd Conference on User Modelling, Adaption, and Personalization (UMAP)*, 2014, vol. 8538, pp. 301–313.
- [2] L.-R. Chien and D. J. Buehre, "A Visual Lambda-Calculator Using Typed Mind-Maps," in *International Conference on Computer and Electrical Engineering*, 2008, pp. 250–255.
- [3] I. A. Zuolkernan, M. A. AbuJayyab, and Y. A. Ghanam, "An alignment equation for using mind maps to filter learning queries from Google," in *Advanced Learning Technologies, 2006. Sixth International Conference on*, 2006, pp. 153–155.
- [4] J. A. P. Contó, W. F. Godoy, R. H. Cunha, E. C. G. Palácios, A. L. Erario, A. L. Domingues, J. A. Gonçalves, A. S. Duarte, and J. A. Fabri, "Applying Mind Maps at Representing Software Requirements," *Contributions on Information Systems and Technologies*, p. 1, 2013.
- [5] B. Holland, L. Holland, and J. Davies, "An investigation into the concept of mind mapping and the use of mind mapping software to support and improve student academic performance," 2004.
- [6] R. Kudelic, M. Malekovic, and A. Lovrencic, "Mind map generator software," in *Proceedings of the 2nd IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, 2012, pp. 123–127.

- [7] A. Bia, R. Muñoz, and J. Gómez, "Using mind maps to model semistructured documents," in *Proceedings of the 14th European Conference on Digital Libraries (ECDL)*, 2010, pp. 421–424.
- [8] Q. M. Ha, Q. A. Tran, and T. T. Luyen, "Personalized email recommender system based on user actions," in *Simulated Evolution and Learning*, Springer, 2012, pp. 280–289.
- [9] M. Göksedef and S. Gündüz-Ögüdücü, "Combination of Web page recommender systems," *Expert Systems with Applications*, vol. 37, no. 4, pp. 2911–2922, 2010.
- [10] F. Zarrinkalam and M. Kahani, "SemCiR - A citation recommendation system based on a novel semantic distance measure," *Program: electronic library and information systems*, vol. 47, no. 1, pp. 92–112, 2013.
- [11] T. Buzan, *The Mind Map Book (Mind Set)*. BBC (BBC Active), 2006.
- [12] K. Hofmann, A. Schuth, A. Bellogin, and M. de Rijke, "Effects of Position Bias on Click-Based Recommender Evaluation," in *Advances in Information Retrieval*, Springer, 2014, pp. 624–630.
- [13] J. Beel, S. Langer, B. Gipp, and A. Nürnberger, "The Architecture and Datasets of Docear's Research Paper Recommender System," *D-Lib Magazine - The Magazine of Digital Library Research*, vol. 20, no. 11/12, 2014.
- [14] J. Beel and S. Langer, "A Comparison of Offline Evaluations, Online Evaluations, and User Studies in the Context of Research-Paper Recommender Systems," *Under Review. Pre-print available at <http://www.docear.org/publications/>*, 2014.
- [15] E. Rich, "User modeling via stereotypes," *Cognitive science*, vol. 3, no. 4, pp. 329–354, 1979.

# Additional Information

**Bibliographic Data** | Joeran Beel, Stefan Langer, Georgia M. Kapitsaki, Corinna Breitinger, and Bela Gipp, "Exploring the Potential of User Modeling based on Mind Maps" in Proceedings of User Modeling, Adaptation and Personalization – 23rd International Conference, UMAP 2015, Dublin, Ireland, 2015, pp. 3-17. DOI 10.1007/978-3-319-20267-9\_1.

**Original Article** | [http://link.springer.com/chapter/10.1007/978-3-319-20267-9\\_1](http://link.springer.com/chapter/10.1007/978-3-319-20267-9_1)

**Preprint** | [http://docear.org/2015-umap-mm\\_potential.pdf](http://docear.org/2015-umap-mm_potential.pdf)

**Related Work** | <http://docear.org/docear/research-activities/>

**Academic Platforms**



**Authors**

Joeran Beel



Stefan Langer



Georgia M. Kapitsaki



Corinna Breitinger



Bela Gipp



**BibTeX**

```
@INPROCEEDINGS{Beel2015f,  
  author = {Beel, Joeran and Langer, Stefan and Kapitsaki, Georgia M. and Breitinger, Corinna and Gipp, Bela},  
  title = {Exploring the Potential of User Modeling based on Mind Maps},  
  booktitle = {Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization (UMAP)},  
  year = {2015},  
  editor = {Francesco Ricci and Kalina Bontcheva and Owen Conlan and Séamus Lawless},  
  volume = {9146},  
  series = {Lecture Notes of Computer Science},  
  pages = {3-17},  
  publisher = {Springer},  
  doi = {10.1007/978-3-319-20267-9_1},  
}
```

**RefMan (RIS)**

```
TY - BOOK  
T1 - Exploring the potential of user modeling based on mind maps  
A1 - Beel, Joeran  
A1 - Langer, Stefan  
A1 - Kapitsaki, Georgia  
A1 - Breitinger, Corinna  
A1 - Gipp, Bela  
SN - 3319202669  
Y1 - 2015  
PB - Springer  
ER -
```

## EndNote

%O Book  
%T Exploring the potential of user modeling based on mind maps  
%A Beel, Joeran  
%A Langer, Stefan  
%A Kapitsaki, Georgia  
%A Breiting, Corinna  
%A Gipp, Bela  
%@ 3319202669  
%D 2015  
%I Springer