Validating Navigation Time Prediction Models for Menu Optimization

Vera Hollink, Maarten van Someren
Faculty of Science, University of Amsterdam
Kruislaan 419, 1098 VA Amsterdam, The Netherlands
{vhollink,maarten}@science.uva.nl

Abstract

Authors of menu optimization methods often use navigation time prediction models without validating whether the model is adequate for the site and its users. We review the assumptions underlying navigation time prediction models and present a method to validate these assumptions offline. Experiments on four web sites show how accurate the various model features describe the behavior of the users. These results can be used to select the best model for a new optimization task. In addition, we find that the existing optimization methods all use suboptimal models. This indicates that our results can significantly contribute to more effective menu optimization.

1 Introduction

Hierarchical navigation menus are a popular medium to allow users of web sites access to the site's contents. These menus consist of hierarchies of categories with the content pages located at the leaf nodes. To reach their target information users navigate top-down through the hierarchy by selecting categories.

The initial design of menus is often far from optimal as designers do not know the goals and strategies of their future users. Moreover, even with a good initial structure navigation can become less efficient when the user population or the contents of the site change over time.

Various authors have attempted to overcome these problems by presenting methods to automatically adapt the structure of a menu towards the site's actual user population, e.g. [Witten and Cleary, 1984; Fisher *et al.*, 1990], or to the behavior of individual users, e.g. [Smyth and Cotter, 2003; Hollink *et al.*, 2005]. These methods address the optimization of menus with a purely navigational function. In these menus the hierarchical structures do not provide information, but are only means to navigate to the content pages on the terminal nodes. Consequently, the optimal menu is the one that minimizes the average time users need to reach their target pages.

All menu optimization techniques involve adaptation of menu structures and evaluation of the adapted structures. The techniques define a set of possible adaptations that can be made to a site's original menu. They choose which adaptations are performed on the basis of an evaluation metric that expresses the quality of the adapted structures.

The evaluation of adapted menu structures is always done offline as online evaluation of all possible adapted menus is not feasible. In an online evaluation all menus need to be placed on the site for some time until a sufficiently large number of users have used the menu. This would not only take an unacceptable amount of time, but also would mean that the users face a continually changing menu that is often even worse than the initial menu. In an offline evaluation the efficiency of the adapted structures is not measured directly, but predicted on the basis of a model of the user population.

If we compare the models of existing menu optimization methods, we find large differences in the underlying assumptions about the users' targets and navigation strategies. For example, some models assume that users read all available menu items before making a choice, while others assume that users stop reading as soon as an acceptable item is encountered. The assumptions behind the models are seldom mentioned explicitly and even more seldom validated. We feel this is a great deficiency as the used model specifies the direction of the optimization and thus determines for a large part the success of the optimization.

In this work we review the assumptions behind models that predict average navigation time. The various models are validated offline on real log data of four web sites with hierarchical menus. The contributions of this paper are threefold. 1) We make the assumptions behind navigation time models explicit, so that for a particular application one can select a model whose assumptions hold. 2) If in the experiments certain assumptions appear to be inherently better than others, this reduces the scope of the models that need to be considered when optimizing menus of new sites. 3) We provide a method to find for a new site the best fitting model among the potentially optimal models.

Section 2 discusses the models underlying various optimization methods. In section 3 we present a framework to compare the available models. Section 4 explains the procedure that we use to validate the models and in section 5 we apply the models to log data of four web sites. The last section contains conclusions and discusses the results.

2 Twelve navigation time prediction models

We examined the models for predicting expected navigation time of twelve menu optimization methods. Below, we briefly describe the context of the methods and their main properties.

One of the first menu optimization methods was developed by Witten and Cleary [1984]. They optimized the hierarchical index of a digital phonebook using the access frequencies of the phonenumbers. A limited time prediction model was used that assumes that the choice lists (the lists of categories located under the same items in the hierarchy) have equal and non-adaptable numbers of items.

Lee and McGregor [1985] explicitly sought to quantify the relation between menu structure and navigation time. They assumed users always searched for only one page and all pages had equal probability of being sought. Later Landauer and Nachbar [1985] extended their model to menus where the choice lists were ordered alphabetically. Paap [1986] added the possibility that the choice lists themselves were categorized. Fisher *et al.* [1990] improved the Lee and McGregor model by adding frequency based page probabilities. Moreover, they invented an algorithm to optimize menus on the basis of the improved model. A limitation of this algorithm is that it can only find structures that can be formed by removing intermediate nodes from the original hierarchy.

Bernard [2002] presented another model for predicting navigation time: the Hypertext Accessibility Measure (H_{HAI}). Like the Lee and McGregor model, the H_{HAI} measure predicts the expected navigation time solely on the basis of the menu structure.

The ClickSmart system [Smyth and Cotter, 2003] adapts WAP menus to the behavior of individual users. The time prediction model that is used is called the click-distance. This model is in fact an instantiation of the model introduced by Fisher *et al.*. To circumvent the problem of creating labels for new menu items, the optimization algorithm can only make hierarchies flatter and not deeper.

In [Hollink *et al.*, 2005] we presented a system that adapts web menus to individual users. We used a model that was similar to Fisher's model but, unlike Fisher's model, our model assumes that users sometimes make navigation mistakes. The applicability of the algorithm is restricted to situations in which the pages are labeled with keywords that can function as labels for the menu items.

The MESA model [Miller and Remington, 2004] is to our knowledge the only quantitative model that links the probability of making navigation mistakes to the quality of the items' labels. The connection between label quality and mistake probability seems natural, but the practical applicability of this model is limited as for all labels quality assessments need to be provided by experts.

Allan and Bolivar [2003] provide three models to assess the quality of a document hierarchy created through hierarchical clustering: the minimal travel cost, the expected travel cost and the expected accumulated travel cost. The models are not designed for predicting navigation time in web menus, but as they predict the amount of time users need to locate documents in a hierarchy, they can be used for this purpose without modification.

3 Time prediction framework

In this section we provide a framework that allows us to systematically compare the available navigation time prediction models. The core of the framework is formed by the dependencies shown in Figure 1. The time users need to navigate through a menu depends on the paths that they follow through the menu and the strategy they use to follow these paths. The followed paths in turn are a consequence of the users' targets, the nature of the menu and the strategy the users use to search the menu for their targets.

All time prediction models that we encountered follow this general schema. The differences between the models lie in their assumptions about the factors that determine navigation time. Below we review the target set features and the strategy features that are used in the optimization methods introduced in the previous section. In addition, we discuss the assumptions that the features represent and the circumstances under which these assumptions are justified. Table 1 lists the features, and positions the twelve navigation time prediction models in the framework.

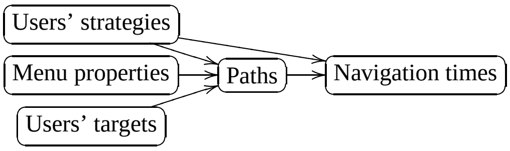

Figure 1: Causal dependencies between a menu structure, the users' targets, the users' strategies, the paths they follow through the menu and the time they spend navigating to their target pages.

Due to space limitations, the features that concern the characteristics of the menus are left out of the discussion. Some models only apply to menus with certain characteristics, for instance menus with equal numbers of items in all choice lists. However, these features can be observed directly from the menus, so that validating whether these features apply to the situation is trivial.

Most models in Table 1 actually represent *classes of* models rather than individual models. The models in these classes share the same features, but some of the features have parameters that need to be determined anew for each site. For example, the fact that a model uses page probabilities is a feature and the parameters of this feature are the probabilities of the pages of a particular site. In this work we evaluate model classes and not individual models. The word 'model' will be used to refer to both model classes and models.

3.1 Users' targets

A user comes to a site to fulfill certain information needs. The pages that together fulfill these needs we call the user's *target pages* or his *target set*. We distinguish three features of the users' target sets (see Table 1). First, most models assume that any set of pages can be a user's target set. Only the travel cost models [Allan and Bolivar, 2003] make use of predefined topics that form the possible target sets. According to these models a user is interested in exactly one topic and searches for all pages on this topic. The travel cost models are developed for assessing document hierarchies. In this setting the topics form the gold standard for the clusters at the lowest level of the hierarchy. The second feature is the size of the target sets. Most models assume each user searches for exactly one target. The travel cost models allow for the possibility that a user has multiple targets, namely all pages belonging to one topic.

The third feature is the probability distribution over the target sets. The models that are explicitly developed to predict navigation time all assume that the target sets have equal probability of being sought (uniform). They compute expected navigation time as the unweighted average of the times needed to reach each of the targets. All models used in optimization algorithms assume that the probabilities are proportional to the frequency of the sets in the log files. This extension has a clear value for menu optimization, as it causes algorithms to place more frequently accessed pages at a more prominent position in the hierarchy.

3.2 Navigation strategies

Table 1 contains seven features that concern the users' navigation strategies, five of which influence the prediction of the users' paths. The first feature, the users' search strategy, involves the order in which users open hierarchy nodes. Most models assume that users use a greedy depth

Table 1: Properties of navigation time prediction models

Model	Features of targets			Features of users' strategies						
	Target set	Target set size	Target set probabilities	Users' search strategy	Multiple target search	Mistake probability	Users' stop condition	Users' choice strategy	Node choice function	Node opening function
Witten and Cleary [1984]	all	one	frequency	greedy	-	0	all targets	-	0	linear
Lee and McGregor [1985]	all	one	uniform	greedy	-	0	all targets	read all/ until found	linear	linear
Landauer and Nachbar [1985]	all	one	uniform	greedy	-	0	all targets	read all	logarithmic	linear
Paap and Roske-Hofstrand [1986]	all	one	uniform	greedy	-	0	all targets	read until found	logarithmic	linear
Fisher *et al.* [1990]	all	one	frequency	greedy	-	0	all targets	read until found	linear	linear
Hypertext accessibility measure [Bernard, 2002]	all	one	uniform	greedy	-	0	all targets	read all	logarithmic	logarithmic
Click-distance [Smyth and Cotter, 2003]	all	one	frequency	greedy	-	0	all targets	read until found	linear	linear
Hollink *et al.* [2005]	all	multiple	frequency	greedy	separate	fixed	all targets	-	0	linear
MESA model [Miller and Remington, 2004]	all	one	uniform	greedy	-	label quality	all targets	read until found	linear	linear
Minimal travel cost [Allan and Bolivar, 2003]	predefined	multiple	uniform	greedy	continual	0	best category	read all	linear	linear
Expected travel cost [Allan and Bolivar, 2003]	predefined	multiple	uniform	exhaustive	continual	0	all targets	-	0	linear
Expected accumulated travel cost [Allan and Bolivar, 2003]	predefined	multiple	uniform	greedy	continual	0	all targets	-	0	linear

first strategy. According to these models, users perform a depth first search to their target pages. The users base their choices on the items' labels and only open items that lead to targets. For users with a single target page this means that they take the shortest path. The expected travel cost model assumes a different strategy. According to this model users perform an exhaustive depth-first search visiting all nodes until they happen to hit their targets. This means that in the worst case a user traverses the whole tree.

Users following the greedy strategy only open items that lead to targets. The second strategy feature, the users' choice strategy, concerns the way the users select these items from the choice lists. A user can read all labels and then select the best node or start reading at the top of the list and open an item as soon as an acceptable item is read.

The third feature concerns the behavior of users with more than one target. The simpler models assume that these users search for each target separately, in other words, that they go back to the starting point after a target is found. More complex models include a continual search pattern which means that users surf from the starting point to the first target and from this target to the second target, etc.

The fourth navigation strategy feature is the probability that users using a greedy strategy make navigation mistakes. Here making a mistake means selecting an item that does not lead to a target page. Most models simply assume users never make mistakes or make random selections with a small but fixed probability. The MESA model uses the quality of the items' labels to determine the probability of a user selecting an item erroneously.

The fifth element of the users' strategy is their stop condition. The minimal travel cost assumes that users stop navigating once they have reached the menu item under which most target pages are located. All other models assume users keep searching until all target pages are found.

The final two strategy features are the node opening function and the node choice function. They specify the relation between the path followed through the site and the navigation time. Navigation time is determined by two properties of the path: the number of menu items a user has opened ($|Path|$) and for each navigation step the number of items in the choice list that the user has read ($\#choices$):

$$Time = \beta.f(|Path|) + \Sigma_{\{n \in Path\}} \alpha.g(\#choices(n))$$

Here f is the node openings function and g is the node choice function. α and β are parameters that respresent respectively the time users need to read an item from the choice list and the time users need to open a menu item. The value of $\#choices(n)$ depends on the choice strategy of the users. As mentioned before, one can assume that users read all items of a choice list or that they stop reading when an acceptable item is found. For both functions f and g three variants appear in literature: a linear function, a logarithmic function and a null function, meaning that the factor has no influence. A linear relation between navigation time and the number of item openings means that opening an item takes equal time at each level of the hierarchy. A linear choice function implies that users go top-down through the choice lists and need equal time to read each item. A logarithmic choice function is justified when the list of items is ordered and people do not need to read every item to find the one they need. Finding a known item in an alphabetic list of n items can be done by making a series of binary splits, which results in reading only $log_2(n)$ items. A logarithmic opening function, which is used in the H_{HAI} model, can not be justified in this way, as one always has to open all items on the path.

4 Validating time models

The many differences between the twelve models make clear that choosing a navigation time prediction model for a menu optimization task is a non-trivial problem. For the menu features one can simply check whether they apply to the menu at hand, but the users' strategy and targets are not so easily observable. Some of the feature values are equivalent variants such as logarithmic and linear choice functions. Others are merely extensions of each other. For instance, a model with uniform target probabilities is in fact a simplified version of a model with frequency based probabilities. To find the best fitting model one needs to determine which of the variants model the situation best and whether the extensions lead to significant improvements.

Below we systematically test all valid combinations of features (including combinations that do not appear in the models in Table 1) to determine the relative importance of the various features. We create instantiations of the models for four web sites (i.e. we set the parameters of the model features). We apply the instantiated models to the sites' menus and log files and measure how well the models predict the users' paths and navigation times. These experiments lead to recommendations for using the more complex or the simpler features. For the features for which the optimal choices differ per site we provide a method to determine for a given site which choices are optimal.

The evaluation consists of three parts: first we validate the assumptions about how users with a given target set choose a path, then we validate the ones that determine navigation time given a path and finally we validate the assumptions about the users' targets. The following sections describe the procedures for validating the assumptions. In section 5 these procedures are applied to four web sites.

4.1 Data preprocessing

From the log files we restore the sessions of individual users. All requests coming from the same IP address and the same browser are contributed to one user. When a user is inactive for more than 30 minutes, a new session is started. The sessions include both target and non-target pages. We determine the most likely targets on the basis of the time the user spent on the pages. All pages with a reading time longer than or equal to the median reading time of the hierarchy's end pages are marked as target pages. The other pages form the paths to the targets. The rationale behind the use of the median reading time is that target pages are pages to which a user pays more than usual attention.

The median reading time is a crude criterion for selecting targets. However, in our experiments we found that choosing higher or lower time thresholds changed the absolute scores of the various models, but not their relative performance. Nevertheless, it is questionable whether reading time is at all a good criterion to select targets. It is plausible that on average users spent more time on target pages than non-target pages, but clearly this does not hold for every individual page view. Without prior knowledge reading time is the only source of information. However, on many sites characteristics of the pages can be used to make a more informed estimation of the users' targets, for example using the page characterizations used in the WUM method [Spiliopoulou and Pohle, 2001].

4.2 User strategies for predicting paths

Table 1 contains five features that influence the paths that users with a given target set follow through a menu. We determine the impact of the users' search strategy, the users' choice strategy, the search for multiple targets and the users' mistake probability. The stop condition is not used as we have no means to determine whether users would have liked to find more pages besides the ones they visited. We only test fixed mistake probabilities, because label quality assessments are generally not available.

Each combination of features is combined into a partial model that predicts paths. The partial models are evaluated by comparing the predicted paths to the paths that the users actually followed on the site. For each target set in the log files the models predict a path along all targets. In the end we count how many of the predicted page transitions actually occurred in the users' sessions. The models are compared on precision and recall. Here precision is the number of correctly predicted transitions divided by the total number of predicted transitions. Recall is the number of correctly predicted transitions divided by the number of transitions in the users' sessions. We focus on the page transitions rather than the visited pages themselves, because the transitions determine the navigation time, as we will see below.

4.3 User strategies for predicting times

We evaluate all features that determine the predicted navigation times: the users' choice strategy, the node opening function and the node choice function. Partial models that predict navigation times are formed for all combinations of features. For each path to a target page in the log files we compute the time it took the user to traverse the path. In addition, we count the number of menu items the user opened along the way and the number of choices he had in each step. To these data the time prediction models are fitted in such a way that the mean of squared errors is minimized. This results in optimal parameter settings for the models (values for α and β, see section 3.2).

A 5-fold cross-validation is used to evaluate how well the models predict navigation times of future users. The models are fitted to the training sets and evaluated on the test sets. As evaluation measure we use the R-square measure, which expresses the proportion of the variance in the users' navigation times that is explained by a model.

4.4 Predicting target sets

We validate models with various values for the target set size and the target set probabilities. All models assume that all target sets are possible. Models with predefined topics are not considered, as in general it is not possible to find a division in topics that applies to all visitors.

Again we split the log data in test and training sessions. The training data is used to compute the target set probabilities. During training each target set model produces a collection of target sets that simulates the targets of the actual users. The simplest model is the single uniform model. It assumes users search for single targets and all targets have equal probability. Its target set collection is a list of all pages of the site. The single frequency model also assumes users search for single targets, but now the target probabilities are based on the number of times each page occurs as a target in the training sessions. The multiple frequency model consists of target sets with more than one page. Its target set collection is a list of all target sets occurring in the

Table 2: Properties of the four sites that are used for evaluation.

Site	Log Period	Number of sessions	Number of menu items	Maximal menu depth
SG	9 months	51,567	92	3
RN	9 months	23,995	100	6
GH	1 month	22,788	59	6
HI	4 days	2,062	288	4

training set. The collection of the multiple uniform model would comprise all possible target sets (the power set of the site's pages), but the computation of this collection is not tractable for sites with more than a few pages.

The purpose of the test sets is to evaluate how well the target set collections of the three models reflect the targets of the actual users of the site. For each target set in each collection we estimate the time users need to locate the target pages using the path and time models that scored best in the previous evaluations. The expected navigation time of a collection is the weighted average time over all targets in the collection. The expected navigation times are compared to the average time that users from the test set really needed to locate a target. As evaluation measure we use the relative error: the difference between the expected navigation time and the real average navigation time as percentage of the real average navigation time.

5 Experiments

We applied the method described in the previous section to log data of four Dutch web sites. The sites are from different domains and their menus vary in size and structure. The SeniorGezond site (SG)[1] gives information about the prevention of falling accidents. It provides many different navigation means one of which is a hierarchical navigation menu. The Reumanet site (RN)[2] contains information about rheumatism. GHadvies (GH)[3] is a site about lay-off compensation. HoutInfo (HI)[4] contains pages about the properties and applications of various kinds of wood. Features of the sites' log files and menus are given in Table 2.

The partial models for path prediction were applied to the four sites. The results of the experiments are given in Table 3. There are only two models with exhaustive strategies, because with this strategy there is no difference between the two choice strategies. The exhaustive models predicted extremely long paths which resulted in moderate recall, but very low precision. The greedy models resemble the true strategy of the users much better: 40-55% of the predicted transitions were actually followed. No large differences were found between the two choice strategies. Possibly, this is because both strategies were used by large user groups. In all cases the continual target search models worked much better than the separate search models.

In a second set of experiments we added fixed mistake probabilities to the greedy models with multiple targets. Including navigation mistakes did not improve the models: both precision and recall decreased almost linearly with increasing mistake probability.

In conclusion, when optimizing a menu, the best choice is a greedy model without navigation mistakes. Either one of the choice strategies can be used. In addition, the model should take into account that users with multiple targets do not start over each time a target is found.

Table 3: Precision and recall of the path prediction models. E is exhaustive strategy, G is greedy strategy, C is continual target search, S is separate target search, A is read all choices, and U is read until good item found.

		Path model					
Data set		ES	EC	GSA	GCA	GSU	GCU
SG	precision	0.010	0.016	0.240	0.442	0.240	**0.443**
	recall	0.234	0.196	0.301	0.308	0.301	**0.309**
RN	precision	0.012	0.028	0.184	0.414	0.184	**0.417**
	recall	0.219	0.203	0.284	0.316	0.284	**0.318**
GH	precision	0.022	0.062	0.196	0.530	0.196	**0.535**
	recall	0.338	0.298	0.308	0.359	0.308	**0.363**
HI	precision	0.007	0.015	0.335	**0.500**	0.335	0.499
	recall	**0.517**	0.376	0.407	0.343	0.407	0.342

Table 4: Average R-square of the time prediction models. L is logarithmic, S is linear (straight), 0 is zero, and U is read until good item found. The first character is the node opening function and the second character the choice function.

	Time model							
Data set	00	S0	SSU	SLU	L0	LSU	LLU	H_{HAI}
SG	-0.01	0.88	**0.88**	0.88	0.73	0.84	0.85	0.74
RN	-0.00	0.67	0.68	0.68	0.69	0.73	0.72	**0.74**
GH	-0.00	0.78	0.79	**0.79**	0.64	0.75	0.74	0.72
HI	-0.00	0.84	0.86	**0.86**	0.62	0.75	0.76	0.80

The results of the experiments with time prediction models are given in Table 4. All figures are averages over the 5 folds. Due to space limitations the table only shows the models with *read until found* choice strategies. The results of the *read all* choice strategies are very similar as was the case in the path prediction experiments. The results of the H_{HAI} model [Bernard, 2002] are shown separately. This model is basically a double logarithmic (LLA) model, but with some small modifications.

The results of the time experiments are less clear than the results of the path experiments. Nevertheless, some observations can be made. Models that use both the number of node openings and the number of choices perform better than models that disregard the number of choices (00, L0 and S0) or the number of node openings (not shown). Apparently, both elements influence navigation time. As expected, on three of the four data sets linear node opening functions gave better results than logarithmic opening functions. Only on the Reumanet data set the logarithmic opening functions worked best, but on this data set all models performed low. Apparently, navigation times were more noisy on the Reumanet site. A possible explanation is that the site is visited frequently by people with rheumatism for who clicking links is more difficult.

The difference in performance between models with logarithmic and linear choice functions is small. We expected to find a preference for linear choice functions, because the sites have unordered choice lists. Apparently, visitors manage to select items without reading all preceding items. This can be a learning effect: when a user has opened an item before, he remembers where the item is located.

The values of the parameters α and β depend on the complexity of the labels and the experience of the users and differ per site. For the LLU model we found that β should be between 2 and 5 times as large as α. This coincides with the values used in the MESA model [Miller and Remington, 2004] $\alpha = 0.25$ and $\beta = 0.5$. In the click-distance model [Smyth and Cotter, 2003] selecting and clicking links takes equal time, but these values are meant for WAP users who navigate via mobile phones.

For a new menu optimization task, we recommend to use a linear node opening function, because this function tends

[1] http://www.seniorgezond.nl/

[2] http://www.reumanet.nl/

[3] http://www.goudenhanddrukspecialist.nl/

[4] http://www.houtinfo.nl/

Table 5: Relative error of the target set prediction models in combination with the GCU path model and the SLU time model.

	Target set model		
	Single	Single	Multiple
Data set	Uniform	Frequency	Frequency
SG	2.16	1.14	**0.15**
RN	2.46	1.11	**0.29**
GH	3.70	2.13	**0.04**
HI	3.10	1.69	**0.10**

to outperform other models and has better theoretical foundations. The best node choice function is strongly site dependent and should be determined anew for each site. This can be done offline in the same way we performed the time model experiments. At the same time these experiments will yield the optimal parameter settings.

In the target set evaluations we used the GCU path model and the SLU time model. Table 5 shows the error of the prediction of the expected navigation time when various target set models are used. The use of target set frequencies considerably improved the prediction. Furthermore, for all sites the model using target sets with multiple targets outperformed the models with singleton target sets. This confirms our earlier conclusion that it is important to model the behavior of users with more than one target.

In summary, in our experiments we found clear evidence that greedy continual search path models and multiple target frequency target set models are the best choices. If we compare these to the models in Table 1 we see that none of the optimization methods uses the optimal model class. This suggests that menu optimization can be improved by using the optimal average navigation time model.

6 Conclusion and discussion

In this work we gave an overview of the assumptions that are explicitly or implicitly used in navigation time prediction models. We presented a method to validate the assumptions offline using a site's log files. The method was applied to the menus of four web sites with hierarchical menus. In our experiments several model features appeared to be inherently better than others. These findings limit the set of models that need to be considered when the optimal model is sought for a new menu optimization task.

For the optimization of a menu in a new domain the path and target set models that performed best in our experiments can be used directly. We found that the optimal features of the path and target set models are the same for all sites. The best choice for the time prediction features is site dependent. Therefore, for a new domain the best time features needs to be determined from the log data. This can be accomplished with the method described in this work.

With the presented methods we can fit a limited set of models to a site's log file and make a well-funded choice for a navigation time prediction model. Using the right model is essential for menu optimization, because an accurate model of the users' behavior makes sure that one optimizes towards the menu with the shortest average navigation time. Comparison of our findings with the models used in menu optimization methods shows that all methods use suboptimal models. Thus, selecting the right models with the presented procedures can make menu optimization much more effective.

To obtain generally valid result we used web sites from different domains and with different characteristics. Nevertheless, it is possible that in other domains with yet other characteristics other models become optimal. Moreover, more relevant features may exists that are not present in any of the examined optimization methods.

Another limitation of the procedures described in this work is that they evaluate the models only on log data produced by users who used the sites' original menus, while the purpose of the models is to predict the average navigation times of menu structures after they have been adapted. To see how well the models generalize to new structures one needs log data created with different menus for the same site. Therefore, the next step of our research will be to incorporate the best performing model in an optimization tool. The tool will be applied to menus of real web sites and the resulting menus will be placed online. Comparison of the users' navigation times before and after the optimization allows us to evaluate the accuracy of the time predictions.

References

[Allan and Bolivar, 2003] J. Allan, A. Feng, and A. Bolivar. Flexible intrinsic evaluation of hierarchical clustering for TDT. *Proc. of the CIKM 2003*, pp. 263–270, New Orleans, USA, 2003.

[Bernard, 2002] M. L. Bernard. Examining a metric for predicting the accessibility of information within hypertext structures. *PhD. Thesis Wichita State University*, Wichita, USA, 2002.

[Fisher *et al.*, 1990] D. L. Fisher, E. J. Yungkurth, and S. M. Moss. Optimal menu hierarchy design: syntax and semantics. *Human Factors*, 32(6):665–683, 1990.

[Hollink *et al.*, 2005] V. Hollink, M. van Someren, S. ten Hagen, and B. Wielinga. Recommending informative links. *Proc. of the IJCAI-05 Workshop on Intelligent Techniques for Web Personalization*, pp. 65–72, Edinburgh, UK, 2005.

[Landauer and Nachbar, 1985] T. K. Landauer and D. W. Nachbar. Selection from alphabetic and numeric menu trees using a touch screen: depth, breadth and width. *Proc. of the SIGCHI conf. on Human Factors in Computing Systems*, pp. 73–78, San Francisco, USA, 1985.

[Lee and MacGregor, 1985] E. Lee and J. MacGregor. Minimizing user search time in menu retrieval systems. *Human Factors*, 27(2):157–162, 1985.

[Miller and Remington, 2004] G. S. Miller and R. W. Remington. Modeling information navigation: implications for information architecture. *Human-Computer Interaction*, 19:225–271, 2004.

[Paap and Roske-Hofstrand, 1986] K. R. Paap and R. J. Roske-Hofstrand. The optimal number of menu options per panel. *Human Factors*, 28(4):377–385, 1986.

[Smyth and Cotter, 2003] B. Smyth and P. Cotter. Intelligent navigation for mobile internet portals. *Proc. of the IJCAI'03 Workshop on AI Moves to IA: Workshop on Artificial Intelligence, Information Access, and Mobile Computing*, Acapulco, Mexico, 2003.

[Spiliopoulou and Pohle, 2001] M. Spiliopoulou and C. Pohle. Data mining for measuring and improving the success of web sites. *Special issue on applications of data mining to electronic commerce, Journal of Data Mining and Knowledge Discovery, 5(1-2):85–114, 2001.*

[Witten and Cleary, 1984] I. H. Witten and J. G. Cleary. On frequency-based menu-splitting algorithms. *Int. Journal of Man-Machine Studies*, 21:135–148, 1984.