Computational Linguistics Applications

Proceedings of the Computational Linguistics-Applications Conference pp. 85–91 ISBN 978-83-60810-47-7

Towards An Enhanced Semantic Approach For Automatic Usability Evaluation

Peter Steinnökel, Christian Scheel, Michael Quade, Sahin Albayrak
DAI-Labor, Technische Universität Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
{peter.steinnoekel, christian.scheel, michael.quade, sahin.albayrak}@dai-labor.de

***Abstract*—Today it is possible to judge the quality of user interfaces by using automatic testing agents, simulating user behavior for given tasks. Furthermore, such evaluations can be improved by giving these agents required knowledge to fulfill these tasks. Currently, such user task knowledge is more or less a string comparison between given knowledge that is represented as strings and the readable labels of the graphical user interface. Unfortunately, this approach leads to ambiguity problems, where real users cannot intuitively fulfill a task, even if the automatic testing found no obstacles. We propose to represent knowledge not in such a simple way, but using language resources and ontologies to represent user knowledge. We show that such representation helps detect and avoid ambiguous labels in graphical user interfaces.**

I. Introduction

IN OUR daily life, we face more and more interaction with technical equipment. Whether at home or at work, we have to operate computers, smartphones, tablets and televisions, to name the most common devices. Therefore not only do specifications, functionality and stability matter, but easy and efficient usability is a crucial factor. With more and more devices supporting our daily life, users often are not motivated to learn each device's operation from manuals [1]. Furthermore, the need for intuitively operable devices becomes clear when talking about devices for children and elderly people, as these user groups have special demands. It is the main challenge of user interface designers to satisfy these needs. Therefore, usability evaluations need to be applied in an early stage of development and the results have to be reintegrated into the design process.

In order to manually evaluate the usability of an application, two main techniques are applied: expert based evaluations like cognitive walkthrough [2] or heuristic evaluation [3] and user tests [4]. However, even if these expert based procedures are cheaper and easier to perform compared to user tests, it would be better if the effort and costs, required for usability testing could be decreased even more. Additionally, interface designers were given the possibility to use tools and automated testing routines in order to automatically judge, if an interface matches given requirements. These *automatic usability evaluation* (AUE) have the advantage that the interface designer can compare different user interfaces (UIs) at different steps of implementation, because these AUE works deterministically. When conducting user tests, it is very likely that one gets varying results with diverse subjects, and different usability experts might come to different conclusions when performing a cognitive walkthrough.

We present an approach to solve the following problem: *How can systems automatically evaluate if a given user interface has misleading semantic captions?*

Our approach enables MeMo [5]–[7], a tool for AUE to relate a simulated agent's user task knowledge (UTK) to the UI's labeling, and to semantically evaluate the labels instead of performing a simple string comparison. This enables UI designers to test interfaces of their application for a given task with different UTKs. The proposed method is able to reveal problems within the menu structure of a given UI, as well as discover ambiguity problems. An important factor for the evaluation are coherent results that do not output vector representations of the solution (which are difficult to interpret), but instead give easily comprehensible suggestions about potential usability problems and how to resolve them. Hence, the presented approach can be used to explain difficulties in a format that might be easier to understand for humans, so that user interface designers can avoid misleading labels in the next interface candidate.

Furthermore, to the knowledge of the authors, this is the first approach to evaluate UIs within the German language, based on German language resources. However, the proposed approach can easily be transferred to other languages where similar resources are available.

II. Related Work

A. Related Work on Information Retrieval

In order to evaluate the semantic relatedness of two terms there are several approaches in the domain of Information Retrieval (IR). A similarity measure is needed to compare terms occurring on the User Interface and the user's knowledge. The most prominent ones which occur in the domain of AUE are latent semantic analysis (LSA) and pointwise mutual information (PMI).

1) Latent Semantic Analysis: LSA [8], [9] is based on the assumption that similar terms, which are close in meaning will occur together in a text. It computes a matrix, where the columns correspond to documents and the rows to terms. Entries in the matrix are given by the number of the term's occurrences in the document. A dimension reduction technique called singular value decomposition (SVD) is applied. This results in a lower dimensional matrix which preserves the

similarity structure among rows. This matrix can be used to compare two terms by taking the cosine for the corresponding rows.

2) Pointwise Mutual Information: PMI-IR [10] is not based on co-occurrences on a given text, instead it uses the results of a search engine. AltaVista is asked how many documents contain both terms and how many contain only one term. The ratio of these numbers is the PMI score. However, the calculation can be extended to other constraints such as, that the terms must occur close to each other or by excluding antonyms.

3) extracting DIStributionally related words using CO-occurrences (DISCO): DISCO [11], [12] uses a tokenized corpus and calculates co-occurrences within a context windows of size $\pm$ 3 words and also stores the relative position of the term. This ordered pair of words and windows position allows to find terms which occur in a similar context.

B. Related Work on Automatic Usability Evaluation

The application domain of the described IR techniques in this paper is AUE. There are different approaches which aid UI designers to automatically evaluate interfaces. The graphical design can be evaluated by algorithms that compute the attention focus and saliency map, e.g. Pirolli [13], Halverson [14], or commercial products, e.g. EyeQuant by WhiteMatter Labs. Objective qualities such as contrast, button size, font size etc. also can be evaluated by rules [7] similar to heuristic evaluations. CogTool [15] is a tool to obtain predictions of skilled performance time for a predefined task. It has been developed for user interface designers with no experience in cognitive modeling or programming [16]. The UI designer has to model an UI within CogTool and then demonstrate the task. CogTool automatically generates an ACT-R model [17] which will then predict the skilled interaction time. The theory is based on the Keystroke-Level Model [18] and combined with the simple ACT compiler [19]. Basically the ACT-R code is generated by rules like "insert a *think operator* before every *look at operation*", which results in a pause of 1.2 seconds before the next operation is performed. Within these rules the semantics of the labels do not play a role, but the length of the word does; i.e., the predicted time for processing or typing a word or sentence is not based on the time needed to process it semantically, but instead on the number of syllables counted and an approximation of how long the processing will take.

1) CogTool Explorer: In order to include semantic processing, John et al. have included SNIF-ACT 2.0 [20] in a new version of CogTool called CogTool Explorer. It is based on the assumption that users follow text labels that are semantically similar to the task description based on the concept of information scent [21]. A visual search strategy guides the "eye", beginning at the upper-left corner and proceeding to look at the closest link to the model's current point of visual attention. The estimation of information scent has used latent semantic relatedness in order to calculate the relation between the search goal and the information on the display. The predicted action of the user is a tradeoff between choosing the best found result, continuing to look for better results or canceling the search [22].

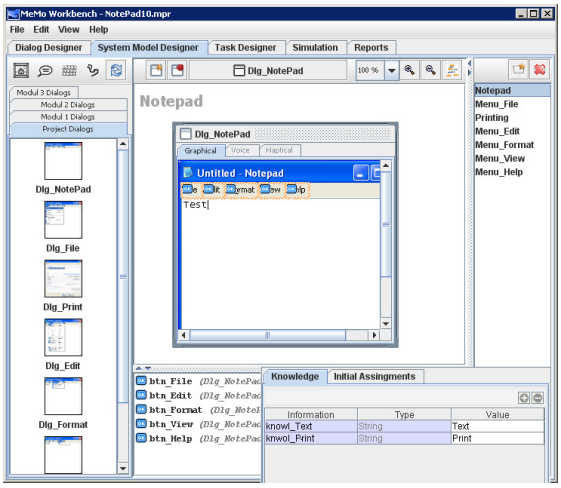

Fig. 1: System Model and UTK of the Notepad Interface

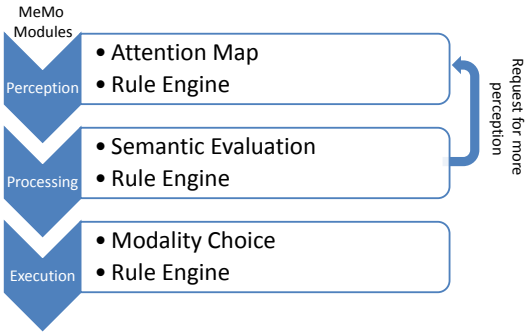

Fig. 2: The Three Modules of the MeMo Usermodel

2) Automatic Cognitive Walkthrough for the Web: The tool Automatic Cognitive Walkthrough for the Web (ACWW) [23], [24] focuses on identifying usability problems in web navigation. ACWW needs a specific search goal and analyzes the text of the links available on the web page. Using PMI-IR, the tool evaluates the familiarity of the texts compared to the search goal. The output is a score of how similar various links are to the search goal. This allows the program to tag those links which might distract the user from choosing the correct link. For the simulation, one can choose from different knowledge bases which reflect varying educational levels (ranging from 3rd grade to first year of college). However, ACWW does not predict an exploration path based on a visual search strategy as CogTool Explorer does [25].

III. MeMo Workbench

The Mental Modeling Workbench (MeMo) is a tool with the goal of automatically evaluating user interfaces. In contrast to CogTool, the objective of MeMo is not to evaluate experienced user interaction time for a given task, but rather estimating how good the task completion rate is for inexperienced users, and thereby suggest how to design more easily usable interfaces. Similarly to CogTool, the designer has to model the complete set of interfaces with all UI-objects (buttons, links, labels, check boxes, etc.) and transitions (to which UI frame an interaction leads). But instead of demonstrating the task to the system, an agent will be supplied with UTK in order to autonomously navigate through the UI. The UTK has to be specified by the designer and should contain all information needed to complete the task. An example is, if the UI designer wants the simulated user to print a given text in the Microsoft Notepad editor, he needs to provide the UTK "print" and "text" (see bottom right of Figure 1). During the simulation three different modules will be executed: perception, semantic processing and execution (see Figure 2). The perception module defines the order in which the UI-objects will enter the processing module. An algorithm builds on the findings of object-based visual attention [26], determines the order in which regions catch the user's attention, and defines probabilities for every UI-object getting forwarded to the processing module [27]. These probabilities are modified by rules which take the user's attributes (e.g., vision and age) into account as well as the properties of the UI-object (e.g., font size and button size).

The processing module is in charge of the linking of the UTK and the annotations on UI-objects. The output of the processing module is a probability distribution over the UI-objects, telling the system how likely it is that the user decides to interact with those UI-objects. Before an UI-object is chosen, the probabilities will again be modified with rules by means of cognitive attributes of the simulated user (e.g., time pressure or mental abilities). The processing module can decide either to choose an UI-object or to request more input from the perception module, in case it could not decide for one object yet (see Figure 2). When a specific UI-object is selected for interaction, the execution decides if the motor action will be successful, based on the coordination skills and size/position of the UI-object.

IV. Semantic Approach to User Interface Evaluation

As mentioned in the previous section, the semantic processing of the UI-labels is located in the processing module of the MeMo Workbench. In order to perform this semantic evaluation, the UI-designer has to specify an agent with the knowledge needed to solve the task. This UTK is comparable to the instructions a usability evaluator gives his subjects in a user test. Considering a simple task of printing an already written text in Notepad, one should provide the user with the knowledge: "print" and "text". The user hereby already knows what the task is and receives some hints how to solve it. However, in more complex cases, the task description might not be as precise and unambiguous as in the Notepad example. In these difficult tasks, it would be interesting to evaluate if the user is still able to complete the task without serious problems. Our approach for this evaluation aims at performing a human-like process and reports the results to the UI-designer.

In the first step of the processing module, all UI-labels and UTK items are converted to their lexical base form via OpenThesaurus stemming [28]. The different base forms of the UI-labels and the task knowledge are compared and if they return one or more direct matches, MeMo can continue to process these UI-objects. In this case there is no need for further semantic evaluation because the UI-labels are very precise given the UTK.

But in the general case, MeMo does not find a direct match in the spelling of the words and therefore we need further evaluation of the semantics. For that purpose we retrieve synonyms and similar words from established semantic resources available in the web. The lexical-semantic networks used are *OpenThesaurus*, *Wortschatz* and *GermaNet*.

1) OpenThesaurus: is a "database-driven website to collect synonyms of the German language." [28] The specialty is that OpenThesaurus is developed by linguists, but everybody is able to contribute to the database. [29] It has the largest corpus of synonyms of all lexical semantic resources for the German language [30] and is therefore very suitable for our approach.

2) Wortschatz: [31] is developed by the University of Leipzig and provides a web service that can be used to retrieve synonyms and "near"-synonyms from a corpus. It has a wider concept of synonyms and also contains looser associations than only true synonyms which is helpful for our approach, because it increases the amount of relations that can be found [32]. These two resources are available for research purposes but can only be accessed online.

3) GermaNet: is a lexical-semantic net and is comparable to WordNet [33] but working on the German language rather than English. GermaNet is the most developed resource used in our approach and it can be used offline to retrieve a very large set of similar terms, and is not restricted to synonyms.

This procedure for obtaining synonyms and similar words leads to a large extension of the UTK and of the information annotated on the UI-objects. This newly generated set of words can now again be compared, and matches indicate a linkage between the original information. However, it is possible that the algorithm finds several potential matches between UTK and UI-labels, either via a direct match or via the similar words. In order to compute the preferred connection we used DISCO [11] to rank these options. DISCO enables MeMo to calculate which word relation seems to be most likely in a more human way.

At this stage MeMo has collected the following information: all connected UI-labels and UTK items, a term that relates those two, and a similarity value retrieved from DISCO. Table I shows an excerpt of the retrieved similarity scores.

MeMo uses this information to determine a probability distribution over all UI-objects, how likely it is that one UI

TABLE I: Collected Information

#	UI-label	UTK	related term	similarity value
1	Assistance	Emergency	Help	0.03
2	Assistance	Medicine	Help	0.05
3	Health	Emergency	Hospital	0.08
4	Health	Medicine	Physician	0.19
5	Communication	Communication	*direct match*	1

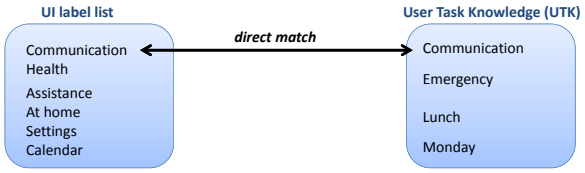

Fig. 3: String Comparison Evaluation

Object is chosen. Based on this probability distribution, MeMo decides whether to interact with one of these objects or to look further at unknown regions of the UI. The likelihood of the interaction with an UI-object *(p)* is calculated based on the similarity value *(s)*. All similarity values are divided by the sum over all values in order to normalize them to 1:

$$p_j = \frac{s_j}{\sum_{i=0}^{I} s_i}$$

However, these probabilities are restricted by maximum values *(mv)*:

- 1st order $(o = 1)$: $mv = 1.0$:
  Direct match between UI-Label and UTK
- 2nd order $(o = 3)$: $mv = 0.9$:
  UI-label and UTK have at least one similar term
- 3rd order $(o = 3)$: $mv = 0.8$:
  UI-label has a similar term which is again similar to UTK

These maximum values are not fixed and can be edited by the designer using MeMo. In order to evaluate if more perception is needed to decide on a UI-object, the order *(o)* of the connection and the ratio *(r)* of seen/un-seen UI-objects is taken into account:

- $o = 1 \wedge r \geq 0.5 \rightarrow$ start execution module
- $o = 2 \wedge r \geq 0.65 \rightarrow$ start execution module
- $o = 3 \wedge r \geq 0.8 \rightarrow$ start execution module

The perception and processing stops when one of these conditions is fulfilled. Again, these values for the ratio are not fixed and can be edited by the designer using MeMo.

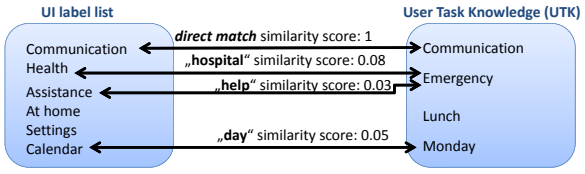

Fig. 4: Advanced Semantic Evaluation

Figure 3 and 4 demonstrates the enhanced possibilities of the proposed advanced semantic evaluation in contrast to a simple string comparison (see Figure 3). While the simple string comparison finds just one relation between the UI labels and the UTK, the advanced semantic evaluation is able to find more relations. The words in the quotation marks show the most similar word between each UI-label/UTK pair, and the similarity score is the value retrieved from DISCO, which indicates how strong the relation between the original words is.

When thinking about the aim of MeMo, one has to keep in mind that the UI-designer should receive as much information as possible in order to enhance the usability of his user interface. Therefore, all retrieved connections of terms are reported, as well as their similarity score (see Figure 4). This is the main reason MeMo detours to retrieve the similar words, instead of using (for example) LSA on all the possible connections. Additionally, this gives us the option to store the retrieved terms in data files and build personalized databases of general knowledge. This can then be edited manually, for example in the case where one wants to model explicitly that a specific user type is not aware of some connections between words. A typical example in which a designer could need this feature is in the case of elderly users or users who are not familiar with the Internet. They might not know that there is a relation between “YouTube” and “video”. This linkage can then be cut by hand.

V. Experimental Results

We evaluated the semantic processing approach with an application for support of the elderly (see Figure 5). It is an application developed to control technological equipment in a smart home and giving seniors the possibility to send an alarm call in case they do not feel well. The evaluated start frame contains several buttons and two of major interest: “Assistance” and “Health”. In the first version of the application, the function “send emergency call” was placed under the menu “Assistance”. But MeMo’s semantic processing relates the UTK “emergency” with higher probability to the “Health” menu. Figure 6 shows the reporting graph created by MeMo. This was confirmed by a usability expert who performed a cognitive walkthrough on the task as well as by ACWW. Table II shows the resulting similarity values from MeMo and ACWW. Even if they differ in the exact values, both approaches identified the same distracting UI-label (see Table II). However the choice of the goal-term in ACWW turns out to be more sensitive: If only “emergency” is chosen as goal, it does not identify “health” as competing term, while MeMo does. Table II shows the retrieved similarity values with the help of DISCO, while the significant ones, chosen by MeMo are printed in bold font.

In order to evaluate our approach against some existing data, we chose to test against data collected by Teo and John [34], who evaluated the effects of the position of a link on a web page. In the first setting of the experiment, the target link was placed in the middle on the right side of two columns

Fig. 5: Application for support of the elderly used for evaluation. For instance, this application was evaluated for the given task "send emergency call".

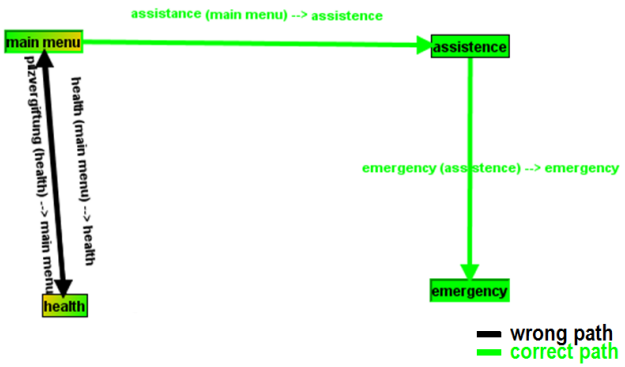

Fig. 6: Screen shot of MeMo's graphical reporting: Automatic evaluation results show that the simulated user interacts with "health" before MeMo chooses the right item to complete the task send "emergency call".

while in the second setting it was placed on the top left. The problem to evaluate was that in the first setting, other links which are highly related to the target link are placed in a position which is more likely to catch the subject's visual attention. Therefore in the experiment with the link on the right side of the web page, participants where distracted by semantically similar links.

Performing the same experiment with MeMo has revealed equal terms which distract the user from choosing the right link as ACWW does. The goal in the experiment is to click on the link which leads to "Canon Law". While "Theology" is the correct term, both MeMo and ACWW identify "Religion" and "Religious" as competing terms. Even if the values of the similarity differ within both approaches it results in the same error (see Table III). Note, that the results of MeMo are based on the German translation of the English terms, this might have led to slightly different values.

Figure 8 shows the data collected by Teo and John [34] (results of user tests and CogTool Explorer predictions) and the data collected by 200 MeMo simulations (with the parameters stated in section IV). MeMo achieved results that are closer to the observations with real participants than CogTool Explorer's

TABLE II: Evaluating the interface in 5 with the goal "Emergency"

UI-label	UTK	MeMo	ACWW
Assistance	Medical Emergency	**0.26**	**0.31**
Health	Medical Emergency	**0.1**	**0.32**
Assistance	Emergency	**0.03**	**0.44**
Health	Emergency	**0.08**	0.04

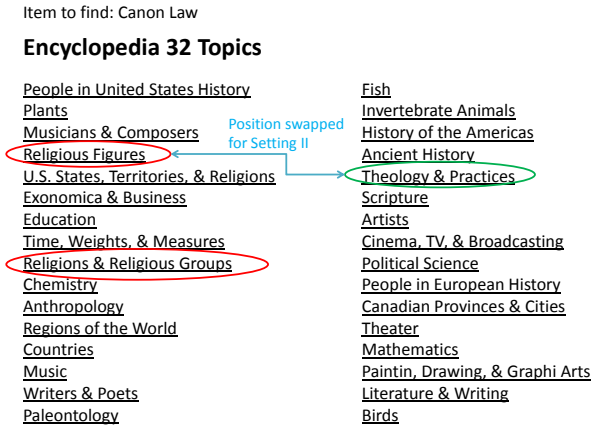

Fig. 7: Setting of the ACWW Experiment

results are. These results rely mainly one the adjustment of the parameters and on the performance of the perception module, which is independent of the semantic processing.

Figure 9 shows the average clicks needed to reach the goal of the task. The experiments with participants have shown that significantly more clicks are needed to fulfill the task. MeMo's results reflect this better than ACWW, which did not indicate a significant difference [34]. However, the number of clicks MeMo needed was significantly larger compared to the results from the user tests or from the CogTool Explorer tests. A first evaluation revealed that this is due to the fact that when the MeMo decides for an UI object based on perception-processing cycles, and the chosen UI object turns out to be

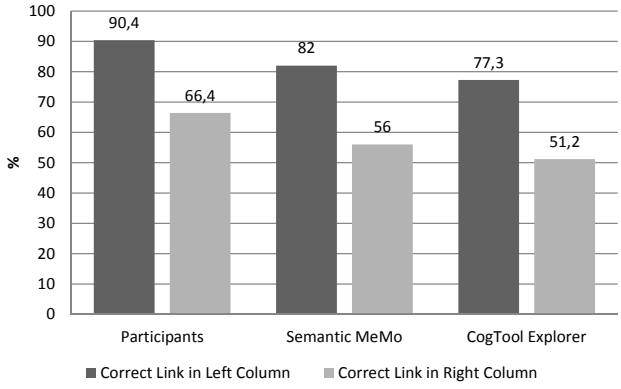

Fig. 8: Percentage of the First Click Success

TABLE III: Evaluating the Web page with the Goal "Canon Law"

UI-label	UTK	MeMo	ACWW
Theology	Canon Law	**0.62**	**0.03**
Religion	Canon Law	**0.25**	**0.03**
Religious	Canon Law	**0.02**	**0.02**

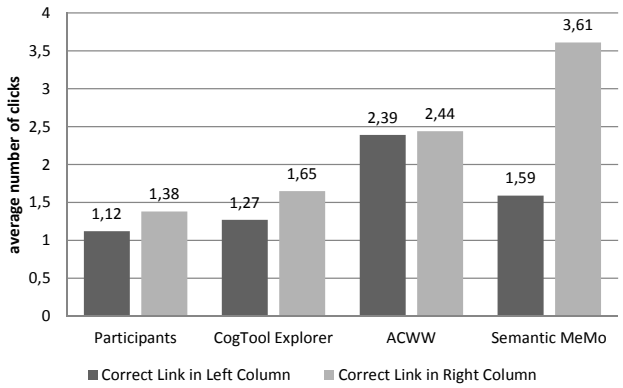

Fig. 9: Mean Clicks on Webpage Until Success

wrong, it does not start perception immediately again, to gather the rest of the UI. Instead the link with the second best match is chosen for interaction. We will update this behavior in the next iteration of our approach.

VI. Conclusion & Outlook

This paper has described an approach for evaluation of the semantic labeling of a user interface in order to find ambiguous captions or misleading ordering of the menus in the German language. With the aid of linguistic resources (GermaNet, Wortschatz, OpenThesaurus and DISCO), we are able to automatically perform this evaluation. We conducted several experiments and achieved results which correlate more with user data than current other approaches. However MeMo must be parameterized and other approches work parameter free.

A. Limitations

An important issue in enhancing semantic AUE is to describe everything semantically. When telling a test person to "make an emergency call", this task is mapped to an internal model in the person's mind. Until mapping from a given task to an ontology representation can be done automatically, the semantic task description should be manually described. The quality of the automatic testing agents may be improved, when not only the labels but also the task is non-ambiguous.

This processing of more complicated UTK, such as "print the last two pages of the document (but without the references)" is not possible at the moment, because this would require further grammatical processing which MeMo is not capable of. The same problem holds for more complex labels e.g. "show the weather report for the next three days" which can not be processed yet, because the used linguistic resources work only on single terms. Here the user of MeMo needs to make simplifications in order to analyze such UIs. However this mapping and syntactic processing should be further researched.

B. Further Work

There are several ideas for future work, which include dynamic evaluation of UIs at runtime [35] or using CogTool's mechanism for time prediction.

Since MeMo needs many parameters which have to be set by hand, we are planning reduce the number of parameters and find some standard values.

We are also planning to evaluate more UIs and conduct case studies with real participants to validate our results with our own data.

VII. Acknowledgment

Work presented in this article was funded within the SmartSenior project by the German Federal Ministry of Education and Research (BMBF, FKZ 16KT0902)

References

[1] W. Ijsselsteijn, H. Nap, Y. de Kort, and K. Poels, "Digital game design for elderly users," in *Proceedings of the 2007 conference on Future Play*. ACM, 2007, pp. 17–22.

[2] C. Wharton, J. Rieman, C. Lewis, and P. Polson, "The cognitive walkthrough method: A practitioner's guide," *Usability inspection methods*, pp. 105–140, 1994.

[3] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," in *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*. ACM, 1990, pp. 249–256.

[4] J. Nielsen, *Usability engineering*. Morgan Kaufmann, 1993.

[5] K. Engelbrecht, M. Quade, and S. Möller, "Analysis of a new simulation approach to dialog system evaluation," *Speech Communication*, vol. 51, no. 12, pp. 1234–1252, 2009.

[6] K. Engelbrecht, M. Kruppa, S. Möller, and M. Quade, "Memo workbench for semi-automated usability testing," in *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[7] S. Feuerstack, M. Blumendorf, M. Kern, M. Kruppa, M. Quade, M. Runge, and S. Albayrak, "Automated usability evaluation during model-based interactive system development," *Engineering Interactive Systems*, pp. 134–141, 2008.

[8] T. Landauer, *Handbook of latent semantic analysis*. Lawrence Erlbaum, 2007.

[9] T. Landauer, P. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, no. 2, pp. 259–284, 1998.

[10] P. Turney, "Mining the web for synonyms: Pmi-ir versus lsa on toefl," in *Proceedings of the twelfth european conference on machine learning (ecml-2001)*, 2001.

[11] P. Kolb, "Disco: A multilingual database of distributionally similar words," in *Tagungsband der 9. Konferenz zur Verarbeitung nat "urlicher Sprache–KONVENS 2008*, 2008.

[12] ——, "Experiments on the difference between semantic similarity and relatedness," in *Proceedings of the ordic Conference on Computational Linguistics (ODALIDA)*, 2009, pp. 81–88.

[13] P. Pirolli, S. Card, and M. Van Der Wege, "Visual information foraging in a focus+ context visualization," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2001, pp. 506–513.

[14] T. Halverson and A. Hornof, "A minimal model for predicting visual search in human-computer interaction," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2007, pp. 431–434.

[15] B. John, K. Prevas, D. Salvucci, and K. Koedinger, "Predictive human performance modeling made easy," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2004, pp. 455–462.

[16] F. Ritter, S. Haynes, M. Cohen, A. Howes, B. John, B. Best, C. Lebiere, R. Jones, J. Crossman, R. Lewis *et al.*, "High-level behavior representation languages revisited," in *Proceedings of ICCM-2006-Seventh International Conference on Cognitive Modeling*. Citeseer, 2006, pp. 404–407.

[17] J. Anderson, "Act: A simple theory of complex cognition." *American Psychologist*, vol. 51, no. 4, p. 355, 1996.

[18] S. Card, T. Moran, and A. Newell, "The keystroke-level model for user performance time with interactive systems," *Communications of the ACM*, vol. 23, no. 7, pp. 396–410, 1980.

[19] D. Salvucci and F. Lee, "Simple cognitive modeling in a complex cognitive architecture," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2003, pp. 265–272.

[20] W. Fu and P. Pirolli, "Snif-act: A cognitive model of user navigation on the world wide web," *Human-Computer Interaction*, vol. 22, no. 4, pp. 355–412, 2007.

[21] L. Teo, B. John, and P. Pirolli, "Towards a tool for predicting user exploration," in *CHI'07 extended abstracts on Human factors in computing systems*. ACM, 2007, pp. 2687–2692.

[22] L. Teo and B. John, "The evolution of a goal-directed exploration model: Effects of information scent and goback utility on successful exploration," *Topics in Cognitive Science*, vol. 3, no. 1, pp. 154–165, 2011.

[23] M. Kitajima, M. Blackmon, P. Polson, and C. Lewis, "Autocww: Automated cognitive walkthrough for the web," in *Human Interface Symposium*, 2002.

[24] M. Blackmon, M. Kitajima, and P. Polson, "Tool for accurately predicting website navigation problems, non-problems, problem severity, and effectiveness of repairs," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2005, pp. 31–40.

[25] L. Teo and B. E. John, "Cogtool-explorer: towards a tool for predicting user interaction," in *CHI '08 extended abstracts on Human factors in computing systems*, ser. CHI EA '08. New York, NY, USA: ACM, 2008, pp. 2793–2798. [Online]. Available: http://doi.acm.org/10.1145/1358628.1358763

[26] Y. Sun and R. Fisher, "Object-based visual attention for computer vision," *Artificial Intelligence*, vol. 146, no. 1, pp. 77–123, 2003.

[27] A. Ruß, "Modeling visual attention for rule-based usability simulations of elderly citizen," *Engineering Psychology and Cognitive Ergonomics*, pp. 72–81, 2011.

[28] D. Naber, "Openthesaurus: Building a thesaurus with a web community," *Retrieved January*, vol. 3, p. 2005, 2004.

[29] ——, "Openthesaurus: ein offenes deutsches wortnetz," *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen. Beitr "age zur GLDV-Tagung*, pp. 422–433, 2005.

[30] C. Meyer and I. Gurevych, "Worth its weight in gold or yet another resource a comparative study of wiktionary, openthesaurus and germanet," *Computational Linguistics and Intelligent Text Processing*, pp. 38–49, 2010.

[31] (2011, Jun.) Deutscher wortschatz. [Online]. Available: http://wortschatz.informatik.uni-leipzig.de/

[32] T. Wandmacher, E. Ovchinnikova, U. Krumnack, and H. Dittmann, "Extraction, evaluation and integration of lexical-semantic relations for the automated construction of a lexical ontology," in *Third Australasian Ontology Workshop (AOW)*, vol. 85, 2007, pp. 61–69.

[33] B. Hamp and H. Feldweg, "Germanet-a lexical-semantic net for german," in *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*. Citeseer, 1997, pp. 9–15.

[34] L. Teo and B. John, "Towards a tool for predicting goal-directed exploratory behavior," in *Human Factors and Ergonomics Society Annual Meeting Proceedings*, vol. 52, no. 13. Human Factors and Ergonomics Society, 2008, pp. 950–954.

[35] M. Quade, M. Blumendorf, G. Lehmann, D. Roscher, and S. Albayrak, "Evaluating user interface adaptions at runtime by simulating user interaction," in *HCI BCS*, 2011.