Guided Exploratory Search on the Mobile Web

Günter Neumann[1] and Sven Schmeier[2]
[1]*DFKI - German Research Center for Artificial Intelligence, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany*
[2]*DFKI - German Research Center for Artificial Intelligence, Alt–Moabit 91c, 10559 Berlin, Germany*
{guenter.neumann@dfki.de, sven.schmeier@dfki.de}

Keywords: exploratory search, unsupervised topic extraction, search query disambiguation

Abstract: We present a mobile touchable application for *guided* exploration of web content and online topic graph extraction that has been successfully implemented on a tablet, i.e. an Apple iPad, and on a mobile device/phone, i.e. Apple iPhone or iPod. Starting from a user's search query a set of web snippets is collected by a standard search engine in a first step. After that the snippets are collected into one document from which the topic graph is computed. This topic graph is presented to the user in different touchable and interactive graphical representations depending on the screensize of the mobile device. However due to possible semantic ambiguities in the search queries the snippets may cover different thematic areas and so the topic graph may contain associated topics for different semantic entities of the original query. This may lead the user to wrong directions while exploring the solution space. Hence we present our approach for an interactive disambiguation of the search query and so we provide assistance for the users towards a *guided* exploratory search.

1 INTRODUCTION

The World Wide Web is a huge set of hyperlinked and semantically correlated documents. When searching the Web using standard search engines users get presented just some nodes of this Web in form of ranked lists of (text snippets and pointers to) documents. The underlying link structure is more or less hidden from the users' perspective.

This kind of Web lookup search has been shown to be quite successful if the user is mainly interested in retrieving facts or answers for her query (Marchionini, 2006). Important reasons are:

- It is hard to find any alternative successful competing way of searching the Web for ordinary users. Hence people got used to that way of searching the Web.
- On ordinary computers human-computer interactions are mainly done by typing on the keyboard. It is not hard to reformulate search queries in case the desired results are not in the best n documents (Hearst, 2009) .

So it seems that the simplicity and easiness of the interactions with current search engines are strongly correlated with the still keyboard-dominated human-computer interfaces.

Nowadays, the mobile Web and mobile touchable devices, like smartphones and tablet computers, are getting more and more prominent and widespread. For such devices the most convenient way to interact with is by tapping on buttons, swiping the screen, squeezing it with two or more fingers etc. It is reasonable to assume that the current success and popularity of such mobile devices is also due to the fact that ordinary users have vastly accepted these kind of touchable interfaces as a very convenient way of interacting with them.

Furthermore, we are convinced that touchable devices and interfaces also support the development and breeding of alternative Web search strategies like *exploratory search*. In such a search activity the user only has a vague idea of the information in question and just wants to explore the information space in order to develop new knowledge about the topic in question which usually involves multiple iterations of search (Marchionini, 2006).

In (Neumann and Schmeier, 2011) and (Neumann and Schmeier, 2012) we have shown that mobile touchable devices can be a very convenient way for realizing simple and intuitive exploratory search strategies and to provide an usable mobile device searches to "find out about something". The core idea of the underlying search strategy is:

1. A user query is considered as a specification of a topic that the user wants to know and learn more about. Hence, the search result is basically

Figure 1: The associated topics refer to three different entities.

a graphical structure of that topic and associated topics that are found.

2. The user can interactively further explore this topic graph using a simple and intuitive touchable interface in order to either learn more about the content of a topic or to interactively expand a topic with newly computed related topics.

However the success of working with such a system heavily depends on the quality of the topics presented in the topic graph. One possible source of insufficient quality is the uncovered, implicit ambiguity of a search query (which usually the user is not aware of or at least not of all possible readings, e.g., natural entities). For example, if the user looks for information about the person *Jim Clark* she might only have in mind either the racing driver or the Netscape founder[1]. As the retrieved search results may contain information about both entities or all of them, the topic graph will show associated topics that might lead the user into wrong directions while further exploring the search space (Fig. 1).

Consequently, the search strategy should be able to detect and uncover this sort of ambiguity and should explicitly use it for guiding the user's further exploratory search into the direction of the selected preferred reading. Hence, the goal and major contribution of the work presented in this paper is twofold: 1) to extend the above mentioned search strategy to a *guided exploratory search* by proposing a method for interactive disambiguation, and 2) to propose an automatic method for its evaluation.

2 A Strategy for Guided Exploratory Search

To begin with we will use the topic "Jim Clark" as a running example to briefly describe our approach of guided exploratory search before we present and discuss its details in the next sections.

A user starts her exploratory search by entering a query q consisting of one or more keywords used to represent the topic in question (in our example, just the two words "Jim" and "Clark"). Instead of directly computing and presenting a topic graph for q (as done in the previous mentioned non–guided exploratory search approach), possible senses of q are identified and enumerated by referring to an external knowledge base, Wikipedia in our case. Beside the fact that Wikipedia is known to cover a huge number of possible senses for a very large number of topics, we also consider Wikipedia as a suitable means of a human–computer interface in the sense that both, a human and a computer, can directly communicate in natural language (NL). Continuing our running example, this means that the search strategy determines all possible senses (i.e., Wikipedia pages) that entail q as part of the Wikipedia title (i.e., the NL name of the concept described in the Wikipedia page). All found readings are then sorted and presented to the user and the user is asked to select her preferred one.

Assuming that the user selects the "British racing driver" sense, then the major content of the Wikipedia concept (basically the first sentence s of a Wikipedia page which usually defines the concept) is used to create a new expanded query q' from q and s. Now, using q' an initial topic graph is computed on the fly from a set of Web snippets that has been collected by a standard search engine (currently, we are using Bing[2]). Rather than considering each snippet in isolation, all snippets are collected into one document from which the topic graph is computed. We consider each topic as an entity, and the edges are considered as a kind of (hidden) relationship between the connected topics. The content of a topic are the set of snippets it has been extracted from, and the documents retrievable via the snippets' Web links.

The topic graph is then displayed on a tablet computer (in our case an iPad) as touch–sensitive graph. By just selecting a node the user can either inspect the content of a topic (i.e, the snippets or Web pages) or activate the expansion of the topic graph through

[1] ... or the baseball player, the football player, the bank robber, the film editor, the war hero,...

[2] http://www.bing.com/

an on the fly computation of new related topics for the selected node. The user can request information from new topics on basis of previously extracted information by selecting a node from the topic graph. Note that each new query sent to the search engine is created from the label of the selected node and the "sense"-information *s* created above from Wikipedia. Thus, each search triggered by a selected topic node is guided towards the user's preferred reading. This is why we call it *guided exploratory search*.

The rest of the paper is organized as follows. We first summarize the major steps of the computation of a topic graph in section 3. In section 4 we present the major steps of the guided exploratory search and present a fully automatic method for its evaluation. Details about the touchable user interface are then presented in section 5. Section 6 relates our approach to others, and finally, section 7 discusses open issues and future plans.

3 Unsupervised Topic Graph Construction

The representation of results in a topic graph provides alternative possibilities to perform search on a mobile device. The process is as follows:

- Show main topics that are generated from snippets retrieved by an ordinary search engine instead of documents in a first step.
- Present topics as interactive graphical structures.
- Let the user interact with the system by different interaction methods.
- Presenting a complete document is the last step in the search process.

We consider the extraction of topics as (1) a specific empirical collocation extraction task where collocations are extracted between chunks combined with (2) the cluster descriptions of an online clustering algorithm. (1) and (2) are computed in parallel for efficiency reasons.

The collocation extraction (step (1)) is done by using a special measure of point-wise mutual information (PMI c.f. (Turney, 2001)) that explicitly takes distance information into account. For this we first tag the snippets with Part–of–Speech (PoS) information using the SVMTagger (Gimenez and Marquez., 2004) and chunk the PoS-tagged text in the next step. The chunker recognizes two types of word chains. Each chain consists of longest matching sequences of words with the same PoS class, namely noun chains or verb chains, where an element of a noun chain belongs to one of the extended noun tags[3], and elements of a verb chain only contains verb tags. We finally apply a kind of "phrasal head test" on each identified chunk to guarantee that the right–most element only belongs to a proper noun or verb tag. For example, the chunk "a/DT british/NNP formula/NNP one/NN racing/VBG driver/NN from/IN scotland/NNP" would be accepted as proper NP chunk, where "compelling/VBG power/NN of/IN" is not.

We compute the chunk–pair–distance model CPD_M using the frequencies of each chunk, each chunk pair, and each chunk pair distance. CPD_M is used for constructing the topic graph in the final step. Formally, a topic graph $TG = (V,E,A)$ consists of a set V of nodes, a set E of edges, and a set A of node actions. Each node $v \in V$ represents a chunk and is labeled with the corresponding PoS–tagged word group. The nodes and edges are computed from the chunk–pair–distance elements. Since the number of these elements is quite large (up to several thousands), the elements are ranked according to a weighting scheme which takes into account the frequency information of the chunks and their collocations. More precisely, the weight of a chunk–pair–distance element $cpd = (c_i, c_j, D_{ij})$, with $D_{i,j} = \{(freq_1, dist_1), (freq_2, dist_2), ..., (freq_n, dist_n)\}$, is computed based on point–wise mutual information (PMI, cf. (Turney, 2001)) as follows:

$$PMI(cpd) = log_2((p(c_i,c_j)/(p(c_i) * p(c_j)))$$
$$= log_2(p(c_i,c_j)) - log_2(p(c_i) * p(c_j))$$

where relative frequency is used for approximating the probabilities $p(c_i)$ and $p(c_j)$. For $log_2(p(c_i,c_j))$ we take the (unsigned) polynomials of the corresponding Taylor series using $(freq_k, dist_k)$ in the k-th Taylor polynomial and adding them up:

$$PMI(cpd) = (\sum_{k=1}^{n} \frac{(x_k)^k}{k}) - log_2(p(c_i) * p(c_j))$$

$$, where\ x_k = \frac{freq_k}{\sum_{k=1}^{n} freq_k}$$

For step (2), we use the online clustering system *Carrot2* (Osinski and Weiss, 2008) to cluster the snippets and to generate sensible cluster descriptions. *Carrot2* is based on the Lingo (Osinski et al., 2004)

[3]Concerning the English PoS tags, "word/PoS" expressions that match the following regular expression are considered as extended noun tag: "/(N(N|P))|/VB(N|G)|/IN|/DT". The English Verbs are those whose PoS tag start with VB. We are using the tag sets from the Penn treebank (English) and the Negra treebank (German).

algorithm. It firstly extracts frequent terms from the input documents and produces a term–document matrix. Secondly, it performs a reduction of this matrix using Singular Value Decomposition (SVD) for the identification of latent structure in the search results.

Finally, we combine the results of both methods (1) and (2), such that the cluster labels are used to filter out the collocation results using simple fuzzy matching methods. The visualized part of the topic graph is then computed from a subset of the filtered CPD_M using the m highest ranked chunk–pair–distance elements for fixed c_i. In other words, we restrict the complexity of a topic graph by restricting the number of edges connected to a node.

4 Guiding Text Exploration by Enumerating Senses

We already mentioned in sec.1 that the topic extraction process may suffer from possible ambiguities of the search query. Suppose, for example, the search query has two prominent senses then the set of retrieved snippets will quite likely also cover two different thematic areas and so the set of extracted topics, too. If the user performs an investigative search (see section 6) she will then possibly end up with confusion more than solution. In (Sanderson, 2008) it is reported that between 7% and 23% of frequent queries in the logs of two search engines are ambiguous. This not only includes ambiguous queries (e.g., caused by homonym keywords like "bank" or "jaguar") but also queries that may lead to a different solution space of a search engine's document pool. Hence in this context the disambiguation task is strongly correlated to the automatic determination of the user's intension or goals. For todays search engines these tasks become very tricky to solve as often enough both problems are correlated and occur at the same time for users' search queries.

Regarding our solution for exploratory search on mobile devices the disambiguation part is less hard to solve. As our system supports the idea and paradigm of exploratory search we also let the user decide in which thematic area her exploration should go. Hence our solution divides the above mentioned problems, query disambiguation and determination of the user goal, in a natural way by presenting to the user the possible directions before actually presenting the topic graph. The difficult part is to filter out topics and to gather new topics in case too many nodes in the current topic graph do not fit the chosen context.

In order to detect possible ambiguities and to present them in an appropriate way we are focussing on a knowledge–based method by making use of Wikipedia (cf. sec. 2 for our motivation of using Wikipedia). The idea is to first match the user query with entries in Wikipedia. If we find more than one match we trigger the disambiguation process. As a starting point we indexed a snapshot of Wikipedia into a *structured* Lucene[4] index containing the title and the abstract of each article in separate fields. The index contains 2.999.597 articles with 4.320.497 different terms and has a size of 7.63 GB on a disc. Using this knowledge base, our query disambiguation algorithm works as follows:

```
10 let Q=user's query;
20 let TG=produce_TG(Q); // initial topic graph TG
30 let LI=Lucene Index;
40 let q[]=SA(tokenize(Q));
50 let query=(title:+q[1] ... +q[n]);
60 let results[]=search(LI ,query);
70 if (num(results[]) > 1) {
80   let ass[]=SA(associated_topics(TG));
90   let Qexp=(title:+q[1] ... +q[n]) AND
       (body:+ass[1] ... +ass[m]);
100  let docs[]=search(LI, Qexp);
110  if (user chooses docs[i]) {
120    let s=definition_sentences(docs[i]);
130    let TGnew=produce_TG(Q + s);
140    return TGnew;
140 }} else {
150    return TG;} // return initial TG
```

We start to compute an initial Topic Graph TG with the original user query (20) using the TG construction process described in section 3. The steps (30) to (60) then compute the degree of sense ambiguity using Wikipedia in the following way. Firstly (40), we tokenize the query and apply Lucene's SimpleAnalyzer SA which lowercases all words in the query and deletes numbers. In a next step Lucene retrieves all documents that entail all tokens of the query in the titles of the articles (50+60). In this way it is guaranteed that we find all instances for an entity. The title of an article uniquely identifies each instance because it typically describes the entity in the article and is further qualified by parenthetical expressions. For example, the query for "Jim Clark" also matches "James (Jim) Clark", "Jim Clark (sheriff)", "Jim Clark (film editor)", etc. If only a single title matches or if there is no match at all, we return the initial topic graph TG (150). Otherwise (70) we know that the query matches different Wikipedia articles, and hence, that the query is potentially ambiguous.

In principle, we could now present the different concepts to the user just in the order determined by Lucene. However, the problem is that this ordering

[4] http://lucene.apache.org/core/

actually ignores the information already expressed in the topic graph TG. It could happen that the higher ranked elements in the ranked list are unrelated with the information used by the search engine and covered in TG. On the other hand, TG already expresses some interesting latent semantic information computed via the use of PMI, e.g., expressing that neighboring nodes of a node n are semantically more related to n than nodes with larger distance. Thus in order to achieve a more user query and TG related ordering we perform the following steps (80) to (140). Firstly, we perform a query expansion by adding topics from TG that are determined by a $1NN$ strategy (80) to the original query, i.e. we use only the directly associated topics. In the next steps (90 ff) we again formulate a query against our Wikipedia index. This time we use the associated topics to also search in the articles' body. The result is an ordered list according to the main topics in TG where the most probable meaning is listed first. The abstracts of the articles are presented to the user to chose from. We extract the most important terms (using the function definition_sentences() defined more precisely in the next listing) from the chosen article (120) and produce the final TG using the combination of the terms and the original query (130).

```
10 let first=article.firstSentence
20 let first_pos=POS_Tagging(first)
30 let sep=first_pos.indexOf(((is|was)(a|the)));
40 let isa_part=substr(first_pos,sep);
50 return filter_pos("N",isa_part);
```

According to Wikipedia article guidelines[5] usually an article contains a definition in the first sentence (10). Therefore we first tag the sentence with *PoS* information (20). If we find the definition phrases "is a", "is the", "was a" or "was the" we choose its right adjacent substring (30+40). If the definition phrase cannot be found, we choose the whole sentence. We filter out all tokens that are not tagged as nouns and return the remaining list (50).

4.1 Experimental evaluation

In the experimental evaluation we present an automatic way of how to determine the accuracy of the knowledge–based disambiguation algorithm. In a first step we use the above mentioned algorithm. Please note we evaluate real ambiguous queries only. Then we alter the original algorithm in the following way:

```
110 let right=0; all=0;
120 foreach(doc in docs) {
130  let s=definition_sentences(doc);
140  let TGnew=produce_TG(Q + s);
150  let ass[]=SA(associated_topics(TGnew));
160  let Qexp=(title:+q[1] ... +q[n]) AND
           (body:+ass[1] ... ass[m]);
170  let articles[]=search(LI,Qexp);
180  if(doc==articles[0]) {
190    right++;
200  }
210  all++;
220 }
230 final_accuracy=right/all ;
```

The idea behind this automatic evaluation is as follows: the topic graph produced, starting from a disambiguated document, results in a new Topic Graph $TGnew$. A search against the Wikipedia index using the original query for the title–field and the $1NN$ associated topics from $TGnew$ should have the disambiguated document as its best result.

In our experiments we took the entries of 'List of celebrity guest stars on Sesame Street"[6] ($Set1$) and the "List of film and television directors"[7] ($Set2$). Furthermore we evaluated both kinds of the topic graph construction process described above in sec. 3: Topic retrieval based on collocations only ($TopCol$) and its combination with the cluster descriptions ($TopClus$). Table 1 shows the results on the two datasets and the two different TG construction approaches (The first column says: 1:Set1; 2:Set2; A:TopCol; B:TopClus).

Table 1: Accuracy of disambiguation.

Set	All	Ambig	Good	Bad	Acc
1+A	406	209	375	54	87.41%
1+B	406	209	378	51	88.11%
2+A	1028	229	472	28	94.4%
2+B	1028	229	481	19	96.2%

4.2 Manual evaluation

To doublecheck the results of the previous section we also did manual evaluations on datasets by randomly picking results from several test runs and let two independent human judges (not the authors) check the correctness and usefulness of the topics for the chosen senses. This kind of evaluation is often used to evaluate unsupervised methods, cf. (Fader et al., 2011). The general setup was to count the number of correct vs. incorrect topics for a given sense. We furthermore gave the judges the chance to intuitively decide whether they would have followed right paths while

[5] http://en.wikipedia.org/wiki/Wikipedia:Lead_section#Introductory_text

[6] http://en.wikipedia.org/wiki/List_of_celebrity_guest_stars_on_Sesame_Street

[7] http://en.wikipedia.org/wiki/List_of_film_and_television_directors

exploring the solution space. i.e. the task of guiding the exploratory search would have been successful. Table 2 shows the results: the first column denotes the kind of topic retrieval like in the automatic evaluation, A for *TopCol* and B for *TopClus*. The next column shows the number of examples or senses that have been checked[8]. Column 3 shows the total number of extracted topics. The combined retrieval delivers less topics but as you can see in column 4 the quality seems to be improved as the ratio between correct and incorrect topics decreases for both testers. The last column shows whether the guidance towards topics for the chosen sence has been successful, i.e. the percentage of followed paths that are appropriate for the given sense. Please note the values in the columns 3–5 are highly subjective. So for example, for the second judge lots of tokens do not make sense in her opinion but on the other hand she would not have followed them during exploration anyway. Hence although she generally judged more topics not to fit, she rated the algorithms original sense, i.e. guiding the search towards the right direction, as more successful than the first judge.

However we see that the manual evaluations seem to proove the results and the method of the automatic evaluation.

Table 2: Manaual evaluation.

Set	All	Topics	Good	Bad	Guidance
A	20	167	132	35	ca. 95%
B	20	145	129	16	ca. 95%
A	20	167	108	59	> 97%
B	20	145	105	40	> 97%

5 Visualisation on Mobile Devices

In this section we briefly introduce the *guiding* part as it is implemented on the mobile device. Whenever the system finds any possible ambiguities in the search query the user receives a list of cells containing short expressive context information for the search term. In our example (Fig. 2) the search query has been "Jim Clark" and the user gets presented all possible found meanings. After selecting one of the cells by simply tapping on it the list-view flips back and the related topic graph is shown. In our example Fig. 3 shows the associated topics for Jim Clark the racing driver, Fig. 4 shows the results for Jim Clark the Netscape founder. The user now may interact with the

[8]Each judge checked the same examples independently

Figure 2: The alternatives to choose from (part)

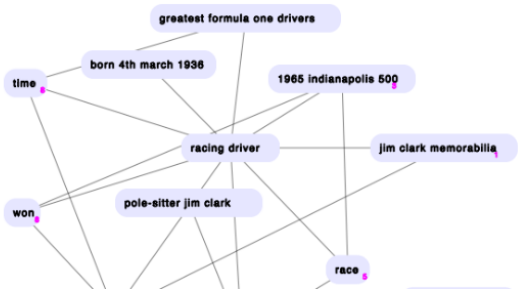

Figure 3: Excerpt of the topics for the British racing driver Jim Clark

graph by a single tap on a node - shows new associated topics for the node; squeezing with two fingers - zooms the view; sliding around - moves the topic graph; double tap - brings in a new view showing the snippets containing the topic of the node (Fig. 5). The cells are interactive and by tapping on a cell the corresponding Web page will be shown (Fig. 6).

In this way the user is able to explore the solution space by simple and well known interaction patterns.

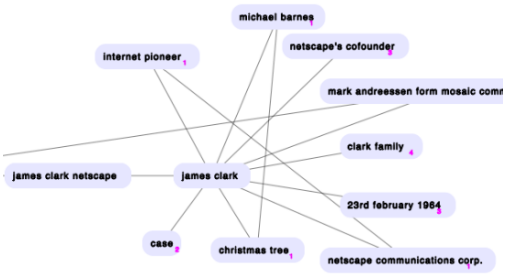

Figure 4: Excerpt of the topics for the Netscape founder James "Jim" Clark

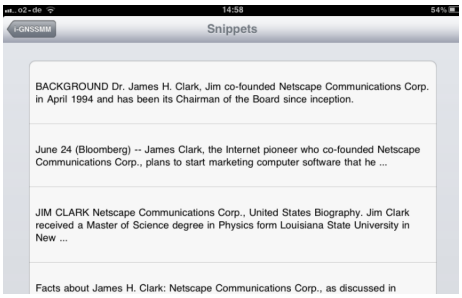

Figure 5: Excerpt of the snippets for the Netscape founder James "Jim" Clark after double tap on the node "netscape communications corp."

Figure 6: The website behind the second snippet of Fig. 5

6 Related Work

6.1 Exploratory Search

Nowadays information has become more and more ubiquitous and the demands of searchers on search engines have been growing, i.e. is a growing need for systems that support search behaviors beyond document oriented simple "one-shot" lookup. The research field *Exploratory Search* embedded in the field of Human Computer Interaction *HCI* explores the process of information seeking and tries to find solutions to support it. Exploratory search systems should for example discover new associations and kinds of knowledge, resolve complex information problems, or develop an understanding of terminology and information space structure. The general aim of this research is to come to a next generation of search interfaces to support users to find information even if the goal is vague, to learn from the information, and to investigate solutions for complex information problems. "Exploratory search can be used to describe an information-seeking problem context that is open-ended, persistent, and multi-faceted; and to describe information-seeking processes that are opportunistic, iterative, and multi-tactical" (White and Roth, 2009).

Exploratory searches are driven by curiosity or a desire to learn about or investigate something. According to Marchionini (Marchionini, 2006) a more detailed view on search is:

1. Lookup: Fact retrieval, Known-item search, Navigation, Transaction, Verification, and Question answering[9]
2. Learn: Knowledge acquisition, Interpretation, Comparison, Integration, and Socialize
3. Investigate: Accretion, Analysis, Exclusion, Synthesis, Evaluation, Discovery, Planning, and Transformation

The still dominating ranked list approach is well suited for lookup up search strategies, but probably less suited for a learn search strategy. For investigative search strategies it is too simple and does not support a discourse of questions and answers. Furthermore it is also known that information placed at the end of a ranked list will perhaps never be accessed (Sping et al., 2001).

The clustering interface *Grouper* (Zamir and Etzioni, 1999) has been originally implemented for the HuskySearch engine and it has been compared with the ranked list interface of the same. A clustering algorithm called SuffixÂt'Tree Clustering (STC) groups the search results into coherent groups. Through the analysis of behavior logs of the search engine with and without clustering it could be proven that finding specific documents that are ranked very high in the result set of the engine without clustering could be used more efficiently. After some time working with the system people enjoyed the clustering system more although not in all cases.

Findex (Käki, 2005) again used clustering to organize search results. An automatic computation of labelled categories/clusters based on the search results by Google is shown to the left side of the web interface. The clusters may be clicked to filter the overall search result set. The evaluation of the system has been based on an analysis of Web-logs and by a final questionnaires for the testers. The results pretty much confirmed the findings by Zamir and Etzioni: specific searches show less improvement than vague searchers concerning user's performance. Also users need to get used to the new kind of result presentation but they accept and even like it more after a very short time.

[9] Answering specific question like: when, who, where, how much - in contrast to: how, why, ...

WordBars (Hoeber and Yang, 2006) provides active user interaction during the search process in contrast to the previous systems. It visualizes an ordered list of terms that occur in the titles and snippets of the first 100 documents gathered by Google. The user has the possibility to add or remove terms from his query and thereby resorting the search results. In fact WordBars helps the user to refine her query and supports result exploration for specific and vague initial queries. They report that one fundamental design of their system is to create the right balance between computer automation and human control. Hence WordBars does not simply expand the original query but instead actually waits for user interaction before activating next steps. The crucial part is to present the possible choices as good as possible in order to create a real interactive Web information retrieval system. The authors show that for specific and for vague initial query their system is able to improve the overall result quality although there was no significant improvement concerning the user's performance.

WebSearchViz (Nguyen and Zhang, 2006) uses the analogy to the solar system for presenting the search results. The query represents the sun, the documents are the planets and location, speed, rotation, color, and distance of objects represent the ranking of the result documents. Relevant Web documents are determined by sending a user query to the Google search engine. Then, the user is asked to provide the subjects of interest and assigns weights for keywords that correspond to each subject. She can choose any subjects to be displayed in the visual space. The others will not be shown, still remain in the system unless the user explicitly deletes them. During the interaction with the visual space, the user is able to modify, add, delete, or redefine subjects at will. The visual space will be updated accordingly.

The *Lighthouse* system (Leuski and Allan, 2000) combines the well known ranked list representation and clustering visualization. The documents are represented as spheres floating in space and they are positioned depending on their mutual relatedness. The more related the closer are the spheres. Hence the result space shows clustered documents and documents that do not belong to any clusters. During evaluations for measuring users' acceptance the result showed positive results. Users showed to be more successful with the Lighthouse system than they are using ranked document lists.

(Akhavi et al., 2007) apply the results of a clustering algorithm on the representation like a fractal tree. It also supports zooming into the leaves of the tree and to find more and more details down to the document itself. The thickness of a branch represents the density, i.e. semantic closeness of the documents.

(Di Giacomo et al., 2007) organize search results of Web clustering engines. The *WhatsOnWeb*–system uses graphs instead of trees to present the clusters and sub–clusters of the result document set for a query. According to their evaluations the graph based interface showed more or less similar successful result identification by users compared to tree based systems. When it comes to find the correct single documents the graph based approach was more appropriate.

6.2 Web Query Disambiguation

There are several approaches for Web query disambiguation. The goal is not only to detect ambiguities in the words of the query but also to decide the right direction in the solution space and present it to the user. Some approaches like (Qiu and Cho, 2006) or (Chirita et al., 2005) try to automatically learn a user's interest based on the click history. In order to achieve this they provide a three step algorithm: 1) a model representing a user's interest based on the click history; 2) a process that estimates the user's hidden interest based on the click history; 3) a ranking mechanism that reranks the search engine result on the basis of 1) and 2). Other approaches like (Shen et al., 2005) or (Gauch et al., 2003) follow the same principle but with different learning and ranking algorithms.

Another approach is based on hyperlink structures of the Web and aims for a personal PageRank that modifies the search engines' PageRanks. Examples for this approach are (Haveliwala, 2002) and (Jeh and Widom, 2003).

A more generalizing approach consists of collaborative filtering methods. Here the search history of groups with similar interests are used to refine the search. This method has been used in (Sugiyama et al., 2004) where users' profiles are constructed using a collaborative filtering algorithm (Breese et al., 1998), or (Sun et al., 2005) where the correlation among users, queries, and clicked Web pages is analyzed. The advantage for the user is by the increased completeness of the search results because the knowledge base for the filtering process is already filled by other users - provided there are users with similar interests.

In contrast there has also been much research trying to post process the search results using clustering algorithms. (Liu and Lu, 2011) propose a very promising approach for disambiguation of person names. This approach does not require user models or a learning and personalization phase. The re-

sults from a search process are clustered by taking different document properties into account: Title, URL, metadata, snippet, context window (around the original query), context sentence, and bag of words of the whole document. The main property of this algorithm is the robustness and speed and hence the disambiguation performance. However it lacks - as reported in this paper - the labeling or definition of the clusters. So again the user has to check by reading at least some snippets inside a cluster (Cucerzan, 2007).

7 Conclusion and Outlook

We presented an approach of guided interactive topic graph extraction for exploration of web content. The initial information request is issued online by a user to the system in the form of a query topic description. Instead of directly computing and presenting a topic graph for the user query, possible senses of the query are identified and enumerated by referring to an external knowledge base, Wikipedia in our case. All found readings are then sorted and presented to the user and the user is asked to select her preferred one. The user–selected sense is then used for constructing an initial topic graph from a set of web snippets returned by a standard search engine. At this point, the topic graph already represents a graph of strongly correlated relevant entities and terms. The topic graph is then displayed on a tablet computer (in our case an iPad) as touch–sensitive graph. The user can then request further detailed information through multiple iterations.

Experimental results achieved by means of an automatic evaluation procedure demonstrate the benefit of the disambiguation method for exploratory search strategies. The automatic evaluation has been approved by another human evaluation. Currently, the main problem of our approach arises when an ambiguous query cannot be found in Wikipedia using our strategy. For example, the query "Famous Jim Clark" would not be found as we require that all words of the query occur in an Wikipedia article's title. Even if we could cope with this using a modified fuzzy search strategy we still would not find out ambiguities in queries that simply are not present in Wikipedia. However, in the running system we plan to give some feedback to the user by changing the color of the search entry. Then the user knows that there may be more then just one meaning for her query. Another open question is whether an improvement of our rather simple way of expanding the query using Wikipedia abstracts will lead to significant improvements of the disambiguation results. We are planning to do some research on this.

8 Acknowledgement

This research was conducted in the context of the project Deependance (funded by the German Federal Ministry of Education and Research, contract 01IW11003)

REFERENCES

Akhavi, M. S., Rahmati, M., and Amini, N. N. (2007). 3d visualization of hierarchical clustered web search results. In *Proceedings of the Computer Graphics, Imaging and Visualisation*, CGIV '07, pages 441–446, Washington, DC, USA. IEEE Computer Society.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann.

Chirita, P. A., Nejdl, W., Paiu, R., and Kohlschütter, C. (2005). Using odp metadata to personalize search. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '05, pages 178–185, New York, NY, USA. ACM.

Cucerzan, S. (2007). Large-scale named entity disambiguation based on wikipedia data. In *In Proc. 2007 Joint Conference on EMNLP and CNLL*, pages 708–716.

Di Giacomo, E., Didimo, W., Grilli, L., and Liotta, G. (2007). Graph visualization techniques for web clustering engines. *IEEE Transactions on Visualization and Computer Graphics*, 13:294–304.

Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011*, pages 1535–1545.

Gauch, S., Chaffee, J., and Pretschner, A. (2003). Ontology-based personalized search and browsing. *Web Intelli. and Agent Sys.*, 1:219–234.

Gimenez, J. and Marquez., L. (2004). Svmtool: A general pos tagger generator based on support vector machines. In *In Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04), vol. I, pages 43 - 46. Lisbon, Portugal, 2004. (ISBN 2-9517408-1-6)*.

Haveliwala, T. H. (2002). Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, WWW '02, pages 517–526, New York, NY, USA. ACM.

Hearst, M. A. (2009). *Search User Interfaces*. Cambridge University Press.

Hoeber, O. and Yang, X. D. (2006). Interactive web information retrieval using wordbars. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference*

on Web Intelligence, WI '06, pages 875–882, Washington, DC, USA. IEEE Computer Society.

Jeh, G. and Widom, J. (2003). Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 271–279, New York, NY, USA. ACM.

Käki, M. (2005). Findex: search result categories help users when document ranking fails. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '05, pages 131–140, New York, NY, USA. ACM.

Leuski, A. and Allan, J. (2000). Lighthouse: Showing the way to relevant information. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, INFOVIS '00, pages 125–, Washington, DC, USA. IEEE Computer Society.

Liu, Z. and Lu, Q. (2011). High performance clustering for web person name disambiguation using topic capturing. *Ratio*.

Marchionini, G. (2006). Exploratory search: from finding to understanding. *Commun. ACM*, 49:41–46.

Neumann, G. and Schmeier, S. (2011). A mobile touchable application for online topic graph extraction and exploration of web content. In *Proceedings of the ACL 2011 System Demonstrations*. ACL.

Neumann, G. and Schmeier, S. (2012). Exploratory search on the mobile web. In *4th International Conference on Agents and Artificial Intelligence (ICAART 2012)*, pages 110–119. SciTePress.

Nguyen, T. and Zhang, J. (2006). A novel visualization model for web search results. *IEEE Transactions on Visualization and Computer Graphics*, 12:981–988.

Osinski, S., J.Stefanowski, and WeissOsinski, D. (2004). Lingo: Search results clustering algorithm based on singular value decomposition. In *Proceedings of the International IIS: Intelligent Information Processing and Web Mining Conference. Advances in Soft Computing, Zakopane, Poland, Springer (2004) 359ï£¡368*.

Osinski, S. and Weiss, D. (2008). Carrot2: Making sense of the haystack. In *ERCIM News*.

Qiu, F. and Cho, J. (2006). Automatic identification of user interest for personalized search. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 727–736, New York, NY, USA. ACM.

Sanderson, M. (2008). Ambiguous queries: test collections need more sense. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 499–506, New York, NY, USA. ACM.

Shen, X., Tan, B., and Zhai, C. (2005). Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 824–831, New York, NY, USA. ACM.

Sping, A., Wolfram, D., Jansen, M., and Saracevic, T. (2001). Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, pages 226–334.

Sugiyama, K., Hatano, K., and Yoshikawa, M. (2004). Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 675–684, New York, NY, USA. ACM.

Sun, J.-T., Zeng, H.-J., Liu, H., Lu, Y., and Chen, Z. (2005). Cubesvd: a novel approach to personalized web search. In *Proceedings of the 14th international conference on World Wide Web*, WWW '05, pages 382–390, New York, NY, USA. ACM.

Turney, P. D. (2001). Mining the web for synonyms: Pmi-ir versus lsa on toefl. In *In proceedings of the Twelfth European Conference on Machine Learning*.

White, R. W. and Roth, R. A. (January 2009). Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services, Vol. 1, No. 1*, pages 1–98.

Zamir, O. and Etzioni, O. (1999). Grouper: a dynamic clustering interface to web search results. In *Proceedings of the eighth international conference on World Wide Web*, WWW '99, pages 1361–1374, New York, NY, USA. Elsevier North-Holland, Inc.