An Exploratory Search User Interface Concept Supporting Vague Querying and a Novel Result Representation

Timo Lüddecke Markus Jüttner Marcus Nitsche Andreas Nürnberger

Data and Knowledge Engineering Group,
Faculty of Computer Science,
Otto von Guericke University Magdeburg
{timo.lueddecke, markus.juettner}@st.ovgu.de,
{marcus.nitsche, andreas.nuernberger}@ovgu.de

Abstract

Common search engines deliver quite good results when the user has a precise notion of what he is looking for. However, the user might have in mind additional prior information regarding the importance of specific terms. Consequently, it seems desirable to avoid the latter and incorporate the knowledge into the query instead. Therefore, we propose a search user interface concept that supports users in modelling their uncertainty in a comfortable way, foster exploratory search and provide a compact yet informative representation of results. An implemented prototype demonstrates the feasibility of the concept. We also present results of a first two-step usability study. The results indicate a good usability of the concept and show that even this novel concept meets user's expectations.

1 Motivation

Modern search engines have become very powerful tools, providing excellent results - even in areas beyond basic document queries like finding a nearby dentist or checking for the weather of next weekend. However, they require textual input of keywords by the user, who is not necessarily capable of formulating suitable terms at the begin of the search, for instance because he is new to the domain. Search engines also lack in the ability to support the formulation of importance of selected search terms as every term has potentially the same impact on the result (apart from context-sensitivity). Also functions for explicitly excluding terms are either hidden - in most cases unknown to users - or do not exist at all. A study on search query logs conducted by Jansen et al. [9] found that the boolean operators "NOT" or "-" were only used in 3.34% of all queries. However, we believe that term exclusion could turn out to be useful in a much higher number of cases. It appears that non-uniform term importance, especially exclusion are desirable in numerous scenarios, e.g. when searching for recipes with a favourite and another nice-to-have ingredient while being allergic to a third component. The weighting of a term does not necessarily encode it's (known) relevance by the user. It might also specify the user's (un)certainty about the suitability of single terms.

Another aspect of search engines is the presentation of results. It comprises in most cases just text containing the title, a text snippet and the URL. This gives no clue about the visual appearance of the actual document, which nevertheless could be helpful for a user's relevance estimation and recognition of previously visited websites.

In this paper, we present a concept designed to overcome these disadvantages of current web search user interfaces by introducing a novel query formulation mechanism and a compact representation of a web page's content as well as visual appearance. Both being integrated into a search user interface prototype that addresses some aspects of exploratory search [15] by providing support in expressing uncertainty. In [8]

In: U. Kruschwitz, F. Hopfgartner and C. Gurrin (eds.): Proceedings of the MindTheGap'14 Workshop, Berlin, Germany, 4-March-2014, published at http://ceur-ws.org

a brief overview of exploratory search tools and evaluation techniques are provided.

2 Related Work

The VIBE-system by [12, 17] also supports users in interactive finding and filtering relevant information. Here, magnets are used to attract relevant documents to specific screen points.

Nitsche & Nürnberger [16] introduced QUEST - a user interface concept where terms are placed radially around a center with the distance to it encoding the uncertainty: The closer a term is, the higher it's specific weight in the whole search query. Also results are represented by small dots or favicons in the radial layout, whereby the distance to the center maps the relevance of the current query constellation. As only the distance is taken into account, an arbitrary angle can be chosen without changing the semantics of a query formulation. Therefore, multiple arrangements of terms can encode the very same query, which might be a shortcoming of this approach. It generates also just a weak structure for user's decision which search result to survey first.

The problem of reducing a web page's content to a compact representation has been addressed in various publications [10, 19, 2]. These representations are - in most cases - based on a screenshot of the entire web page or an extraction of a salient region combined with the title, while being only remotely related to the textual content. Both, evaluations conducted by Dziadosz & Chandrasekar [6] and Aula et al. [2] suggest though that combining text and image enables the user to judge relevance best. Dörk et al. [5] presented an exploratory search environment with a result representation heavily relying on zooming in various maps: in temporal, spatial and semantic domain. However, with books, blogs and photos, only fairly structured types of content are considered – at least compared to arbitrary web pages.

3 System

Our system consists of three major components (Fig. 1): A crawler, a backend and a frontend. This section deals with the first ones, while the frontend will be described separately in section 4.

Crawler. The rich result representation prohibits the utilization of APIs of common search engines as they deliver too little information about the web page and crawling these in real time is infeasible given a reasonable number of results. Therefore, we developed our own crawler computing a colour histogram, a salient extract of the web page and a wordcloud as well as the text document for indexing. First, a screenshot of the web page is taken. For the colour histogram only

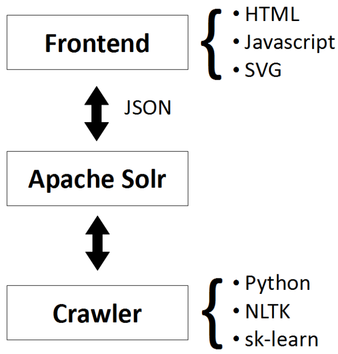

Figure 1: The three system components: Frontend, Backend (Apache Solr) and the Crawler.

the top 600 pixels of the web page are considered as most websites have a characteristic header. The pixel values are clustered by KMeans [14] into ten groups and stored in a database. The computation of a salient region is carried out by the algorithm of Achanta et al. [1], where saliency is defined as a pixel's distance to the image's mean colour in lab space. The saliency map is searched for areas of high saliency on multiple scales. The best candidate is selected and extracted from the screenshot. For text extraction the html content is first converted to plain text by nltk [3] and fed into the database. The text is further processed by removing stop-words and each remaining word is scored by it's frequency in the Brown corpus [7], which contains roughly one million words. The score is computed similar to tf-idf calculation [11] by the following formula:

$$score(w) = \begin{cases} tf(w)log(|\mathbf{B}|/tf_{\mathbf{B}}(w)), & \text{if } w \in \mathbf{B} \\ tf(w)(log|\mathbf{B}|+1), & \text{else} \end{cases} \quad (1)$$

With $\mathbf{B}$ being the set of words in the corpus and tf respectively $tf_{\mathbf{B}}$ the term-frequencies in the web page and in the Brown corpus.

Backend. Our database relies on Apache Solr without any profound modifications. Besides a typical text field, we added fields for additional features the crawler captures. Term weighting is implemented by the boosting mechanism of Solr. Communication with the javascript-based frontend is realized via HTTP and JSON encoding, which is natively supported by Solr.

4 Search User Interface

The user interface consists of five main elements, with query formulation and result representation as the most innovative ones. Query formulation is placed at the top of the screen and the result representation below. Both cover the entire width. Below them a navigable web page preview is set, surrounded by navigation buttons to the left and right. The small result

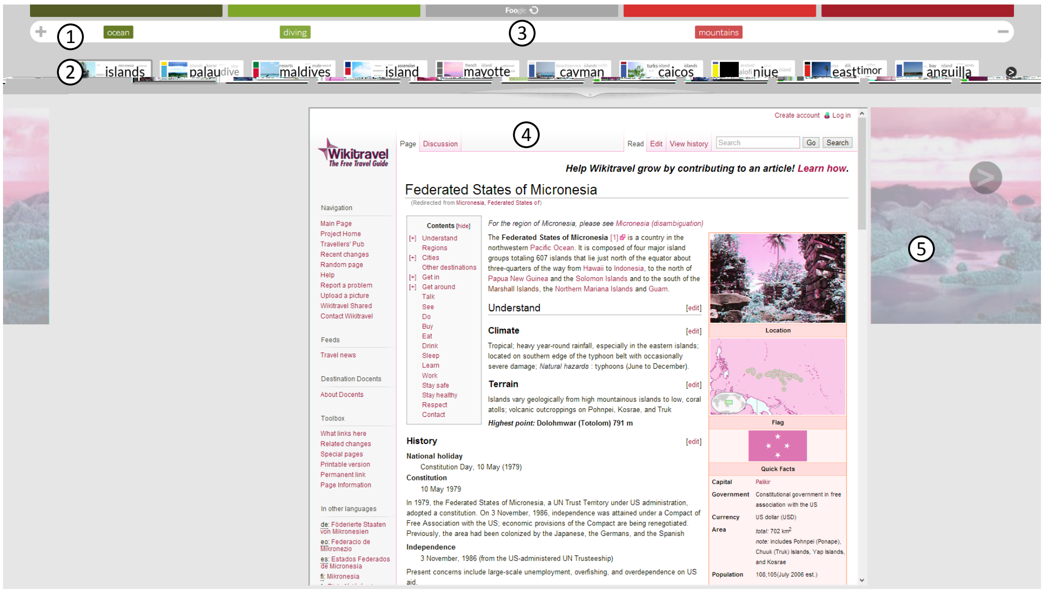

Figure 2: Screenshot of the prototype featuring (1) the search bar, (2) result representation, (3) a restart button, (4) the browsable web page preview and (5) preview for the next result in the list.

representation and the big preview follow the design pattern of "overview and detail" [4], while the navigable web page previews can be seen as contextual cues in the result space. To facilitate getting started and to ensure conformity with the user's expectation, just a simple common-known text-box is presented at start. After submitting an initial query, the layout transforms smoothly into the one shown in Fig. 2.

Query Formulation. In order to constitute a query, terms are first typed in the simple text-box as usual. After submitting the first query, all terms move to the left side without overlapping each other. Note that this implies a positive initial weighting. For a refinement and the expression of uncertainty, terms can be moved horizontally. Fig. 3 depicts how the arrangement of terms affects the query semantics. With x being the position of a term in the interval $[-1.0, 1.0]$, $sgn(x)$ indicates whether the term is explicitly wanted or unwanted in the result documents. $|x|$ denotes the confidence of the former statement. Single terms can be removed by triggering a small remove button that pops up on mouse over. New terms can be added by clicking the query bar at the position associated with the wanted weighting of the new term and simply starting to type. It is also possible to restart the entire search, i.e. to remove all terms, by clicking a single button.

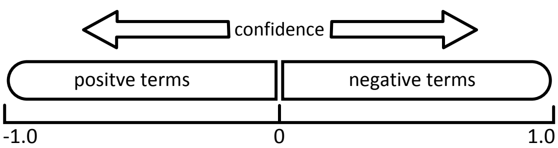

Figure 3: Query mechanism with "confidence": Weighting terms positive and negative.

Result Representation. Due to the elongated and – compared to common search user interfaces without preview – small result bar, the crucial goal in designing the result representation was to keep it as compact as possible and to allow a horizontal arrangement. Previously seen web pages should be recognizable and the content of unknown web pages should be as obvious as possible when looking at the result representation. Our approach consists of three different constituents (Fig. 4):

- A colour bar (1) on the left as well as the background of the whole element indicate frequent colours of the respective web page. When colours are known in advance, it allows to quickly redis-

Figure 4: Top: Conceptual representation with (1) four main colours (including background), (2) a salient extract of the web page's screenshot, (3) the word-cloud indicating a topic by presenting frequent words. Bottom: Two result representations from our crawler implementation. It can be seen how a web page's background colour affects the representation.

Figure 5: The query bar containing the terms of the second evaluation task.

cover a previously visited web page because colour is a pre-attentive attribute [13]. Otherwise, the bar at least provides useful cues on what to expect, e.g. websites for children are often very colourful while a business website is likely to have black text on a white background.

- An extract of the rendered web page's screenshot (2) provides a small preview of the most salient region of the web page. This might also support recognition and could additionally serve as a hint for the web page's topic.
- The wordcloud (3) - it's computation is described above - gives a general overview of the website's content by putting an emphasis on words that occur rarely in general but frequently in this document and are hence more likely relevant for the current topic.

By dragging a result representation into the query bar (Fig. 5), the query can be manipulated depending on the results content. When there is an intersection between wordcloud of the element and query terms, they are shifted to the left giving them a more positive effect. If there is no intersection, the most popular word in the wordcloud is added to the query. This way exploratory search is further supported in the proposed search user interface concept.

Implementation. The implementation of the described concept is based on current web standards: Javascript for the logic and SVG using raphael.js[1] for rendering graphics. Since the elements of the result representation are not retrievable from common APIs, we had to make a decision between re-crawling the elements in real-time as soon as results from an API are delivered or to build our own index with colours, salient region and wordcloud directly stored. We decided in favour of the latter as the re-crawling takes too much time and results could not be presented instantaneously. The index is based on an unchanged (except for configuration) Apache Solr[2] server. It is filled with content by our own crawler that captures a screenshot for colour and a salient region extraction as well as it processes the html content, ending up with term scores (see Formula 1) of the wordcloud. The crawler is implemented in python using nltk [3], scikit-learn [18] and various scripts, e.g. A. Müller's[3] for

[1]http://raphaeljs.com (28.10.2013)
[2]http://lucene.apache.org/Solr (28.10.2013)
[3]https://github.com/amueller/word_cloud (28.10.2013)

	vacation	hiking	mountains	alps	climbing
	5	7	4	1	0
	1	0	0	1	0
	1	0	1	2	1
	0	0	0	2	5

Figure 6: Results of the formative evaluation ("vacation" example): A dark box indicates a strong correlation between the respective values.

wordcloud rendering. We crawled two indices, a general one without restrictions (85 entries) and a special one with travelling and recipe sites only (455 entries), where the feature of vague query formulation is a big benefit.

An open question is how the system reacts in a larger scale, but as we use Solr for storage and query handling, we are confident that the system scales well, possibly by utilizing Apache Hadoop [20]. The user interface was not optimized to work in a mobile context like on a tablet. But due to the use of standard techniques it also runs on a Google Nexus 7 (2013) with only minor drawbacks.

5 Study Design

We conducted two evaluations with 17 participants in total, i.e. nine respectively eight participants each: A formative evaluation guided us for some design decisions. A summative one tested the final prototype implementation. Note that the evaluation was originally carried out in German and translated to English for this paper.

5.1 Formative Evaluation

The entire formative evaluation was implemented as an interactive form, where the study participants has been asked to interact with mock-ups of parts of the later implemented user interface. We offered a discrete and a continuous version of the query formulation mechanism (Fig. 5) and tried to assess which one is easier to handle. Therefore, we created two challenges:

Query formulation. The first task involved creating a query given the following brief note about the goals as well as the terms we wanted to be used: *You are looking for a destination for your hiking vacations in the mountains, not necessarily in the alps as you have been there before. As you suffer from vertigo you want to avoid climbing.* The results in Fig. 6 show that the test users were able to formulate a proper query, i.e. putting the relevant terms to the left side of the query and the negative ones to the right side.

Query understanding. To solve the second task, users were asked to do the inverse. Given a final query formulation, six different images needed to be ordered or removed. Five of them were images of cakes, the sixth one was an image of a dog. This way, we wanted to see if the representation of a query in the query bar (Fig. 5) is understandable. Furthermore, it gives insight to a deeper interpretation of the participants: Should the dog be in the result list though it has no relation to the query terms? If yes, should it be placed in front of the unfitting results? The results reported in Fig. 7 suggest that the basic principle has been understood as the rightmost images were correctly put top of the results in most cases.

Regarding the dog, the participants agreed on scoring it lower than all cakes. But there is a dissent on whether to include the result or to remove it.

Result representation. In addition to the query formulation, we also evaluated prototypes of the result representation (Fig. 4). Four manually assembled representations of web pages were provided and we asked the study participants for possible search terms, a category of the web page and which traits of the representation were pivotal for that decision. Not all participants filled out all fields. But when they did, they correctly predicted the web page's content, with only one exception. Often, the participants were able to specify even the subtopic.

5.2 Summative Evaluation

The summative evaluation was carried out by giving the participants some task, while observing them and making notices. Afterwards, they were given the opportunity to express feedback.

In general, most users succeeded in working with the search user interface. Minor problems involved confusion between user interface and result representation, colours and interpreting the plus/minus button at the end of the scale as being actually a button. We attribute this to the short time frame the participants had to get used to the prototype and its underlying novel concept. Colours in the result representation indicating page colours are confused with the colour scale for weighting a term. The plus respectively minus icon at the ends of the scale is sometimes mistakenly interpreted as a button.

						strawberry cream cake[4]
1st	3	0	0	0	4	2
2nd	0	1	1	0	3	4
3rd	4	3	1	0	0	1
4th	2	1	4	2	0	0
5th	0	3	3	1	1	0
6th	0	0	0	3	0	2
removed	0	1	0	3	1	0

Figure 7: Formative evaluation with "cake" example.

[4]Study participants saw copyright protected image.

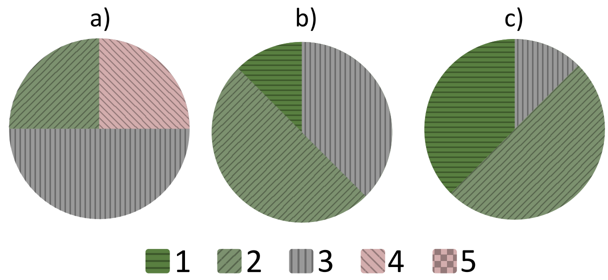

Figure 8: User ratings of our prototype on a five grade scale from strong green (good/easy) to strong red (bad/difficult) for the questions: a) How easy was the transition from common search engines?, b) How do you like the representation of the results?, c) How useful did you find the weighting ability?

Furthermore, we found that the search bar can be seen as a text field in the user's mental model and could therefore support corresponding interactions (i.e. placing a cursor and editing text). However, all participants considered the term weighting as a useful tool and the majority liked the result representation as well. The ratings shown in Fig. 8 indicate minor problems regarding the transition from common user interfaces while both, result representation and the weighting ability, are for most parts considered as good.

6 Conclusion

We presented a novel search user interface concept for exploratory web search addressing the problem of incorporating uncertainty with respect to user's confidence while searching. The main contributions are a novel query formulation mechanism and a compact visualization. This supports an efficient recognition. It also helps users to concern a web page's topic by linking visual and textual information. The implementation demonstrates the feasibility of the concept and the small evaluation suggests that users are able to properly interact with the interface.

Future work will cover the improvement of the system's usability in practice. For instance, by offering a function to save interesting web pages and using more elaborative methods for visual and textual information extraction in the crawler. The compact representation of results might also be interesting for mobile use, where screen space is limited.

Acknowledgement

We thank the flickr users tjstaab, freakgirl, lovebuzz, Kirti Poddar and jeff_ro for releasing their images under a creative commons licence.

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. Frequency-tuned salient region detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1597–1604, 2009.

[2] A. Aula, R. M. Khan, Z. Guan, P. Fontes, and P. Hong. A comparison of visual and textual page previews in judging the helpfulness of web pages. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 51–60, New York, NY, USA, 2010. ACM.

[3] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python.* O'Reilly Media Inc., 2009.

[4] A. Cockburn, A. Karlson, and B. B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1):2:1–2:31, 2009.

[5] M. Dörk, S. Carpendale, and C. Williamson. Fluid views: A zoomable search environment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 233–240, New York, NY, USA, 2012. ACM.

[6] S. Dziadosz and R. Chandrasekar. Do thumbnail previews help users make better relevance decisions about web search results? In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, pages 365–366, New York, NY, USA, 2002. ACM.

[7] W. N. Francis and H. Kucera. Brown corpus manual. *Brown University Department of Linguistics*, 1979.

[8] T. Gossen, M. Nitsche, S. Haun, and A. Nürnberger. Data exploration for bisociative knowledge discovery: A brief overview of tools and evaluation methods. In M. R. Berthold, editor, *Bisociative Knowledge Discovery*, volume 7250 of *Lecture Notes in Computer Science*, chapter Part IV, pages 287–300. Springer Berlin Heidelberg, 2012.

[9] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, 36(2):207–227, 2000.

[10] B. Jiao, L. Yang, J. Xu, and F. Wu. Visual summarization of web pages. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 499–506, New York, NY, USA, 2010. ACM.

[11] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

[12] S. L. Koshman. Vibe user study, 1997.

[13] H. Levkowitz. *Color theory and modeling for computer graphics, visualization, and multimedia applications.* Springer, 1997.

[14] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. L. Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[15] G. Marchionini. Exploratory search: From finding to understanding. *Commun. ACM*, 49(4):41–46, Apr. 2006.

[16] M. Nitsche and A. Nürnberger. Quest: Querying complex information by direct manipulation. In S. Yamamoto, editor, *Human Interface and the Management of Information. Information and Interaction Design*, volume 8016 of *Lecture Notes in Computer Science*, pages 240–249. Springer Berlin Heidelberg, 2013.

[17] K. A. Olsen, R. R. Korfhage, K. M. Sochats, M. B. Spring, and J. G. Williams. Visualization of a document collection: the vibe system. *Information Processing & Management*, pages 69–81, 1993.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, Nov. 2011.

[19] J. Teevan, E. Cutrell, D. Fisher, S. M. Drucker, G. Ramos, P. André, and C. Hu. Visual snippets: Summarizing web pages for search and revisitation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2023–2032, New York, NY, USA, 2009. ACM.

[20] T. White. *Hadoop: The Definitive Guide.* O'Reilly, first edition, 2009.