Query Suggestion for On-Device Troubleshooting

Frédéric Roulland, Stefania Castellani, Ye Deng, Antonietta Grasso and Jacki O'Neill

Xerox Research Centre Europe
6 Chemin de Maupertuis 38240 Meylan, France
{firstname.lastname}@xrce.xerox.com

Abstract. This paper describes a novel query suggestion tool we have designed and implemented to help users of office printing devices better formulate their queries, while searching a troubleshooting knowledge base provided as a service on the device itself. The paper traces the main motivations of the design of the query suggestion tool and outlines its technical details with an emphasis on its combination of features in relation to prior work.

1 Introduction and Motivation

This paper describes a novel query suggestion tool that we have designed to help users more easily formulate their queries when searching solutions while troubleshooting a printing device. The tool is part of an on-device support that lets users search a troubleshooting knowledge base (TKB). The problem that we are considering is one of search by non-experts in the specific technical domain of office device troubleshooting. To investigate troubleshooting practice, we carried out an ethnographic study of the call centre [1]. An immediate finding from the ethnography was that customers' telephones, for communicating with experts, and PC's, for online help, were rarely located by the printer/copier. It is an extra burden for customers to have to leave the ailing device to find out how to fix it. This inspired the design of a system to access the TKB on the device. The ethnography also revealed the work that customers and experts put in to co-construct suitable problem descriptions, suggesting that users are likely to have difficulties specifying their problems without expert help. There is a major terminology issue: customers do not always know the technical names of parts or how to best describe the problem they experience such that they can retrieve troubleshooting instructions from the TKB [9]. Initial designs to improve the online service included features designed to help customers better explore query results sets [10]. However, users still had problems formulating their initial query. Moreover, turning to designing on-device access to the TKB we also had to take into account a number of characteristics of its situated use. Although larger than normal copier/printer screens, the interface is relatively small. Sizing and spacing need to be set to 'finger-size'. In addition, a soft keyboard was required for entering queries. Together these conditions provided the inspiration for the query suggestion tool.

In this paper we present the details of the working of the tool with an emphasis on its combination of features in relation to prior work.

2 Related Work

The problem of how to support people while they search for information and they are not sure exactly what it is they are looking for has been studied both from the human (e.g. [2]) and the technology side. On the human side, it is important to consider the search domain. When considering searches of the web the quantity of content is such that almost any search will return results and the problem is one of appropriately filtering the results. In comparison, in domain specific databases, there may be much fewer and less diverse content and the problem of finding the right terminology becomes more central [3]. On the technology side, the predominant approach has been the one of Information Retrieval, based on the model of matching user's requests to the documents' content by using keywords. This model led to a first generation of systems used to access both the unstructured content of the Web and domain specific databases. However, a problem with these systems is that the users tend to be very generic in the formulation of their queries which often only contain two or three keywords [4]. One approach to address these issues is relevance feedback [5], which incorporates user feedback on the relevance of the retrieved documents and uses it to provide more precise answers. However, outside of the laboratory, this mechanism has drawbacks due to the lack of immediate incentives for the users to "instruct" the system to perform more precisely. Another approach is to complete query formulation through *query expansion* (e.g. [6]). However these systems, in an attempt of saving the user from additional effort, work as a black-box and remain too generic to really help the user in carving appropriate queries [7]. To go toward systems where the user is more active, recently there has been the development of *query suggestion* techniques. For example in [8] query suggestion has been tested on a domain specific search engine proving improved precision and recall of the results. Additionally, query suggestion, since it asks the user to pick just one of the suggested terms, seems to be more appropriate than query expansion when inter-document variation is small but the user has no prior knowledge of the terms used in the documents. This is the situation for example in the TKB in which editors enforce a controlled vocabulary for technical terms. Interestingly, several web search engines, e.g. Google, have introduced query suggestion mechanisms, taking the popularity of various searches into account in the ranking. However, little work has been done in investigating how to tune these mechanisms to domain specific contexts as our, which has led us to the design of a novel mechanism mixing popularity and domain vocabulary based suggestions.

3. Query Suggestion

The design of our query suggestion tool has been guided by the aim of improving the quality of the searches and reducing the burden of typing. Our approach can be summarized as follows: a collection of expressions candidate for suggestion is created from the TKB, so that only queries that would retrieve at least one result from the TKB are taken into account. They are then ranked according to the frequency of their occurrence in past queries and the relevance of the suggestions in the TKB, so that top

ranked suggestions are both likely to correspond to a valid user problem description and to provide good search results. The collection of query candidates for suggestion is created from the TKB contents, indexed using Natural Language Processing techniques to identify noun phrases, and the queries examined as containing a noun phrase used in the system. We decided to use the frequencies of queries in the ranking as a way to favour popular suggestions. Indeed, an analysis on a sub-set of queries of the TKB logs showed us that users can be observed to have some favourite ways of expressing their common problems. To compute the frequency of a suggestion we compare the set of lemmatized meaningful words of the expressions selected from the TKB with the set of lemmatized words computed for each past query stored in the logs. When the set of words of the expression is a subset of the set of words of the old query we increment its frequency. In addition, the content and structure of the TKB is also used to rank suggestions. Expressions that, if used as a query, would generate results scored highly by the search engine are favoured. For computing the score component of a suggestion a search using the suggestion is performed and the list of results for this search is retrieved. We simply use the highest score of the results returned by the search. This parameter balances the previous one and helps differentiate suggestions that cannot be effectively ranked from the frequencies in the logs.

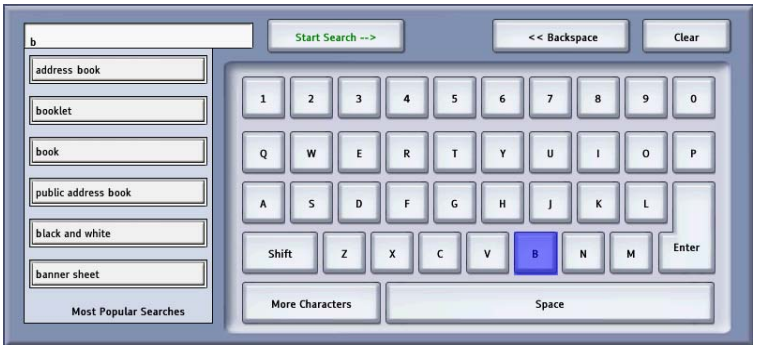

Fig. 1. The touch screen query interface.

The query suggestion tool is part of a troubleshooting system accessible on the device and assist users when they want to search the TKB. The query interface is shown in Figure 1. It consists of a simplified keyboard and, on the left side of the screen, a text box that displays the query typed by the user below which is the list of the "best" six suggestions provided by the tool. At any moment the user can select one of them from the list and the user input will be replaced with the selected suggestion. As in Google Suggest[1], when typing queries, the sequence of characters already typed in is used to filter the ranked collection of suggestions. The filtering we perform is however different from Web query suggestion systems, as it needs to point people toward expanded technical expressions they may not know or let them understand what they are using may be appropriate only in part. To achieve this goal, in our proposal, valid suggestions are ones that contain words starting with what is being typed or that contain the words already typed by the user. For example, if the

[1] http://www.google.com/support/websearch/bin/answer.py?answer=106230&hl=en

user is typing "li", suggestions could be expressions with one word starting with "li" like "light" but also "black lines". In order to implement the filtering, the query is tokenized into words which are then lemmatized. Then, a search is performed over the list of suggestions, returning every expression containing the set of words of the query. When the query contains several words and no suggestion is available for the whole expression, we propose suggestions for the longest, latest part of it. For example, if the user has entered "noise car" (for "noise in the cartridge"), and the system does not find any suggestions for the whole input then it looks for suggestions for "car" returning "cartridge", "toner cartridge", etc.

4 Conclusion and Future Work

We designed the query suggestion tool to help users formulate their queries by making the contents of the TKB available as users type queries and by reducing the burden of typing on the soft keyboard. Some preliminary tests of the tool, performed in order to both understand how quickly accurate query suggestions would appear on the interface and investigate the tool's usability, produced promising results and we are using them to refine the tool for which we will perform more extensive user tests.

References

1. Castellani, S., Grasso, A., O'Neill, J., Roulland, F.: Designing Technology as an Embedded Resource for Troubleshooting. Journal CSCW special issue on "CSCW, Technology and Diagnostic Work" (2009)
2. Crabtree, A., O'Neill, J., Tolmie, P., Colombino, T., Castellani, S., Grasso, A.: The practical indispensability of articulation work to immediate and remote help-giving. In: 20th CSCW, pp. 219—228, ACM, New York,(2006)
3. McCray, A., T., Tse, T.: Understanding search failures in consumer health information systems. In: American Medical Informatics Symposium, pp. 430—4, (2003)
4. Silverstein, C., Marais, H., Henzinger, M., Moricz, M.: Analysis of a Very large Web Search Engine Query Log. ACM SIGIR Forum, vol. 33(1), pp. 6—12, (1999)
5. Salton, G., Buckley, C.: Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, Vol. 41, No. 4, pp. 288-297 (1990)
6. Qiu, Y., Frei, H. P.: Concept Based Query Expansion. In: 16th SIGIR, pp. 160--9, ACM, New York, (1993)
7. Muramats, J., Pratt, W.: Transparent Queries: Investigating Users' Mental Models of Search Engines. In: 24th SIGIR, pp. 217—224. ACM (2001)
8. Feuer, A., Savev, S., Aslam, J. A.: Evaluation of phrasal query suggestions. In: Sixteenth ACM CIKM, ACM, New York, NY, pp. 841—8, (2007)
9. O'Neill, J., Grasso, A., Castellani, S., Tolmie, P.: Using real-life troubleshooting interactions to inform self-assistance design. In: Cotabile, M. F., Paterno', F. (eds.) INTERACT 2005. LNCS, vol. 3585, pp. 377—390. IFIP (2005)
10. Roulland, F., Kaplan, A., Castellani, S., Roux, C., Grasso, A., Pettersson, K., and O'Neill, J.: Query Reformulation and Refinement Using NLP-Based Sentence Clustering. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp.210—221. Springer, Heidelberg (2007)