Interactive Search Support for Difficult Web Queries

Abdigani Diriye†, Giridhar Kumaran‡, and Jeff Huang⋆

†University College London Interaction Centre, University College London, UK
‡Microsoft Corporation, Redmond, WA, USA
⋆ Information School, University of Washington, USA
a.diriye@ucl.ac.uk, giridhar@microsoft.com, ecir@jeffhuang.com

Abstract. Short and common web queries are aptly supported by state-of-the-art search engines but performance and user experience are degraded when web queries are longer and less common. Extending previous solutions that automatically shorten queries, we introduce searchAssist: a novel search interface that provides interactive support for difficult web queries. The query logs and questionnaires from a naturalistic study of 90 web users' search behaviors show that the usage rate of searchAssist for difficult queries was almost 40%. The results also highlight the importance of term dropping for long queries, and the improvements obtained in topical relevance when our searchers used searchAssist.

1 Introduction

Advancements in the field of Information Retrieval (IR) have brought about a rich information ecology that enables people to access information easily. A web search on Bing, Google and Yahoo! for the query *"britney spears"* reveals several million hits comprising different verticals and temporally fresh information. This query is an example of a popular or "head" query, which are typically short in length (≤ 3 terms long), and represent a significant portion of all queries found in query logs. Such queries are handled well by state-of-the-art web search engines because of the abundance of log data [2] and its conciseness enabling more effective information retrieval [6, 14]. A common strategy amongst commercial IR systems is a tendency to optimize for shorter and more popular queries as a way of improving the overall user experience. But the year-on-year increase in query length and infrequent queries [2] means this approach neglects a sizable portion of these *difficult* queries. This inevitably results in comparatively poor performance on these queries.

The goal of this work is to address instances where there is a lack of interactive search support for long and difficult queries. In Section 3 we briefly discuss the particular issues and typical existing types of support for these kinds of queries. We present our solution: searchAssist in Section 4, where we go over the design philosophy behind it as well as key features, and provide implementation details in Section 4.2. While there has been substantial work into long and difficult queries [3, 4, 6, 7, 13, 14], and numerous search interface features built to support different aspects of information-seeking (Section 2), ours is the first interactive search feature to simultaneously support term addition, deletion, negation and collation for long and difficult queries (Figure 1). We conducted a large scale naturalistic user study with 90 users (Section 5) that provided interesting insights into the relative utility and usage of different components of searchAssist. To complement this study, we also used questionnaires to gain a better understanding of user perceptions about searchAssist, detailed in Section 6.2.

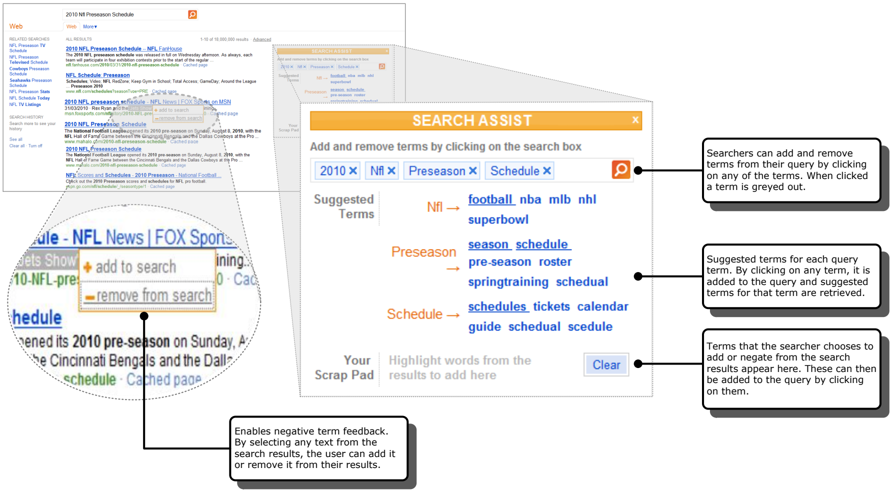
The main contributions of this paper are: (1) an analysis of the shortcomings of existing search interface features for difficult web queries; (2) a new rich interactive search interface that enables multiple search strategies with immediate feedback to guide the user; (3) a large scale naturalistic user study along with a log analysis of searchers' usage of our search feature, supplementary questionnaires that provide insights into user behavior, the usefulness of different search strategies, and implications for future design of interactive web search systems.

2 Related Work

Over the decades, a number of IR methods have been developed to help users reformulate their queries; these methods can be classified as either automatic or interactive methods [9]. Automatic methods like pseudo-relevance feedback, query reduction and query reweighting improve performance by leveraging statistical data to inform decisions about which query alterations to suggest. But the lack of user feedback during the reformulation process can alter the inherent meaning and context of the query [15]. Despite the advantages in retrieval performance of automatic methods [15], users tend to prefer interactive methods like suggested queries, real-time query expansion, popular search destinations and spelling corrections as it provides the users with more control during their search.

As a result of its capacity to leverage system generated information whilst eliciting user feedback and guidance during the search process, interactive IR methods have recently undergone a renaissance, and have received progressively more attention from the IR research community. Examples of this include Kelly et al.'s [11] lab-based user study examining participants' usage of query and term suggestions; White et al.'s [17] comparative lab-based user study involving a novel popular search destinations feature for specific queries; Anick's [1] naturalistic log-based study of searchers' usage of suggested queries; Koenemann and Belkin [12] and Ruthven's [15] empirical studies on the effectiveness of Interactive Query Expansion.

The majority of interactive IR features supporting query reformulation have provided support in the form of term addition, term substitution, term alteration, or new query reformulation [1, 11], but in spite of the fact query reduction and query rewriting have been reported to increase precision by up to 25% [3, 13], there has not been any interactive search support methods published that supports this query alteration method.


SEARCH ASSIST
Add and remove terms by clicking on the search box
2010 ✕ Nfl ✕ Preseason ✕ Schedule ✕
Suggested Terms
Nfl →
football nba mlb nhl superbowl
Preseason →
season schedule pre-season roster springtraining schedual
Schedule →
schedules tickets calendar guide schedual scedule
Your Scrap Pad
Highlight words from the results to add here
Clear
+ add to search
− remove from search

Searchers can add and remove terms from their query by clicking on any of the terms. When clicked a term is greyed out.

Suggested terms for each query term. By clicking on any term, it is added to the query and suggested terms for that term are retrieved.

Terms that the searcher chooses to add or negate from the search results appear here. These can then be added to the query by clicking on them.

Enables negative term feedback. By selecting any text from the search results, the user can add it or remove it from their results.

Fig. 1: Screenshot of searchAssist on a live web search engine.

3 Solutions for Long and Difficult Queries

Query reduction and rewriting methods are particularly effective for long web queries [3, 13, 14]. Long web queries are typically greater than 3 terms, and tend to perform badly on state-of-the art search engines because of their verbose nature which can skew the search result set towards an aspect of the query causing *query drift* [2–4, 6, 7, 14]. Similarly, difficult web queries retrieve an inadequate search result set which tends to have low levels of nDCG (nDCG@10 $\leq$ 0.35) [6, 7]. But, for difficult queries, the underlying problem is not the query length, but several factors like the query being too broad, or unsuitable query terms being selected by the user, or one or more of the query terms being misspelled. Generally, searchers address most of these issues by reformulating their query either by **manually** altering the query terms in the search box, or **interactively** using the suggested queries and terms offered by the system. Both query reformulation methods can present an unnecessary cognitive burden on the searcher during query reformulation, either through the act of manually selecting terms to alter and replace or deciding on what terms or queries to use. Also, in the case of interactive methods like suggested queries, there is a limit on the number of suggestions that can be leveraged during reformulation.

In this paper we propose a novel search feature called searchAssist that addresses some of the shortcomings of existing interactive methods, and provides rich interactive search support for long and difficult queries.

4 searchAssist

The design philosophy behind existing interactive search support features like suggested queries, affords little **control**, **freedom** and **interaction** between the system and user. This is obvious in the static manner they are presented. But, the real issue is that some of the most common search actions performed by searchers like term removal and addition are not properly supported by existing search features. Even in instances when they are used, there is significant cognitive workload involved in using them [8], and little guidance or indications to the kind of information they will retrieve is provided.

4.1 Design Process

To effectively support long and difficult web queries, we find these issues need to be resolved so better search support is provided during and after query reformulation. To design an interactive search feature that provides effective support, we identified a number of guiding principles from the literature [15, 16]:

- Enable actions that could improve queries;
- Lightweight seamless interaction;
- Unobtrusive: Appears only when needed, and easy to ignore;
- Intuitive: Easy to use;
- Provide guidance, context and immediate effect of selection .

With these guiding principles in mind, searchAssist was initially conceived and designed by brainstorming and identifying features that would be needed to provide effective support. Best design practises were followed [16], and search scenarios were created to identify how the search features would be used, and possible shortcomings. Mock-ups and paper prototypes were used to get a look-and-feel and foster a more fluid and iterative design process. The final design of our interactive search feature supports long and difficult web queries by making common search actions easy and intuitive to do, therefore increasing the likelihood of successful query reformulation. searchAssist enables richer manipulation of query terms by allowing users to not only add terms, but also to remove and negate terms interactively from their query. searchAssist allows searchers to navigate back and forth between queries, and to preview the search results selecting or removing a suggested term will retrieve. searchAssist features a rich set of suggested terms and support for term removal via the search box, and uncommon search actions such as collation of terms and negative term feedback (Figure 1).

4.2 System Overview

searchAssist is a subcomponent implemented in a large commercial search engine. searchAssist is surfaced to the user on the SERP (Search Engine Results Page) when our heuristic-based *difficult query spotter* identifies a web query that could be improved with interactive search support. The difficult query spotter is a sub-component of our system, and triggers searchAssist either when:

a) A query consisting of 4 or more terms has been entered;
b) When there are fewer than 5 relevant search results retrieved;
c) Or when there are fewer than 10 search results found.

In these cases, the query is considered long and therefore needs to be reformulated into fewer key terms, or difficult because it has retrieved a poor set of search results. To further identify when a set of search results are poor, we used

Query Quality Predictor [4] as a sub-procedure within our difficult query spotter. This predictor gauges the relevance of a set of search results by training a model using features derived from the query as well as the result set [4].

The suggested terms that are presented next to each query term were derived from our Suggested Term Index. The Suggested Term Index was constructed from synonyms found in WordNet and term alterations found from search engine query logs. These data sources produced over 5 million tuples of terms and their respective suggested terms. To ensure this data could be accessed efficiently in real-time, we reduced its size to just over 1 million tuples by removing stopwords, alphanumeric terms, and infrequent tuples.

4.3 Scrap Pad: Interactive Negative Query Term Feedback

The scrap pad feature enables interactive negative term feedback, a method of query reformulation that has been hardly explored. Belkin et al. [5], in one of the few studies on interactive negative term feedback, showed that it was both valued and used by searchers, but issues surrounded how the search feature was designed and presented. This search feature warrants further examination, and could prove instrumental as searchers are able to recognize relevant and non-relevant information more easily than recall it. Interactive negative term feedback has been incorporated into the scrap pad feature so searchers can remove terms that are not related to their information need; by simply clicking on any term from the search results, the searcher can negate it from their query.

One example where interactive negative term feedback might be useful is for the query *"SIGIR"*; this query retrieves search results for Information Retrieval and Special Inspector General for Iraq Reconstruction. If our searcher was exclusively looking for documents on Information Retrieval they could reformulate their query more effectively by using interactive negative term feedback to negate the term "Iraq" and reformulate their query to *"SIGIR -Iraq"*.

5 Naturalistic User Study

To understand how useful and effective searchAssist is in helping searchers reformulate queries, we conducted a naturalistic user study of its use. The study was primarily log-based, but supplementary questionnaires were also used to help triangulate and gain better insight into the log data. The questionnaire data was intended to elicit users' feelings towards searchAssist, and shed light on their query reformulation strategies. We opted for a naturalistic approach in this study as a way to gauge searchAssist's actual use in the "wild". Naturalistic studies are known to be more ecologically valid than lab-based studies as the observations are gathered from users' natural search setting rather than from the confines of a lab, which may produce artificial search behaviors unfaithful to a user's regular patterns of search. We provided access to potential users through an Internet Explorer plugin that directed all online searches towards our system. Embedded into the SERP HTML source was client-side JavaScript that helped aggregate our users' search behaviors by collecting all user interactions with the

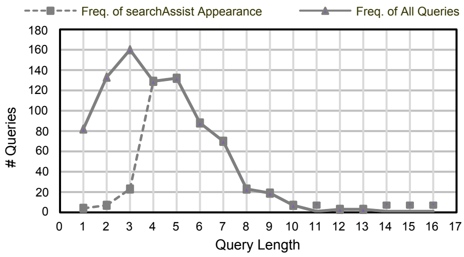


Fig. 2: Number of times searchAssist appeared for different query lengths.

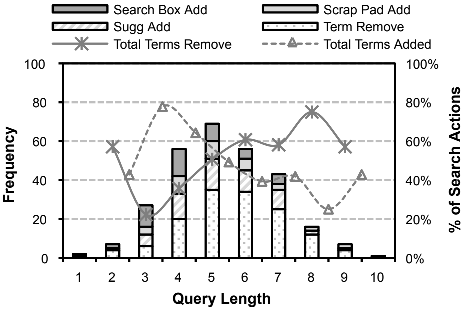


Fig. 3: Decomposition of searchAssist usage for different query lengths.

search engine. All the participants in this study were employees of a large IT company, and had a variety of job roles ranging from software engineers to administrators and management. All participants were based in the US and all were proficient in English. Though our users were not entirely representative of the general web search populace because of their above average technical proficiency, we still believe interesting insights can be found from the data. The participants for this study were recruited via e-mail; invitations to over 300 individuals were sent using several company mailing lists requesting them for their participation.

6 Results

In this section, we present the data from our query logs and questionnaire. From the logs, we identify users' search behaviors, and analyze how their usage of the search features changed with query length, and examine the kinds of terms searchers added or removed. From the questionnaire, we elicit qualitative feedback about searchAssist and users' reformulation strategies.

6.1 Log Data

All of our users' interactions with the SERP were recorded in our system logs over the length of the study. Our logs revealed 90 unique users conducting searches using our searchAssist enabled version of Bing. Over the 10 days, our 90 unique users formulated 848 queries of which 559 where manually formulated via the search box, and 289 queries were generated via searchAssist's query reformulation features. Throughout the study, 1,286 SERPs were served to our users by our system, and of these SERPs, searchAssist was triggered 56.3% of the time (724 times). From the 724 times it was triggered, we identified 289 instances when ours searchers used searchAssist (39.9%) for difficult web queries.

Queries and Search Feature Usage. The number of times searchAssist surfaced to the user is plotted in Figure 2 with the frequency for all queries. Unsurprisingly, because of our heuristics, we see in Figure 2 that our system had surfaced searchAssist infrequently for queries with less than 4 terms. In these cases, the search results are found to be adequate by our difficult query spotter, so search support is not offered to the user.

The mean average query length for searches where searchAssist was surfaced was 5.54 (*Mode* = 5; *Standard Deviation* = 1.94), compared to 4.19 (*Mode* = 3; *SD* = 2.28) for all queries submitted.

The usage of the searchAssist features shows that some features were more popular for difficult queries than others. Figure 3 shows term removal feature were the most used search feature. On one hand, this finding is intuitive as term dropping is an effective strategy for long queries [3, 6], yet surprising that a search feature that consumes very little interface real estate compared to the term addition features (i.e. search box, scrap pad and suggested terms) is so frequently used. In fact we actually see that term removal was used significantly more than any other feature, and almost as frequently as all the term addition features combined (142 vs. 147). We next move our attention to how our users' search behaviors changed with query length. More specifically, we will examine the relationship between the type of search feature used and the query length.

Impact on Query Length. We plotted search feature usage for each searchAssist query (Figure 3), transposing the percentages of term addition and removal taking place. Figure 3 shows that there is a statistically significant tendency by searchers to use the term removal feature as the query length increases. Conversely, the search features that enable term addition decrease with the increase in query length. As we have seen, users interact with the term removal feature more and the term addition features less as their query gets longer. We next look to analyze the effects searchAssist has had on our users' queries at the search session-level. We identified 76 search sessions from our logs, where a session is defined as whenever a user issues a new contextually unrelated query.

Based on these search sessions, we examined the change in query length when our searchers were using searchAssist. We compared the length of the initial query for each search session against the length of the other queries in the same search session (Figure 4). The graph shows that initially for sessions starting with a short query, users try and add more terms to their query, but a point of inflection occurs at search sessions with queries of initial lengths 6, and our users start reducing the length of their subsequent queries. To summarize this section, our data suggests that a relationship exists between the kind of search features used, search behaviors performed, and the characteristics of a web query.

Grammatical Tagging of Query Terms. During query reformulation, searchers invariably change terms to alter their query. In these two sections, we take a closer look at the kinds of terms our users added and removed from their query. The aim of these sections is to investigate the relationship between the type of search terms used and the search actions performed.

From our log data, we identified all the instances where terms were added or removed from queries using searchAssist. We initially divided the data into two bins: terms removed and terms added. We decided to categorize queries based on whether they were added or removed from a query because it changes the

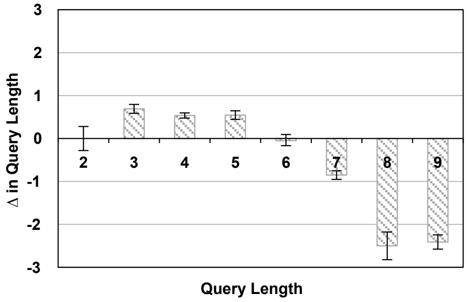


Fig. 4: Change in query lengths. Error bars denote standard error of the mean.

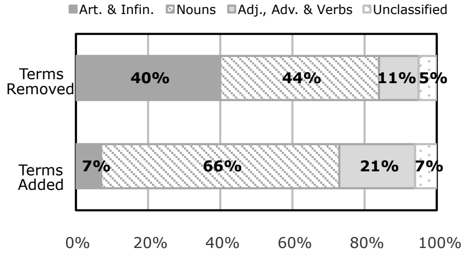


Fig. 5: Types of terms added and removed from queries. Differences statistically significant (χ^2, $p<.01$).

underlying intention behind the action. When terms are added to a query, the searcher is enriching their query by appending supplementary terms, whereas term removal is an indication that the user wants to broaden their query and remove additional terms. The queries from the logs were linguistically analyzed, and each query term was assigned a part-of-speech label.

Due to the sparsity of information in the queries, we were only able to construct a simple schema (i.e. nouns, verbs, adjectives, adverbs, articles and infinitives) for the part-of-speech labels we assigned the queries. In total 289 queries were manually labeled. 10 terms could not be tagged due to their ambiguity (e.g. kalo, sdf, etc.). In Figure 5, we see the compositions of the terms added and removed using searchAssist; this figure tells us that 40% of all terms removed involved articles and infinitives, compared to only 7% for term addition. In the context of web search, infinitives and articles (e.g. "a", "the", etc.) do not, for the most part, add any useful content; this is perhaps why it is removed so often by users and added rarely. We have also seen users adding more noun than verbs, adjectives adverbs and articles and infinitives to their query. This tells us that users selected noun terms (e.g. "britney spears", "iphone", etc.) as a method to further refine their query. This finding could inform what terms would be preferable to show searchers when they are refining their queries.

Analysis of Positive and Negative Interactive Term Feedback. We continue looking at the kinds of terms added and removed, but move our focus onto the negation and collation actions afforded by the scrap pad feature. Collation and term negation are two poorly supported search actions on most SERPs. In this section, we examine users' query reformulation habits to understand the kind of terms they added and negated from their queries when given the option.

Scrap pad was used significantly less than other search features using auto-generated content like the term removal or addition features (41 instances of scrap pad usage vs. 248 instances of term adding/dropping usage). This is likely a result of the workload involved in manually identifying and adding/negating terms. In the data, we see that positive term selection was more frequent than negative term selection (24 instances vs. 17). From our users' query reformulation behaviors using scrap pad, we have seen that negative and positive term played different roles in people's searching. The terms positively added by users were used to further elaborate and refine their query; these terms would be conceptually related terms and would provide a narrower and more focused conceptualization of their information goal. In one example a user enters *"r glm example matlab"* and identifies the term "Documentation" from the search results as a suitable term to use in reformulating their query. Traditionally, our user would have to manually add terms to their query, and not have the capacity to collate and store terms between query reformulations.

In contrast, the negative term selection feature was used to prune concepts from their search results that were not related to their query. For example, one user entered the query *"Enrique Iglesias Water Ski"*, and upon browsing the search results, they engaged in negative term feedback by selecting a term from the search results and negating it. In this instance, our user removes the term "Julio" from their query as search results relating to Julio Iglesias are present within the current search result set. In this example our user uses negative term feedback to further refine their search result set so undesirable topics are pruned.

Topical Relevance. Ideally, nDCG scores would be used to compare the improvement attained via searchAssist, but due to the nature of the study this was not possible. Instead, the improvement in topical relevance of the search result set for the searchAssist queries was performed.

This was done by comparing the topical relevance of the search result set for the initial queries with those generated using searchAssist. We sampled 100 random queries where searchAssist was used and labelled the queries based on whether they hurt, helped, or had no effect on the topical relevance of the search results against those for the initial query. The queries were labelled by two judges, and to attain a level of reliability the judge's labels were compared. Inter-coder reliability was performed to assess the accuracy of the topical relevance judgements; a Cohen's kappa (κ) of 0.81 was found which indicates a significant level of agreement between the judges. From our analysis we found that searchAssist improved the topical relevance for 19% of the queries, hurt 6% and had no effect on 75% of the queries. A look at the search sessions within these queries reveals 28.6% of the search sessions had an improvement in the topical relevance when searchAssist was used, and 71.4% resulted in no improvement.

6.2 Questionnaire Data.

Qualitative data was collected in the form of questionnaires to complement the quantitative data we had gathered, and to also gain further insight into how our users reformulate difficult web queries. The questionnaires were administered to a randomly selected sample of 10 users to identify their general disposition towards searchAssist, and help identify usability issues. The questionnaire collected data identifying what the users' typical query reformulations strategies were, and how useful searchAssist was in supporting them. It used a series of 7-point semantic differentials (higher values indicate greater levels of agreement), and open- and close-ended questions.

The results from the questionnaire relating to the usefulness of the search revealed that users rated searchAssist's term removal feature the most useful feature on the interface ($\underline{Mean}$ = 5.67) compared to the term addition from the suggested terms ($\underline{M}$ = 5.56), the search box ($\underline{M}$ = 4.00) or the scrap pad ($\underline{M}$ = 3.11, One-way ANOVA, $p < .01$). For difficult queries, the users rated searchAssist as being more useful than other existing search features ($\underline{M}$ = 2.44) like suggested queries ($\underline{M}$ = 2.22) and autocomplete suggested terms ($\underline{M}$ = 1.44, One-way ANOVA, $p = 0.016$). They identified that the autocomplete suggested terms were more useful during query formulation, but not for longer queries or during reformulation. In this case, suggested queries and searchAssist was perceived as more effective.

The open-ended questions identified a number of usability issues with searchAssist. This included a lack of contextually relevant and cohesive suggested terms, an overabundance of suggested terms being offered, difficulties in learning some of the more advanced features (i.e. scrap pad), and poor visibility of the scrap pad function. The users also identified a number of additional features they would like to have seen. They listed a preview of the search results being a desirable additional feature; some search-tips or explanatory text identifying what steps to take when searchers are using it for the first time. They also brought up the need for more context-sensitive suggested terms, and tuples of terms to be suggested for addition and deletion, e.g. "britney spears", instead of presenting "britney" and "spears" as two different terms to drop/add. There was also a need to leverage more computation to identify the importance of terms, and recommendations of terms to drop. Our users also recommended providing information about the kind of terms other web searchers dropped and the URLs that they clicked to aide them in their query reformulation.

7 Discussion

In this section, we discuss what has been learnt from this study, and its implications on the design of IR systems and future work. The findings from this study revealed a number of interesting insights: searchers were more amenable to interactive search support when dealing with difficult web queries. We recorded searchers interacting with searchAssist almost 40% of the time for these kinds of queries, and term dropping was observed as the most frequently used search action. Our data also showed that as the length of a query gets longer, the number of terms searchers drop also increases. Another interesting finding was the initial query length of search sessions with more than 6 terms reduced, whereas search sessions with fewer than 6 terms increased, also the usage of searchAssist resulted in an improvement in the topical relevance for 28.6% of our sample query sessions.

The data from the questionnaire also supports the notion that term dropping was the most useful feature for reformulating difficult queries. This is perhaps unsurprising as term dropping has been shown to improve information retrieval [3, 13]. We have also seen searchers prefer searchAssist over existing search features like suggested queries and auto-complete refinements for difficult queries.

As it stands, most IR systems do not provide rich interactive search support for long and difficult queries. As our findings suggest, efforts are clearly needed to interactively support more common search strategies like term dropping, term alteration and term negation on IR search interfaces. Work by Huang and Efthimiadis [10] on query reformulation strategies identified 13 possible search strategies employed by searchers during query reformulation such as spelling correction, query term replacement, term dropping, etc. Of these search strategies, we find that aside from term addition, term replacement and spelling correction, most search strategies are poorly supported on state-of-the-art IR systems. One step that can be taken to improve the usability of IR systems would be to support a wider variety of search strategies on the SERP. This might take the form of search features like searchAssist being incorporated into existing search interfaces, or perhaps embedding specific search strategies into the functionality of existing search features like suggested queries or autocomplete refinements. For example, for the query *“ECIR 2012”*, specific terms in this query can be struck through for the autocomplete refinements to show different options and alterations to the query like *“ECIR* ~~***2012***~~ *conferences”* or *‘European Conference* ~~***EC***~~ *IR 2012”*. Interfaces should allow searchers to interact with the terms on the page via features similar to those found on the scrap pad feature of searchAssist; search features that enable more user freedom would enable them to access additional terms and augment their query more freely.

While we learnt that searchers did not use the scrap pad feature as heavily as some of the search features using auto-generated content like suggested terms or term removal, the scrap pad enables interactive negative term feedback, a method of query reformulation that has been underexplored. This capacity might be incorporated into search interfaces by showing the searcher a list of conceptually different terms generated from their search result set. This search feature would enable them to negate irrelevant terms and concepts from the search result set, for example, for the query *“SIGIR”*, the suggested terms to negate, based on the search result set, might be: *“**–Iraq**”*, *“**–Information Retrieval**”*, and depending on whether they are looking for information related to the Special Interest Group in Information Retrieval or the Special Inspector General for Iraq Reconstruction, they might negate one topic or the other from their search result set to focus their attention on the relevant topic.

8 Conclusion

In this paper, we have reported on a naturalistic study of an interactive search support feature for difficult web queries. We have described the design and implementation of the search feature, and detailed the design of the study, the results and implications. Our findings revealed that richer forms of search support that allow searchers to add, alter and remove terms can be useful for difficult web queries. We have also seen that searchers adopted our search feature in their day-to-day online search activities for a significant portion of their difficult queries without any monetary incentive for usage. There was also tangible benefits in terms of topical relevance with 28.6% of search sessions experiencing an improvement in topical relevance. Amongst the users' search strategies, term dropping was revealed to be the most prominent search strategy for long and difficult. Also, the search feature was generally preferred by searchers for long and difficult queries over other search features like suggested queries, and auto-complete refinements.

This study has been the first step in interactive search support features for difficult queries. Our work complements the existing body of work on long and difficult web queries [3, 7, 6, 13, 14], and extends it by taking a user-centered angle on the problem and providing an analysis in the context of an interactive environment. This paper has detailed the usage and usefulness of this search feature, further work should look to leverage more system-oriented computation to enhance the interactive aspects of searchAssist, and possibly look into incorporating aspects of it into existing search interface features.

References

1. P. Anick. Using terminological feedback for web search refinement: a log-based study. Proc. SIGIR '03, pages 88–95.
2. P. Bailey, R. W. White, H. Liu, and G. Kumaran. Mining historic query trails to label long and rare search engine queries. ACM Trans. Web, 4:15:1–15:27, 2010.
3. N. Balasubramanian, G. Kumaran, and V. R. Carvalho. Exploring reductions for long web queries. Proc. SIGIR '10, pages 571–578, 2010.
4. N. Balasubramanian, G. Kumaran, and V. R. Carvalho. Predicting query performance on the web. Proc. SIGIR '10, pages 785–786, 2010.
5. N. Belkin, C. Cool, D. Kelly, S. Lin, S. Park, J. Perez-Carballo, and C. Sikora. Iterative exploration, design and evaluation of support for query reformulation in interactive information retrieval. Inf. Proc. Manag., 37:403–434, May 2001.
6. M. Bendersky and W. B. Croft. Discovering key concepts in verbose queries. In Proc. SIGIR '08, pages 491–498, 2008.
7. M. Bendersky and W. B. Croft. Analysis of long queries in a large scale search log. In Proc. of the 2009 workshop on Web Search Click Data, pages 8–14, 2009.
8. S. Dennis, R. McArthur, and P. Bruza. Searching the world wide web made easy? the cognitive load imposed by query refinement mechanisms., 1998.
9. E. Efthimiadis. Query expansion. Ann. Rev. of Info. Syst. and Tech., 31, 1996.
10. J. Huang and E. N. Efthimiadis. Analyzing and evaluating query reformulation strategies in web search logs. In Proc. CIKM '09, pages 77–86, 2009.
11. D. Kelly, K. Gyllstrom, and E. Bailey. A comparison of query and term suggestion features for interactive searching. In Proc. SIGIR '09, pages 371–378.
12. J. Koenemann and N. J. Belkin. A case for interaction: a study of interactive information retrieval behavior and effectiveness. In SIGCHI '96, pages 205–212.
13. G. Kumaran and J. Allan. Effective and efficient user interaction for long queries. In Proc. SIGIR '08, pages 11–18, 2008.
14. G. Kumaran and V. R. Carvalho. Reducing long queries using query quality predictors. In Proc. SIGIR '09, pages 564–571, 2009.
15. I. Ruthven. Re-examining the potential effectiveness of interactive query expansion. In Proc. SIGIR '03, pages 213–220, 2003.
16. J. Tidwell. Designing Interfaces: Patterns for Effective Interaction Design. O'Reilly Media, Inc., 2005.
17. R. W. White, M. Bilenko, and S. Cucerzan. Studying the use of popular destinations to enhance web search interaction. In SIGIR '07, pages 159–166, 2007.