Enhancing Case-Based, Collaborative Web Search*

Oisín Boydell and Barry Smyth

Adaptive Information Cluster
School of Computer Science and Informatics
University College Dublin, Dublin 4, Ireland
oisin.boydell@ucd.ie, barry.smyth@ucd.ie

Abstract. This paper describes and evaluates a case-based approach to personalizing Web search by post-processing the results returned by a Web search engine to reflect the interests of a community of like-minded searchers. The search experiences of a community of users are captured as a case base of textual cases, which serves as a way to bias future search results in line with community interests.

1 Introduction

Web searchers continue to struggle when it comes to efficiently locating precise information and recent evidence suggests that up to 50% of search sessions fail to deliver relevant results [20]. The types of queries used in Web search are a significant part of the problem due to *query ambiguity* and *vocabulary mismatches*. Web queries usually fail to clearly identify the searcher's true information needs and many studies have highlighted how a typical query contains only 2 or 3 terms [12]. For example, queries like *"jordan pictures"* offer no clues about whether the searcher is looking for images of the racing team, the middle eastern state, the basketball star, or the British celebrity. At the same time, recent evidence highlights the lack of correspondence between queries and target pages, suggesting that there is a vocabulary mismatch between search terms and index terms[5]; for example, [2] go so far as to dismiss the traditional view of there being a single conceptual space for information retrieval consisting of both query and document terms. As we shall see, encouraging users to submit more detailed queries is unlikely to provide a solution because such queries tend to exacerbate the *vocabulary gap*: users often add query terms that do not help to identify the document they are seeking.

The work of [1] described a case-based approach to Web search that involved maintaining a case base of *search cases* to reflect the combined search experiences of a community of like-minded searchers. Very briefly, each search case encoded the results that had been selected by community members in response to a particular query, q_i. When responding to some new target query, q_T, results

* This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361.

R.O. Weber and M.M. Richter (Eds.): ICCBR 2007, LNAI 4626, pp. 329–343, 2007.

that are contained within search cases for similar queries are *promoted* within the result-list returned by some underlying search engine such as Google. In this way, results that are preferred by community members are promoted in response to recurring future queries. So if our searcher is a member of a community of motoring fans then their search for *"jordan pictures"* is more likely to refer to pictures of the racing team, and such results will be promoted assuming that they have been frequently selected for similar queries in the past.

One of the potential shortcomings of the work of [1] is the limited *retrievability* of search cases. Search cases are indexed using the query terms that led to a particular set of result selections, and these cases can only be retrieved (and their associated results promoted) if there is a term-overlap between the target query and case query. However, because search queries tend to be short (typically 2-3 terms in length [12]) such overlaps cannot always be guaranteed. For example, staying with our motoring community, a target query for *"F1 photos"* would not see the retrieval of a search case for *"jordan pictures"* and so relevant promotion opportunities will be missed. In this paper we describe an alternative approach, which seeks to provide a richer set of retrieval opportunities for search cases. Instead of indexing cases by their queries alone we describe how cases can also be indexed by the terms that occur in the snippet texts associated with the selected results. In this way case specifications contain a much richer vocabulary and offer far greater opportunities for retrieval and reuse during future searches.

The remainder of this paper is organised as follows. In the next section we motivate our work by highlighting the extent of the vocabulary gap in Web search. This is followed by a review of related work, focusing on recent attempts to bridge the vocabulary gap by harnessing context and experience within Web search. In particular, we review a number of case-based approaches to Web search that form the starting point for our own work. Sections 4 and 5 go on to describe and evaluate our approach to personalizing Web search across a number of different communities with a comparison to two separate benchmark search services.

2 How Wide Is the Vocabulary Gap in Web Search?

Further evidence in support of the vocabulary gap comes in the form of the recent emergence and popularity of tagging services such as *Del.icio.us*[1] and *Shadows*[2]. Such services allow users to explicitly *tag* Web pages with terms of their own choosing and they provide users with various ways to recall tagged pages. Thus, these services provide an alternative way for users to locate relevant and interesting pages. For example, *Del.icio.us* is a collaborative bookmarking service that allows users publish their tagged bookmarks online.

The availability of this tagging data can be considered as the basis for an experiment to estimate the extent of the vocabulary gap in Web search. It seems reasonable to view the set of terms used to tag a bookmarked page as a proxy for a search query that the user might submit to a search engine to locate this

[1] http://del.icio.us/
[2] http://www.shadows.com/

page. Indeed when we analysed 7692 *Del.icio.us* bookmarks and tags, we found that the tags share many of the same basic term distribution characteristics as search queries, such as average length, expected overlap, etc. This begs the question as to how Web search engines might respond to ambiguity and vocabulary problems among these 'queries': will they tend to return the bookmarked page, for example? To test this we submitted the 7692 queries to Google, Yahoo, and MSN search and noted whether their corresponding bookmarked pages occurred within the top ten pages returned. The results are presented in Figure 1 as a graph of the percentage of search sessions where the bookmarked result was located against different sizes of queries from the test set.

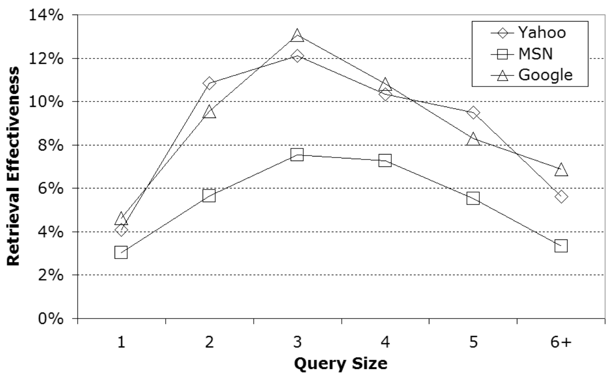


Fig. 1. Retreival Effectiviness vs. Query Size

There are a number of interesting observations to be made from these results.

1. The leading search engines only retrieve the target pages among their top ten results less than 15% of the time, with Yahoo and Google outperforming MSN Search.
2. All of the search engines achieve their maximum retrieval effectiveness for 3-term queries, which corresponds closely to the average size of the typical Web search query [12], suggesting that they are at least well-adapted to modern search queries.
3. Retrieval effectiveness degrades for longer, less ambiguous queries (the vocabulary gap at work) demonstrating a tendency for users to draw on increasingly less useful terms as part of more elaborate queries. Thus, retrieval performance is unlikely to be enhanced by encouraging Web searchers to extend their queries unaided.

3 Related Work

Primarily, this work is motivated by the need to bridge the vocabulary gap that obviously exists in Web search. Specifically we need to look for ways to improve how Web search engines cope with vague user queries and vocabulary mismatches.

3.1 Context Sensitive Search

Vague queries are often problematic because they lack context; to re-visit our previous example, the query *"jordan pictures"* does not help to distinguish between pages relating to motor racing, basketball, the country or the celebrity, any of which might be relevant to the searcher. One way, therefore, to improve a vague query is to expand it by including additional context terms. This can be done according to two basic approaches: either by explicitly establishing context up-front or by implicitly inferring it. For example, the Inquirus 2 meta-search engine [8] supplements keyword-based queries with a context category; users explicitly select from a set of categories such as "research paper", "homepage" etc. Alternatively, implicit context can be automatically inferred. Systems such as Watson [3] take advantage of user activity prior to the search to judge context; Watson monitors a user's word processing activities and uses document text as the basis for query terms. The interested reader is also referred to the Remembrance Agent [16] and Letizia [14].

3.2 Query-Log Analysis

Query-log analysis resonates well with a case-based approach to Web search, in the sense that it considers the value of historical search session information contained within query logs. For example, [6] mine a search engine's query log in order to discover correlations between query terms and document terms, which can then serve as candidate expansion terms as part of a query-expansion technique. The basic idea is that, if a set of documents is often selected for the same queries, then the terms in these documents must be strongly linked to the terms in the queries. Although this technique focuses on query-expansion rather than result re-ranking, it is similar in spirit to Collaborative Web Search which our work is based on.

3.3 Early Case-Based Approaches to Web Search

The use of case-based methods in information retrieval tasks has a long history. For example, the work of Rissland [18] looks at the application of CBR to legal information retrieval, and [4] describe a case-based approach to question-answering tasks. Similarly, in recent years there has been considerable research looking at how CBR techniques can deal with less structured textual cases. This has led to a range of so-called *textual CBR* techniques [13]. However these approaches have all tended to focus on particular application domains for textual CBR rather than the broader area of Web search. In the context of Web search, one particularly relevant piece of work concerns the *Broadway* recommender system [10], and specifically the Broadway-QR query refinement technique that uses case-based techniques to reuse past query refinements in order to recommend new refinements. Briefly, Broadway's cases reference a precise experience within a search session and include a problem description (made up of a sequence of behavioural elements including a sequence of recent queries), a solution (a new query refinement configuration), and an evaluation (based on historical explicit user satisfaction ratings when this case was previously recommended). The work of [9] apply CBR techniques to Web search in a different way. Their *PersonalSearcher* agent combines user profiling and textual case-based reasoning to dynamically filter Web documents according to a user's learned preferences.

3.4 A Review Collaborative Web Search

The work presented here is most directly influenced by the work of [1], on *Collaborative Web Search* (CWS), which adopted a case-based approach to personalizing search for communities of like-minded users. Very briefly, CWS is a form of personalized meta-search [7] with two novel features. First, personalization occurs at the level of a community of like-minded searchers. For a given target query q_T, the results returned by some underlying search engine(s) are modified so that those results which are most likely related to the learned preferences of the community are *promoted*. Second, personalization is based on the reuse of previous search sessions: the promotions for q_T are those results that have been previously selected by community members for queries that are *similar* to q_T.

$$c_i = (q_i, (r_1, h_1), ..., (r_k, h_k)) \tag{1}$$

Each community of searchers is a associated with a case base of *search cases* such that each case, c_i, is represented as a $k + 1$-tuple made up of the query component (a set of query terms, q_i used during some previous search session) plus k result-pairs; see Equation 1. Each result-pair is made up of a result page id (r_j) and a hit count (h_j) and reflects the number of times that a given community has selected r_j in response to q_i. In this way, each search case is a summary of the community's search experience relative to a given query. The *problem specification* part of the case (see Equation 2) corresponds to the query terms. The *solution* part of the case (see Equation 3) corresponds to the result-pairs; that is, the set of page selections that have been accumulated as a result of past uses of the corresponding query.

$$Spec(c_i) = q_i \tag{2}$$

$$Sol(c_i) = ((r_1, h_1), ..., (r_k, h_k)) \tag{3}$$

Each new target problem (corresponding to a new query q_T) is used to identify a set of similar cases in the case base by using a term-overlap similarity metric (such as that shown in Equation 4) to select the n most similar search cases $(c_1, ..., c_n)$ for q_T.

$$Sim(q_T, c_i) = \frac{|q_T \cap Spec(c_i)|}{|q_T \cup Spec(c_i)|} \tag{4}$$

These search cases contain a range of different result pages and their selection frequencies. Bearing in mind that some results may recur in multiple cases, the next step is to rank order these results according to their relevance for q_T. Each result p_j can be scored by its *relevance* with respect to its corresponding search case, c_i by computing the proportion of times that p_j was selected for this case's query q_i, as shown in Equation 5.

$$Rel(r_j, c_i) = \frac{h_j}{\sum_{\forall k} h_k \epsilon c_i} \tag{5}$$

Next the relevance of a result with respect to the current target query q_T is calculated by computing the weighted sum of the individual case relevance scores, weighting each by the similarity between q_T and each q_i. In this way, results which come from retrieved cases $(c_1, ..., c_n)$ whose query is very similar to the target query are given more weight than those who come from less similar queries; see Equation 6.

$$WRel(r_j, q_T, c_1, ..., c_n) = \frac{\sum_{i=1...n} Rel(r_j, c_i) \cdot Sim(q_T, c_i))}{\sum_{i=1...n} Exists(r_j, c_i) \cdot Sim(q_T, c_i)} \tag{6}$$

In this way, for given user u, a member of some community C, with query q_T we produce a ranked list of results R_C that come from the community's case base and that, as such, reflect the past selection patterns of this community. In parallel, q_T is used by a meta-search component to retrieve a set of *traditional* search results, R_M, from some underlying search engine(s). Finally, R_M and R_C are combined and returned to the user as R_T. This combination typically involves promoting prominent results in R_C ahead of those in R_M; for example, typically the top 3 results from R_C are promoted ahead of R_M results to the user while other results from R_C are marked as community-relevant within the final result-list, which follows the original R_M ranking. In this way, results that have been previously preferred by community members are either promoted or marked as relevant to provide community members with more immediate access to results that are likely to be relevant to their particular needs.

3.5 From Selections to Snippets

The work of [1,21,20] has shown that *CWS* can be effective in search scenarios where natural communities of searchers can be identified, but its case-based approach is ultimately limited to the promotion of previously selected results. Thus, *CWS* relies on *what* searchers have selected in the past rather than *why* they made their selections, an important limitation that motivates a new approach to collaborative Web search presented in this paper. While still fundamentally experience-based, we describe an alternative model of case representation, indexing, and retrieval that offers a much greater potential to influence future searches. In particular, we attempt to capture *why* a certain result has been selected by a community member by mining the terms that appear in selected result snippets (the short query-focused document summaries that are associated with documents in search engine result-lists). These terms then provide a much richer opportunity to index search cases than queries on their own.

As an aside, the use of snippets for document indexing in IR was first suggested in 1958 [15], and more recently by [19]. These works propose the use of generic document summaries as an alternative index to be queried in parallel to a full content index or for use as a source for pseudo relevance feedback. Our approach differs in its use of query sensitive snippets and the importance of user selection behaviour (real search experiences) when it comes to combining such snippets community-sensitive search cases. Alternatively, the work of [11] on *document transformation* modifies document indexes according to previous selection behaviour, but in a more limited way to our proposal: query terms are simply added to the default index for a selected document to boost the weight of these terms in the document. Over time, this allows the document to drift towards the query terms for which it was selected in the past. In our work a search case base corresponds to a community-level index, which can be updated separately by using the snippet terms of selected documents as well as the query terms that led to their selection.

4 A Snippet-Based Approach to Case-Based Web Search

The main contribution of this paper is an alternative approach to case-based Web search, which is inspired by the CWS model. We continue to encode the search experiences of a community as a case base of search cases, however, this time there are two important differences. First, each case now reflects the selection behaviour of the community with respect to a single result page, rather than a single query. Second, each case is indexed according to two separate sets of terms, the query terms (as in the traditional model of CWS) but also the snippet terms that were associated with the result when it was retrieved. By their nature, these snippet terms are likely to have played some role in attracting the attention of the searcher. In the following sections we will describe how community search behaviour is used to generate these so-called *snippet cases* and how these cases are retrieved and reused when responding to a new target query.

4.1 Snippet Surrogates as Cases

Let (C, u, q_T) denote a search for query q_T by user u in community C. Consider some result r_j selected in response to such a search. This result will have been accompanied by a snippet in the result-list that was presented to the searcher and we can reasonably assume that this snippet, $s(r_j, q_T)$, must have contained terms $t_1, ..., t_n$ that were relevant to the searcher's needs. Therefore these terms can be used to index future retrievals or r_j. In short, we create a search case whose solution is the result r_j and whose specification contains the queries that led to its selection (as in standard CWS) plus the terms that occurred in the snippets that led to these selections.

More generally then, a result r_j, which has been selected for a number of different queries, $q_1, ..., q_n$, will be associated by a number of different snippets, $s(r_j, q_1), ..., s(r_j, q_n)$. Then each search case will include these queries and the terms from these snippets, to capture the community's overall experiences as they relate to r_j. In this way each search case now includes the following components: (1) a solution, in the form of a selected search result r_j; (2) a set of queries, $q_1, ..., q_n$, that have led to r_j being selected; and (3) the union of the snippet terms. As in the traditional model of CWS, in each case, every query q_i is associated with h_i, the number of times that r_j has been selected for q_i. In addition, each snippet term t_i is associated with a frequency count f_i that counts the number of occurrences of this snippet term across the various snippets $(s(r_j, q_1), ..., s(r_j, q_n))$ that make up the selection history of r_j; see Equation 7. These hit counts and frequency counts will be used as part of the ranking procedure when it comes to retrieving cases and ranking result pages in response to a target query, as discussed in the following section.

$$c(r_j) = (r, (q_1, h_1), ..., (q_n, h_n), (t_1, f_1), ..., (t_m, f_m)) \tag{7}$$

In this way, a given result page will be represented very differently in the search cases of different communities of users. For a start, each result will be indexed under different sets of queries, as in the standard model of CWS, to reflect the retrieval patterns of each community who has selected it. But in addition, according to our new snippet-based approach, each result will now also be indexed under the terms that occur frequently within the snippets shown for this result during retrieval. Because these snippets are query-sensitive they too will tend to reflect the preferences of particular communities. For example, a given community might use queries that probe a particular section within a given result page and so, for this community, this page will come to be indexed under the terms that occur within that particular section. In this case the document in question will be promoted in a limited set of circumstances. In contrast, if the same page is more broadly applicable to a different community, then it's snippets will tend to be extracted from a greater range of the page's contents and so its search case base will come to index this page under a broader set of snippet terms. In this case the page in question will be promoted for a much broader set of retrieval scenarios.

4.2 Ranking and Promotion

As in standard *CWS*, the final result-list R_T is made up of the set of meta results R_M and a complementary set of promotions R_C. As in CWS R_C is produced by retrieving relevant cases from the search case base, using the current target query q_T as a retrieval probe. This time, instead of comparing q_T to search cases only indexed by previous successful queries we can also compare q_T to the snippet terms of search cases as an alternative route to retrieval.

Thus, each search case c_j (representing the selection history of community for some result page r_j) is scored according to the *relevance* metric shown in Equation 8. As presented this metric is made up of two separate relevance components. First, similar to standard CWS, we compute the weighted relevance score for r_j with respect to a set of similar queries (queries which share terms with q_T), based on the proportion of times that r_j has been selected in response to each similar query (see equations 4 and 5); note that the notations used for

Sim and *Rel* in Equation 8 have been modified slightly because of the different case representation used in the current snippet-based approach.

$$SRel(c_j, q_T) = (1 + \sum_{i=1}^{n}(Rel(r_j, q_i) \cdot Sim(q_T, q_i))) * TFIDF(c_j, q_T) \qquad (8)$$

Secondly, r_j is scored, relative to q_T, based on the snippet terms encoded in c_j. Specifically, in the current implementation, we use a standard TFIDF (*term frequency, inverse document frequency*) term weighting metric commonly used by the information retrieval community. Very briefly, the TFIDF score of a term t_i with respect to the result page r_j, is calculated by dividing the frequency count of the term for c_j by the frequency count of the term across the case base as a whole. Thus, a higher score is given to those terms that occur frequently in a particular case but which are relatively rare among the snippet terms of other cases.

A more detailed account of TFIDF weighting is beyond the scope of this paper by the interetsed reader is directed to the work of [17]. For the purpose of the current work it is sufficient to understand that the TFIDF contributes an additional relevance component based on the relative frequency of snippet terms which overlap with the target query. Thus, result pages which are frequently selected for similar queries and whose snippets contain frequently recurring target query terms that are otherwise rare in the case base as a whole, will be ranked highly in R_C. Importantly, results that have never before been selected for q_T, or queries similar to q_T, may still come to be promoted if they have a high enough TF-IDF score, for example. As in the standard implementation of *CWS*, our promoted results R_C are returned ahead of the meta-search results R_M to produce the final results list R_T for the user.

5 Evaluation

So far we have described an approach to manipulate the results returned by a Web search engine so that they are better aligned to the learned preferences of the searcher's community. Our case-based approach is unique in the way it attempts to learn more about a community's implicit preferences by mining the terms that tend to occur within the snippets of selected results. In this section we seek to evaluate our research by comparing our snippet-based approach to the standard *CWS* and a leading search engine across four different communities of searchers.

5.1 Experimental Data

Ideally we would like to evaluate our techniques using real search data. Unfortunately the availability of comprehensive search logs, with query and selection information, is extremely limited, and so we have adopted an alternative strategy. As discussed previously, bookmarking services such as *Del.icio.us* can provide a reasonable source of search-like log-data if we interpret bookmark tags as queries for specific bookmarked documents.

In addition, it is possible to extract communities of 'searchers' from *Del.icio.us* by following sequences of related tags and extracting the bookmark data associated with these tag sequences. For example, consider the construction of an *iPod* community by starting with *'ipod'* as a seed tag. From this tag we can extract the top k ($k = 100$) bookmarked pages; for example, *'50 Fun Things To Do With Your iPod'* is the top page for the *ipod* tag at the time of writing. This page has been bookmarked by in excess of 1000 people and we can extract the tag-sets used to tag it, for a subset of u users; we extract the tag-sets for the first $p\%$ of all users who bookmarked the page. Thus, for example, one particular user has tagged the above page with *'ipod fun hacks'* and so this tag-set and page becomes a *query-result* pair in our *iPod* community. For each seed tag we can also get a list of related tags from *Del.icio.us* to expand the community and collect a new set of bookmarks. In this way we can, for example, expand the original seed to produce new tags such as *'ipod mp3'*, *'ipod apple'* or *'ipod hacks'*. We have used this community extraction technique to build four different communities of varying sizes from the *Del.icio.us* service as shown in Figure 4(a).

	Skiing	**iPod**	**Travel**	**WebDev**
Users	627	1287	1773	2206
Unique Queries	649	1485	1729	2545
Unique Pages	313	576	1025	1050
Selections	1042	2643	3218	3938
Queries/User	1.66	2.05	1.82	1.79
Queries/Page	2.07	2.58	1.69	2.42

Fig. 2. Community Statistics

5.2 Systems and Setup

Our evaluation uses Yahoo as the underlying search engine. Over this we implemented two case-bases systems: a standard version of *CWS* [20] and the new snippet-based approach. Thus we can compare three separate search services: (1) basic Yahoo; (2) *CWS* (with Yahoo); (3) *Snippet* (with Yahoo).

We randomly split the *query-result* pairs extracted from *Del.icio.us* in half to produce disjoint sets of training and test data and all results reported below are averaged over 10 such splits. The training data is used to build the necessary *CWS* and *Snippet* communities by 'replaying' the query-result pairs through *CWS* and *Snippet* as searches. Thus each community's case base was updated to reflect the selection of each result for its corresponding query and, in the case of *Snippet* each result was also represented by its corresponding snippet terms, generated using the Lucene snippet generator[3].

In this evaluation we are primarily concerned with overall *retrieval effectiveness*: the ability of a search engine to retrieve a target result for a given query.

[3] http://lucene.apache.org/

Using the 50% of search data that we held back as test data, for each query-result test pair we check to see if the target result (bookmarked page) was retrieved for the target query (bookmark tags) within the top 10 results. In this way retrieval effectiveness is expressed as the percentage of test searches for which the appropriate page was retrieved among the top 10 results. We will also look at the position of the test page within the result list.

5.3 Overall Retrieval Effectiveness

The overall retrieval effectiveness results are presented in Figure 3 (a). Each line refers to one of the evaluation systems and each data-point refers to the overall retrieval effectiveness for that system for a given configuration of promoted and baseline results. For example, the configuration $3 + 7$ refers to a maximum of 3 promoted results and 7 Yahoo results. Obviously the overall effectiveness of Yahoo remains unaffected (at 7.72%) by the promoted results, but we can see how *CWS* and *Snippet* systems benefit greatly from the availability of promotions. The *CWS* system can retrieve the target page within its top 3 results more than 3 times as often as Yahoo, and an even more significant benefit is seen for the *Snippet* system with retrieval effectiveness of over 4.5 times that of Yahoo. These benefits are largely due to the top 3 promotions and further promotions do not result in additional improvements in retrieval accuracy; this plateau effect is probably a result of our evaluation methodology as we are looking for the occurrence of a specific result in each search session and do not choose to assess the potential relevance of other promotions.

5.4 A Community-Based Analysis

In this section we take a more detailed look at retrieval performance on a community by community basis. Figure 3 (b) shows the average number of promotions per test query for *CWS* and *Snippet* for each of the test communities when a maximum of 10 promotions are allowed. As expected the *Snippet* system is capable of producing more promotions (8.5 on average) than the *CWS* system (7.7). But how relevant are these promotions to each community? We see from the results above that overall the *Snippet* promotions are contributing more positively

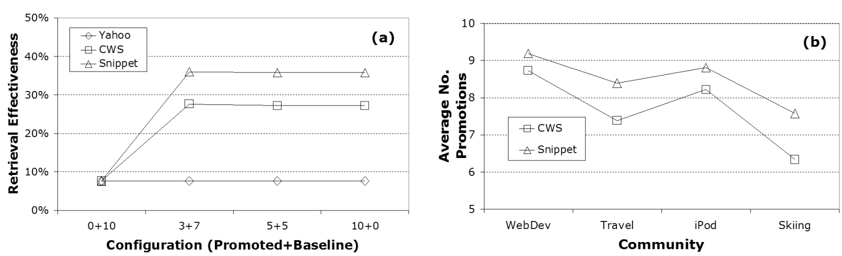


Fig. 3. The (a) overall retrieval effectiveness and the (b) number of promotions per community

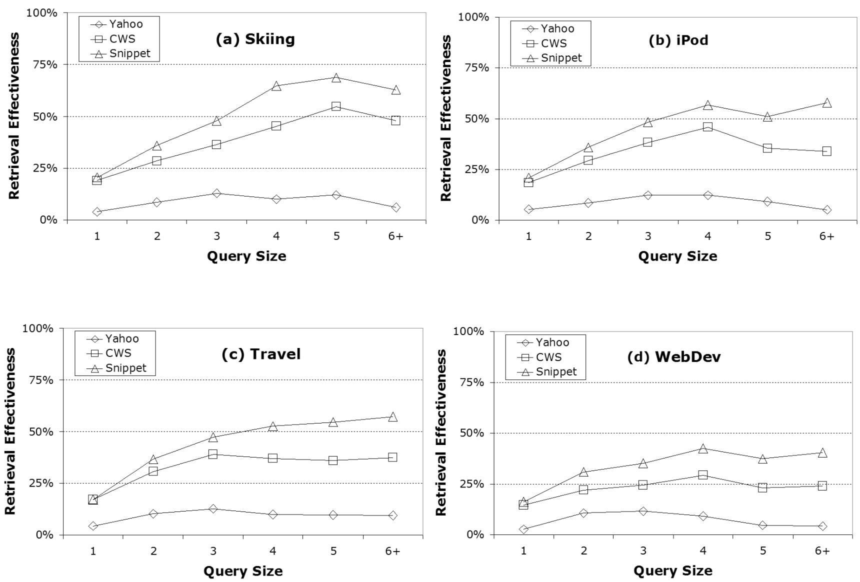


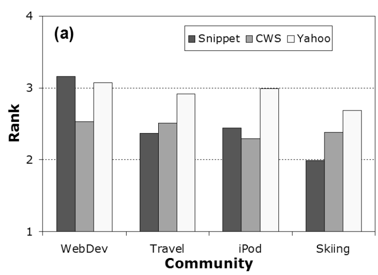
Fig. 4. Community Retrieval Effectiveness

to retrieval effectiveness than the *CWS* promotions but is this effect consistent across all communities? We know that the target result is likely to be one of the top 3 promotions, and for this reason, we will limit our next experiment to a 3+7 configurations (3 promotions plus 7 Yahoo results) and measure retrieval effectiveness on a community by community basis.

Figures 4 (a)-(d) compare retrieval effectiveness for *Snippet*, *CWS*, and Yahoo for test queries of different sizes, to investigate how retrieval effectiveness varies for the two community-based techniques with changes in query length; remember in Figure 1 we saw how traditional search engines were seen to suffer when faced with longer queries. This remains evident for the Yahoo system, as expected, but we see the retrieval effectiveness for *Snippet* improving with increasing query length. Both *Snippet* and *CWS* significantly out-perform Yahoo across all query categories, and *Snippet* in particular enjoys dramatic improvements in retrieval effectiveness, especially for the longer queries. These longer queries are the very ones that traditional Web search engines appear to struggle with — the vocabulary gap making its presence felt — and yet we find our new snippet-based approach is especially well able to cope with such queries. The terms that have been mined from the selected snippets as part of the community's snippet index are effectively bridging this vocabulary gap.

5.5 Ranking Analysis

Finally, it is worth considering the position of target results within successful sessions and in this experiment we look at the ranking of the target result within the top 10 promotions for *Snippet* and *CWS*, and Yahoo's top 10 results. Figures 5(a) and 5(b) present two different variations on average rank for each of the communities and all 3 search engines. Figure 5(a) shows the mean rank of the target result in those sessions where the target is actually retrieved and indicates that all 3 search engines perform similarly with no single approach winning outright. However, this version of average rank is clearly flawed since the 3 test systems locate the target pages in different search sessions and so this average rank is computed over different test sessions for each system; for example, the *Yahoo* rank is computed over only 8% of the sessions and the *Snippet* rank is computed over 43% of sessions. Hence, we report a *adjusted rank* in which every test session is considered, with those that do not contain the target page among their top-10 results penalized with a rank of 11; this is a conservative penalty since in all likelihood the real rank of the target will be much greater than 11. With this conservative penalty-based ranking function we see in Figure 5(b) that both community-based engines (*Snippet* and *CWS*) significantly outperform *Yahoo*. For example, in the *iPod* community *Snippet* locates the target result at an average adjusted rank of 7.6, compared to 8.41 for *CWS* and 10.36 for *Yahoo*. To put this another way, on average, over the 4 test communities, we find that the adjusted rank of *Yahoo* is 31% greater than *Snippet* and 20% greater than *CWS*.



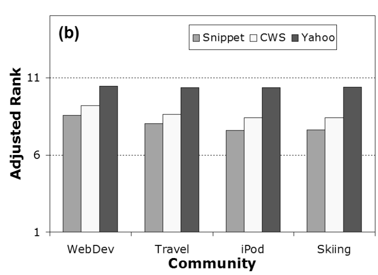


Fig. 5. (a) Average rank for successful sessions per community; (b) Adjusted rank per community

6 Conclusions

The main contribution of this work is a new experience-centric approach to the community-based personalization of Web search results that is based by the selection behaviour of a community of like-minded searchers. This work extends previous work in the use of case-based reasoning techniques for Web search [1] by minining the snippets of selected results to provide a much richer case representation that facilitates more flexible retrieval and result promotion. We have described a comprehensive multi-community evaluation of our unique snippet-based technique compared to a benchmark CWS case-based approach and a leading Web search engine. The results demonstrate the potential benefits of the two case-based approaches by highlighting how both are more successful than the standard Web search engine when it comes to locating target pages across a large set of realistic user data; these targets are located more frequently and positioned earlier in the result lists. In addition, we have shown that our novel snippet-based approach significantly outperforms the CWS benchmark across all 4 communities. Moreover, retrieval effectiveness tends to increase with query length, a desirable outcome that was not found for traditional term-based search engines, and an outcome which suggests our snippet-based cases provide a more effective representation with which to begin bridging the vocabulary gap that hampers Web search.

References

1. Balfe, E., Smyth, B.: Case-based collaborative web search. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 489–503. Springer, Heidelberg (2004)
2. Bollmann-Sdorra, P., Raghavan, V.V.: On the Delusiveness of Adopting a Common Space for Modeling IR Objects: Are Queries Documents? Journal of the American Society for Information Science and Technology 44(10), 579–587 (1993)
3. Budzik, J., Hammond, K.: User Interactions with Everyday Applications as Context for Just-In-Time Information Access. In: Proceedings International Conference on Intelligent User Interfaces, pp. 44–51. ACM Press, New York (2000)
4. Burke, R., Hammond, K., Kulyukin, V., Tomuro, S., Schoenberg, S.: Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System. AI Magazine 18(2), 57–66 (1997)
5. Cui, H., Wen, J.-R., Nie, J.-Y., Ma, W.-Y.: Probabilistic query expansion using query logs. In: Proceedings of the 11th International Conference on World Wide Web, pp. 325–332. ACM Press, New York (2002)
6. Cui, H., Wen, J.-R., Nie, J.-Y., Ma, W.-Y.: Probabilistic Query Expansion Using Query Logs. In: Proceedings of the 11th International Conference on World Wide Web, pp. 325–332 (2002)
7. Dreilinger, D., Howe, A.E.: Experiences with selecting search engines using metasearch. ACM Transactions on Information Systems 15(3), 195–222 (1997)
8. Glover, E., Lawrence, S., Gordon, M.D., Birmingham, W.P., Giles, C.L.: Web Search - Your Way. Communications of the ACM 44(12), 97–102 (2001)
9. Godoy, D., Amandi, A.: PersonalSearcher: An Intelligent Agent for Searching Web Pages. In: Monard, M.C., Sichman, J.S. (eds.) SBIA 2000 and IBERAMIA 2000. LNCS (LNAI), vol. 1952, pp. 62–72. Springer, Heidelberg (2000)
10. Kanawati, R., Jaczynski, M., Trousse, B., J-M, A.: Applying the Broadway Recommendation Computation Approach for Implementing a Query Refinement Service in the CBKB Meta-search Engine. In: Conférence Française sur le Raisonnement á Partir de Cas (RáPC'99) (1999)
11. Kemp, C., Ramamohanarao, K.: Long-term learning for web search engines. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, Springer, Heidelberg (2002)
12. Lawrence, S., Giles, C.L.: Context and Page Analysis for Improved Web Search. IEEE Internet Computing, 38–46 (July-August 1998)
13. Lenz, M., Ashley, K.: AAAI Workshop on Textual Case-Based Reasoning, AAAI Technical Report WS-98-12 (1999)

14. Lieberman, H.: Letizia: An agent that assists web browsing. In: Mellish, C. (ed.) Proceedings of the International Joint Conference on Artificial Intelligence, IJ-CAI'95, pp. 924–929. Morgan Kaufman Publishers, Montreal, Canada (1995)
15. Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of Research and Development 2, 159–165 (1958)
16. Rhodes, B.J., Starner, T.: Remembrance Agent: A Continuously Running Automated Information Retrieval System. In: Proceedings of the 1st International Conference on the Practical Applications of Intelligent Agents and Multi-Agent Technologies, pp. 487–495 (1996)
17. Rijsbergen, C.J.V.: Information Retrieval, 2nd edn. Dept. of Computer Science, University of Glasgow (1979)
18. Rissland, E.L., Daniels, J.J.: A hybrid CBR-IR Approach to Legal Information Retrieval. In: Proceedings of the 5th international conference on Artificial intelligence and law, pp. 52–61. ACM Press, New York (1995)
19. Sakai, T., Sparck-Jones, K.: Generic summaries for indexing in information retrieval. In: SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 190–198. ACM Press, New York, USA (2001)
20. Smyth, B., Balfe, E., Boydell, O., Bradley, K., Briggs, P., Coyle, M., Freyne, J.: A Live-user Evaluation of Collaborative Web Search. In: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05), pp. 1419–1424. Morgan Kaufmann, Ediburgh, Scotland (2005)
21. Smyth, B., Balfe, E., Freyne, J., Briggs, P., Coyle, M., Boydell, O.: Exploiting query repetition and regularity in an adaptive community-based web search engine. User Modeling and User-Adapted Interaction 14(5), 383–423 (2004)