Further Experiments in Case-Based Collaborative Web Search

Jill Freyne[1] and Barry Smyth[1]

School of Computer Science and Informatics,
University College Dublin, Belfield, Dublin 4, Ireland
{Jill.Freyne, Barry.Smyth}@ucd.ie *

Abstract. Collaborative Web Search (CWS) proposes a case-based approach to personalizing search results for the needs of a community of like-minded searchers. The search activities of users are captured as a case base of search cases, each corresponding to community search behaviour (the results selected) for a given query. When responding to a new query, CWS selects a set of similar cases and promotes their selected results within the final result-list. In this paper we describe how this case-based view can be broadened to accommodate suggestions from multiple case bases, reflecting the expertise and preferences of complementary search communities. In this way it is possible to supplement the recommendations of the host community with complementary recommendations from related communities. We describe the results of a new live-user trial that speaks to the performance benefits that are available by using multiple case bases in this way compared to the use of a single case base.

1 Introduction

Improving the quality of Web search results is a challenging problem—the sheer scale and heterogeneity of the Internet is exacerbated by vague and ambiguous queries [4, 8]—but if improvements can be made they will have a significant impact on this very important application area. In our work we have looked at the application of case-based techniques to Web search by looking for query repetition and selection regularity amongst user search patterns. Our key insight has been that, although repetition and regularity is often absent from generic search, it is present in the search patterns of like-minded communities of users that naturally exist [16]. Our collaborative Web search (CWS) approach is designed to operate as a form of meta-search. It relies on some underlying search engine(s) to provide a basic result-list for a user query, but then uses a case base of past search patterns from the user's community to identify key results for promotion.

CWS contemplates a society of community-based search engines, each with their own case base of search cases corresponding to some distinct community of

* The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged.

searchers. Ordinarily the searches of a specific (*host*) community are answered with reference to their local case base: the traditional single case base model of CBR. Recently a number of researchers have investigated the benefits available from combining multiple case bases, each providing access to a different set of problem-solving experiences [9, 11–13]. We adopt a similar strategy in CWS by leveraging the search experience of search communities related to the host (multiple case bases) when responding to queries originating from the host. In doing so we build on work reported in [3], which considered this multiple case base approach in the context of a simple notion of community relatedness, and which provided evaluation results based on an artificial user evaluation. We propose a more sophisticated model of community relatedness and demonstrate the value of the new approach in terms of a new extended live-user trial.

The work presented in this paper touches on a number of areas of related research by combining ideas from Web information retrieval and case-based reasoning. Of particular importance is the idea that Web search experience can be usefully captured as a case base of reusable cases and that this experience can be distributed across multiple case bases which correspond to the different needs of different communities of searchers. There is a long history of the use of case-based methods in a variety of information retrieval tasks. For example, the work of Rissland [14] looks at the application of CBR to legal information retrieval (see also [1]), and Burke et al. [2] describe a case-based approach to question-answering tasks. However these approaches have all tended to focus on particular application domains rather than the broader area of Web search. That said there is some CBR work in the broader context of Web search. For example, the *Broadway* recommender system [7] is notable for its use of case-based techniques to recommend search query refinements, based on refinements that have worked well in the past. Perhaps more related to the core work in this paper is the *PersonalSearcher* [5] which combines user profiling and textual case-based reasoning to dynamically filter Web documents according to a user's learned preferences.

The idea that experience can be distributed across multiple case bases is not new, and in recent years many researchers have considered the use of multiple case bases during problem solving. For example, Leake et al. [9] consider the benefits and challenges when reusing the experience of multiple case bases that reflect different tasks and environments. They consider how a local case base can usefully determine when to look to external case bases as a source of knowledge, and how external cases might be adapted in line with the local task and environment; see McGinty & Smyth [11] for similar work in the route planning domain. Nagendra Prasad & Plaza [13] investigate cooperative problem solving among agents possessing either the same or different capabilities and incorporate potentially different knowledge and problem solving behaviors. Nagendra Prasad et al. [12] present a different situation where no single source of information may contain sufficient information to give a complete solution. They envisages the piecing together of mutually related partial responses from several distributed sources of queries in order to create a complete solution.

2 A Review of Collaborative Web Search

The CWS technique is conceived of as a form of meta-search; see Figure 1. Each new query, q_T, is submitted to a set of underlying search engines and their results are combined to form a meta-search result-list, R_M. The key novelty stems from how a second result-list, R_T, is produced which reflects the learned preferences of a community of like-minded searchers. This involves reusing selection results from past search cases for similar queries, promoting those results that were reliably selected in the past.

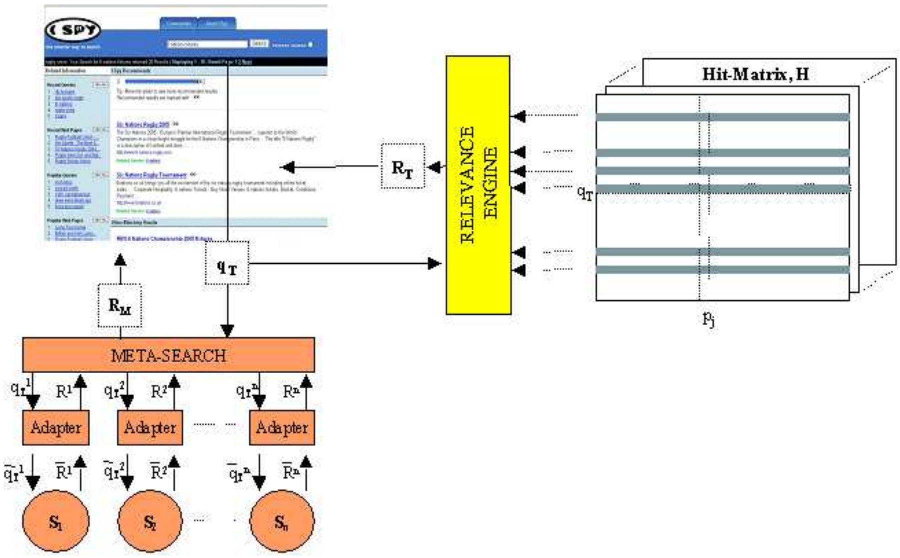

Fig. 1. Collaborative Web search as implemented in I-SPY (*ispy.ucd.ie*).

2.1 The Community Search Case Base

The *hit-matrix* associated with community C, H^C, is a key data structure for CWS, which relates page selections to past queries for a community of users. Specifically, H^C_{ij} (the *hit value* of page p_j for query q_i in community C) is the number of times that page p_j has been selected for query q_i by members of community C; H^C_{ij} is incremented each time p_j is selected for q_i. The hit-matrix forms the basis of a case base. Each row corresponds to a search case (see Equation 1) or, equivalently, a $k + 1$-tuple made up of the query component (a set of query terms) plus k result-pairs, each with a page id and an associated percentage relevance value computed from the hit value for this page and query combination; we will explain how this relevance value is computed below in Equation 6.

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. The *target problem* is, of course, represented by the target query terms.

$$c_i = (q_i, (p_1, r_1), ..., (p_k, r_k)) \tag{1}$$

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

$$Sol(c_i) = ((p_1, r_1), ..., (p_k, r_k)) \tag{3}$$

$$Rel(p_j, c_i) = r_j \texttt{ if } (p_j, r_j) \in Sol(c_i); = 0, otherwise. \tag{4}$$

2.2 Retrieving Similar Search Cases

For each new target query q_T we retrieve a set of similar search cases to serve as a source of relevant results. Case similarity can be measured using a simple term-overlap metric (Equation 5); evaluating alternative metrics is a matter of ongoing research. During the retrieval stage, this allows CWS to rank-order past search cases according to their similarity to the target query so that all, or a subset of, these similar cases might be reused during result ranking.

$$Sim(q_T, c_i) = \frac{|q_T \cap Spec(c_i)|}{|q_T \cup Spec(c_i)|} \tag{5}$$

2.3 Reusing Result Selections

Consider a page, p_j, that is part of the solution of a case, c_i, with query, q_i. The relevance of p_j to this case is given by the relative number of times that p_j has been selected for q_i; see Equation 6. And the relevance of p_j to the current target query q_T is the combination of $Relevance^C(p_j, q_i)$'s for all pages that are part of the solutions to cases $(c_1, ..., c_n)$ deemed to be similar to q_T, as shown in Equation 7. Essentially each $Relevance^C(p_j, q_i)$ is weighted by $Sim(q_T, c_i)$ to discount the relevance of results from less similar queries; $Exists(p_j, c_i) = 1$ if $H_{ij} <> 0$ and 0 otherwise.

$$Relevance^C(p_j, q_i) = \frac{H_{ij}^C}{\sum_{\forall j} H_{ij}^C} \tag{6}$$

$$WRel^C(p_j, q_T, c_1, ..., c_n) = \frac{\sum_{i=1,...,n} Relelevance^C(p_j, c_i) \bullet Sim(q_T, c_i))}{\sum_{i=1,...,n} Exists(p_j, c_i) \bullet Sim(q_T, c_i)} \tag{7}$$

This weighted relevance metric is used to rank-order the promotion candidates. These ranked pages are then recommended ahead of the remaining meta-search results, which are themselves ranked (according to a standard meta-search scoring metric), to give R_T. Of course, alternative promotion models can also be envisaged but are omitted here due to space constraints.

3 Reusing Multiple Case Bases

There are a number of reasons why we might want to look beyond the host community for a complementary source of search experience. A host community might be immature and, as such, may not have accumulated sufficient expertise to respond effectively to a target query. However, other similar, more mature, communities may be available and perhaps they could provide relevant results for the host query. Even a mature host community may not contain sufficient information on a target query; perhaps the query relates to a very specialised information request within the community context. For example, in a community of automobile enthusiasts a specialised query related to the specialised task of restoring a classic s-type Jaguar might be better answered by a related community that is more focused on car restoration.

The main focus of this paper is to explore the various ways that we might exploit the complementary search expertise of related communities by allowing their search cases to contribute to searches by members of the host community using a community cooperation (CC) model. To do this we need to solve two core issues: 1) how to evaluate the relatedness of two communities so that related communities may be identified; 2) how to present the results of a related community to the searchers.

3.1 Evaluating Community Relatedness

When is one community related to another? For the purposes of helping to respond to the search results of our host community, C_h, we can consider two important factors—community similarity and community experience—and we use these measures to evaluate the relatedness of C_h to some other community C_r as shown in Equation 8.

$$\begin{aligned}&Related(C_h, C_r, q_T) = \\ &\quad CommunitySimilarity(C_h, C_r) * CommunityExperience(q_T, C_r)\end{aligned} \tag{8}$$

Community Similarity. It makes sense to look to the recommendations of communities that are demonstrably similar to the host community. But how might community similarity be measured? There are potentially many ways to look at the concept of community similarity. For example we might start by supposing that if two communities have similar query term distributions then they might reflect the interests of two similar communities of users. However this is not necessarily the case, and not sufficient for our needs. For instance, a motoring community might share many queries with a community about wild cats (e.g., 'jaguar', 'puma', 'cougar' are all common car names) but very different result selections will have been made by each community's searchers. Instead we propose to look at the shared results that have been selected in response to searches as an estimate of community similarity (see Equation 9 for the similarity between some host community, C_h, and another community, C_r).

$$CommunitySim(C_h, C_r) = \frac{|Results(C_h) \cap Results(C_r)|}{|Results(C_h)|} \quad (9)$$

Community Experience. As a measure of relatedness, community similarity only tells part of the story. We wish to exploit communities which are similar to the host community and also have a rich store of search information pertaining to a target query. Thus, community experience measures the amount of search history a hit-matrix has for a query. That is, community C is considered to be experienced for a target query, q_T, if its hit-matrix, H^C, contains lots of similar queries and if these similar queries have been successful (users have selected their results frequently) in the past. To measure this, we compare q_T to each of the queries stored in H^C to look for related queries; that is queries with a non-zero similarity to q_T; see Equation 5. For each of these related queries, q_r, we can compute a success score. The *success score* for a query in a hit-matrix is the relative number of hits (selections) that it has contained within its matrix entry, compared to the total number of hits in that hit-matrix; see Equation 10. This metric will deliver high success scores to queries that have resulted in lots of page selections. The degree to which q_T is related to H^C can be computed as the sum of the success scores for each similar query weighted by the degree of similarity; see Equation 11.

$$Success(q_r, H^C) = \frac{\sum_{\forall i} H^C_{ri}}{\sum_{\forall j} H^C_{ij}} \quad (10)$$

$$Related(q_T, H^C) = \sum_{\forall q_r : Sim(q_T, q_r) > 0} Sim(q_T, q_r) * Success(q_r, H^C) \quad (11)$$

$$CommunityExperience(q_T, C) = \frac{Related(q_T, H^C)}{\sum_{\forall C} Related(q_T, H^C)} \quad (12)$$

Community C's experience score reflects the percentage of total query experience contained in its hit-matrix for a target query as shown in Equation 12. This technique allows us to identify a set of communities which all have a rich information history on a target query.

3.2 Result Ranking

Once a set of related communities has been identified (by their similarity to the host community) they can each be used to produce a set of results in response to the target query, q_T, from the host. In this case, for each related community we only seek to retrieve the set of result recommendations coming from their respective hit-matrix (search case base). Thus, each related community, C_i, produces a set of recommended results, R_i. These result-lists complement the result-list R_T that is produced for the host community;

$$CommunityScore(p_j, q_T, C_h, C_1, ..., C_n) = \sum_{i=1,...,n} CommunitySimilarity(C_h, C_i)\dot{W}Rel_i^C(p_j, q_T) \quad (13)$$

When it comes to ranking the results produced by each C_i we use the weighted relevance equation (see Equation 7) for that community. Bearing in mind that an individual results p_j may now be promoted from a number of different communities, $C_1, ..., C_n$ we can also produce a combined ranking score from the weighted sum of the relevance scores for this result across the various communities (see Equation 13). So, results that are promoted by lots of communities that are very similar to the host are considered more relevant to the host that those that are rarely promoted by a few marginally related communities.

3.3 Result Presentation

Equation 13 provides a mechanism for combining all of the promoted results, from the various related communities, into a single promoted result-list for presentation to the user. In this paper, however, we propose keeping the recommendations in their original result-lists and presenting the searcher with a selection of recommendation lists, each from a related community. Each list is labeled with its community name and context and, we argue, that allows the searcher to better understand the nature of the promoted results. In effect this provides for a unique approach to result clustering [6, 10, 17]. Instead of clustering search results by some analysis of their overlapping terms, we are clustering results based on their selection frequencies in different communities of searchers.

An example of this approach is presented in Figure 2 for a collection of search communities related to skiing; these examples use the I-SPY system (ispy.ucd.ie) which is a robust, fully deployed version of CWS. The target query, 'late deals', is provided by a member of the host community, *European Skiing*, and this community's recommendations are shown in the main result page. The section of the result-list shown presents the recommendations from the host community; those results that have been selected and ranked from previous similar cases for this community.

Notice that along the top of the recommended results there is a set of tabs containing the title of a related community. In this example, there are 3 related communities shown, in order of their similarity to the host community. Inset into the figure is the recommendation list from the *American Skiing* community. These recommendations offer late deals in American resorts complement those of the host community. They are however still clearly relevant to the target query.

4 Evaluation

In previous work we have demonstrated the benefits of the standard single case base version of CWS, through a range of live-user trials. For example, in [15,

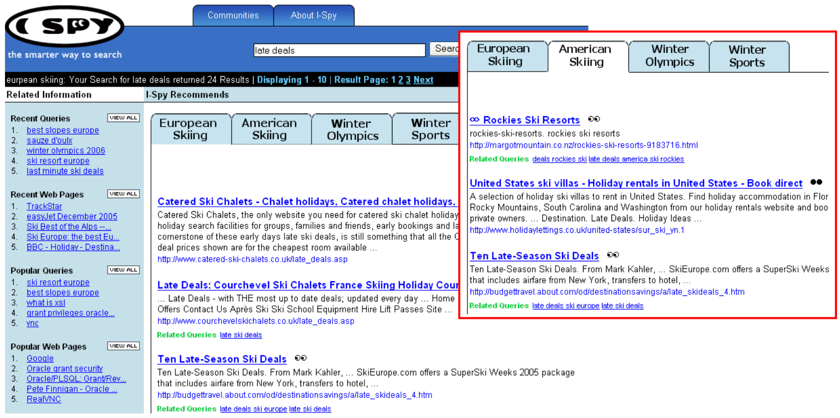

Fig. 2. The recommended results from a selection of skiing communities including the host community (European skiing) and one of the related communities (American skiing).

16] we present the results of two different user trials that show how CWS can significantly improve the precision and recall of an underlying search engine (Google in this case) with respect to the needs of a community of like-minded searchers.

In this paper we have speculated about the value of including recommendations from other search communities when responding to a query submitted in a specific host community. In particular, we have claimed that similar communities will recommend results that are related to the target query and the searcher's needs. Indeed we believe that, in general, communities that are more related to the host will be a more reliable source of relevant results. We also suspect that communities which are less closely aligned to the host may still have a role to play in suggesting results that are partially relevant and that might not otherwise be promoted by the host community. In this section we will describe the results of an experiment designed to test these hypotheses.

4.1 Methodology

The evaluation is carried out in the IT domain with search information collected from a Dublin based software company. The search data was collected over a 9 week period in 2004 and is made up of 1986 search sessions, each containing an internet protocol addresses, a query and at least one result selection, (ip, q, r_1-r_n). This data was used to populate a set of search communities, each made up of the employees of a different division within the company. We tested our hypothesis by querying the resulting case bases with separate sets of real-user queries and judged the recommendations in terms of coverage and precision.

Community Creation In order to test our community cooperation theory, we needed to create a series of separate communities from the collection of data available. The simplest and most effective way of separating the information was to split the data into standard company departments, each department having its own search community. In total, 7 communities were created, varying in department topic and size, from the *Development B* community containing 749 sessions to the *Marketing* community with 52 sessions; see Table 1.

Hit-Matrix Population Populating the hit-matrices for this experiment was a straightforward task. Each community's search data (i.e. query result pairs) was arranged in chronological order. The first 80% of the data - the training data - was used to populate the hit-matrix for that community. The result was a hit-matrix populated as it would have been had a CWS engine been used by searchers at the time the searches were conducted.

Community (sessions)	Web Devel	Marketing	Proj Man.	Devel A	QA	Finance	Devel B
Web Devel(58)		0%	15%	19%	22%	3%	33%
Marketing (54)	0%		9%	25%	13%	11%	20%
Proj Man (204)	3%	2%		17%	20%	4%	29%
Devel A (370)	3%	3%	12%		21%	3%	29%
Quality A (486)	3%	1%	11%	17%		3%	28%
Finance (53)	3%	8%	17%	18%	22%		24%
Devel B (749)	2%	1%	10%	15%	18%	2%	

Table 1. Pairwise community similarities.

Relevance Testing When training was complete we had 7 communities of various sizes, all in some way related to the business of the company. The remaining 20% of search sessions for each community were combined to form a global test set containing 403 queries, each tagged with its host community. The queries contained an average of 2.66 terms and were a mix of general and computing queries; e.g. "*public holidays Ireland*"and "*profiler 3.0 linux installation*".

Our hypothesis is that more experienced, similar communities are better candidates for cooperation than less experienced, less similar communities. Table 1 shows the community similarity figures for all 7 communities. Figure 3 shows the average experience (see Equation 12) of each related community for the test queries of a selection of hosts and the relatedness scores (see Equation 8) for each related community for the same hosts. The performance of each host community for their own test queries (those that they actually contributed to the test set) is compared to the performance of the other six communities, their related communities. Thus for each host community, its test queries are submitted to I-SPY in the traditional manner, generating a host result-list, R_h from the

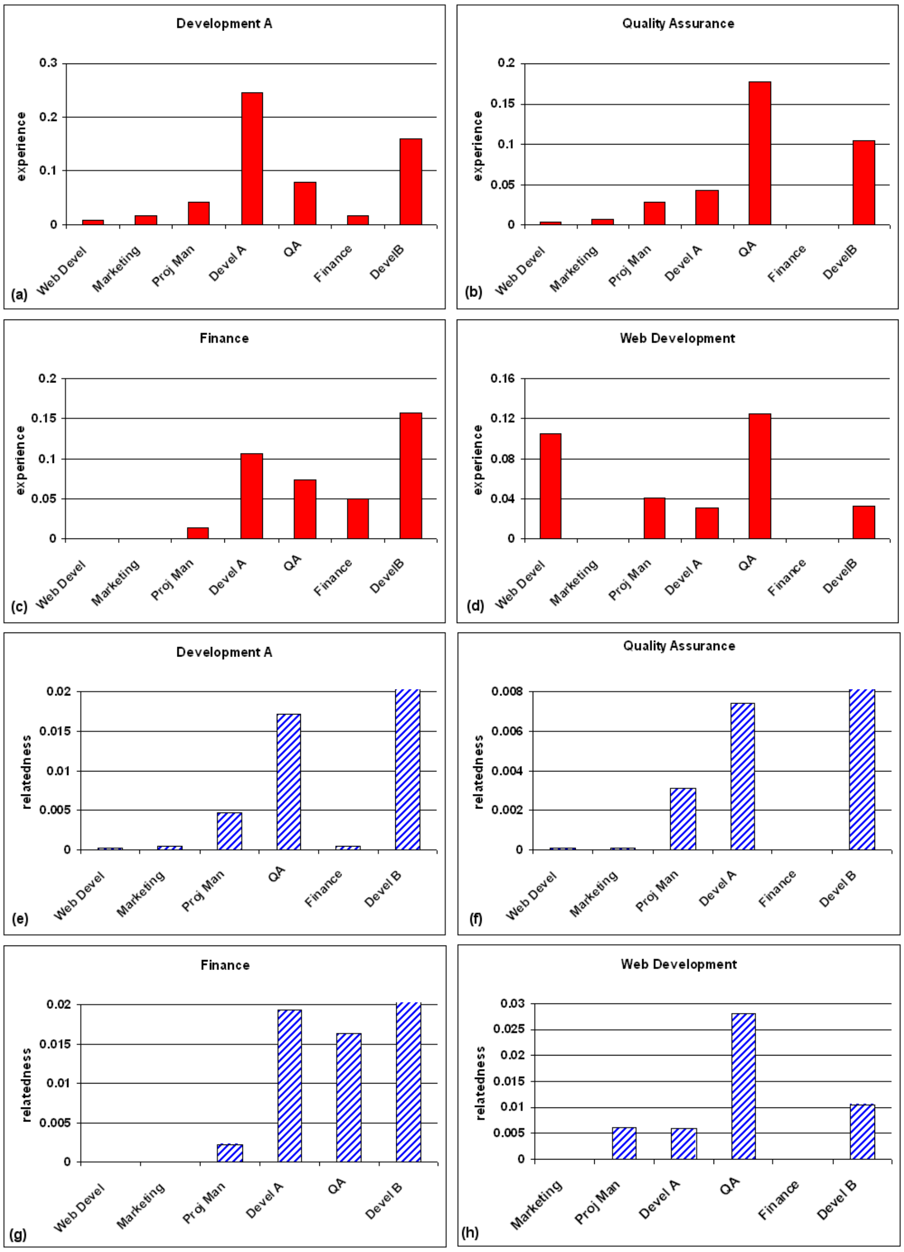

Fig. 3. (a)-(d) Average experience of related communities for host queries from the Finance, Development A, Quality Assurance and Web Development communities. (e)-(h) Relatedness of related communities for host queries from the Finance, Development A, Quality Assurance and Web Development communities.

host's hit-matrix. In parallel, each of the six related communities also receive the test query and produce recommendation lists based on the information in their hit-matrices, R_1,..,R_k. We should point out here that no meta-results were contained in the result-lists just promoted results from the relevance engine.

Although we have access to the results that the original searchers selected, it does not follow that we can assume that unselected results are irrelevant. For our evaluation we needed some way of identifying other results as potentially relevant. Our solution was to use Google's "similar pages" feature as a means to generate lists of results that are similar to those selected by the original searchers (the *seed pages*). This allowed us to generate a list (on average 15.15 results) of *relevant candidates* for each search session from its seed pages. Finally, to determine if some recommended page was relevant for a given query we used Lucene's page similarity function to evaluate the similarity between the page and each of the seed pages and relevant candidates for the test session in question. If the page exceeded a given similarity threshold then it was deemed relevant; a threshold of 0.3 was used for seed pages and 0.5 for the relevant candidates.

4.2 Query Coverage

In our first test we looked at the number of queries for which recommendations could be generated by CWS in comparison to the CC approach. It is important to realise that when we talk about a *recommendation* being generated, we are referring to the promotion of a specific search result based on its previous selection history. Table 2 shows that the CC approach enjoys a clear advantage over the standard CWS approach. Only 82 (20%) of the 403 queries submitted to the standard CWS system resulted in recommendations being generated compared to 130 queries for the CC approach, representing a relative increase in recommendation coverage of more than 58% for the CC approach.

	CWS	CC
Recommendations	82	130
Successful Queries	54	69

Table 2. Technique performance

Recommendations	Web Devel	Marketing	Proj Man.	Devel A	QA	Finance	Devel B
Host	0.50	0.83	0.67	0.75	0.70	0.14	0.47
Related	0.45	0.33	0.67	0.78	0.52	0.86	0.92

Table 3. Percentage of queries to receive recommendations

In Table 3 we break down these figures and examine how each community performed for its own test queries. The graph shows for each community the percentage of their queries that received promotions from their own case base and the percentage that received recommendations from the 6 related communities combined. It shows, for example, that 15% of the *Finance* queries received recommendations from the immature *Finance* community, but 86% of the *Finance* queries received recommendations from a related community. Examination of the *Finance* experience graph in Figure 3(c) shows that a number of related communities have more experience relating to the Finance queries than the *Finance* community itself and thus produce more recommendations. However even the larger, more established *Development A & B* communities see an increase in recommendation numbers, when cooperation is in place. Overall a 13% increase in number of queries to receive recommendations is observed.

4.3 Result Relevance

Of course query coverage is not a revealing measure as it says nothing about recommendation relevance. Thus, we look at the quality of the recommendations generated by each approach. Specifically, we look at the number of queries for which at least one *relevant* recommendation was generated—*successful queries*. The results presented in Table 2 again speak to the benefits of the CC approach, which delivers 69 successful queries against CWS's 54; a relative increase of 27% for the CC approach over standard CWS.

166 relevant results were generated by the traditional CWS technique across its 54 successful queries. When we look at the similar-community recommendations generated for these queries we find 45 relevant results. However crucially, we see that 38 of these 45 relevant recommendations are unique. In other words, over 84% of the relevant recommendations that originate from similar communities are different from the recommendations generated by the host community. It is worth noting that the community with the greatest similarity to the host in most cases, the *Development B* community, did not contribute any unique results to this set, thus showing that communities that are very similar to a host often do not contribute as many unique results as less similar communities.

4.4 Result Precision

403 queries were submitted to each of the 7 communities in turn, noting the performance of the host community in comparison to the other related communities. Figure 4 shows the precision scores for different result-list sizes, $k = 5...100$ for each of the communities and compares the host's result-list precision scores to the result-lists provided by related communities. It is worth noting that only 33% of the test queries received recommendations, which immediately reduces the average precision scores across the test queries. Taking this into consideration, we look at the precision scores in order to compare the traditional CWS technique and the community cooperation model.

As expected, precision values are highest at low values of k and fall as k increases. An immediate trend appears; in four out of seven graphs (Figure 4 (a, c, d & f) the *Development B* community's recommendations outperform the host community's recommendations in terms of accuracy. In these cases a related community has returned more relevant results than the host community,

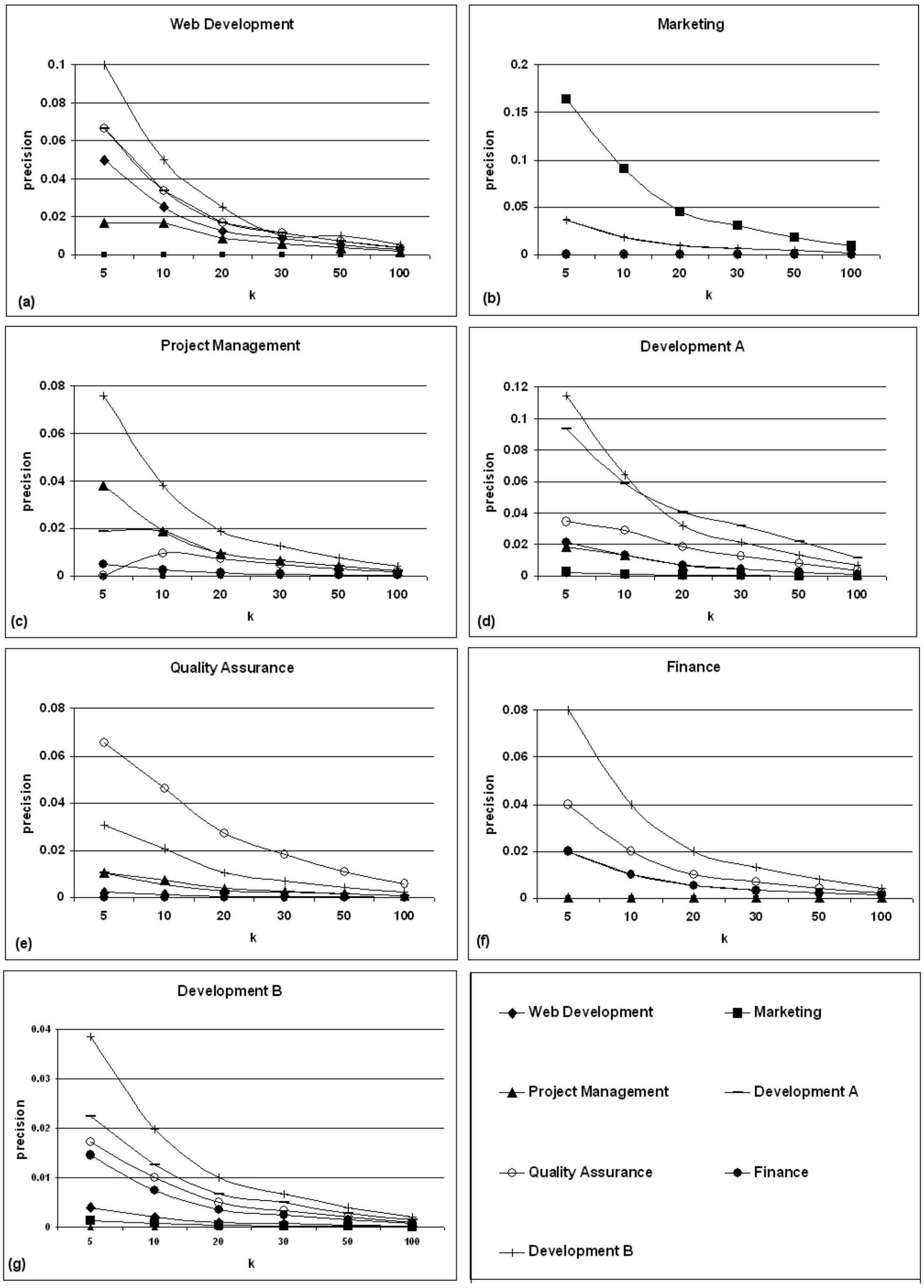

Fig. 4. Precision scores for the (a) Web Development, (b) Marketing (c) Project Management, (d) Development A, (e) Quality Assurance, (f) Finance and Development B communities.

a trend not observed in the previous simulated evaluation [3]. That is, a related community exists, that is better equipped to answer queries than the host. It is worth noting that, even when the *Development B* community does not outperform the host community, it is the best performing related community. The next best performing communities are the *Quality Assurance* and *Development A* communities, which also often equal or outperform host communities.

We proposed that considering a candidate community's experience for a target query and its similarity to a host community informs us of its relatedness to a search scenario. In this evaluation we see the proof of this concept. On average the three most experienced communities for the test queries are the *Development B*, *Quality Assurance* and *Development A* communities; see Figure 3 (a-d). These three communities also have the highest average similarity to the other communities; see Table 1. It follows that these community's suitability to cooperation be reflected in their precision scores. The encouraging finding is that the correlation between precision at k=5 and relatedness is 0.82, supporting our hypothesis that *related* communities, i.e. those that are similar and posses the search knowledge required for the task, are the best candidates for community cooperation where highly similar result-lists are favoured.

4.5 Conclusion

In this work we have shown how Web search experience can be captured as a case base of search cases, and how this experience can be distributed across multiple case bases. Our community collaboration technique aims to supplement a host community's recommendations with complementary recommendations from the search cases contained within the case bases of related communities. By doing this it is hoped that immature search communities will benefit from the experience of more mature communities and that even mature communities will benefit from the complementary viewpoint offered by related communities. We have shown that the experience of communities of searchers can be usefully leveraged to help searchers from other communities. These related communities can serve as a source of recommendations that are both relevant and distinctive.

References

1. Kevin D. Ashley. Modeling Legal Argument: Reasoning with Cases and Hypotheticals. Technical 88–01, Department of Computer and Information Science, University of Massachusetts., Amherst, MA, 1990.
2. Jay Budzik and Kristian J. Hammond. User Interactions with Everyday Applications as Context for Just-In-Time Information Access. In *IUI '00: Proceedings of the 5th International Conference on Intelligent User Interfaces*, pages 44–51, New York, NY, USA, 2000. ACM Press.
3. Jill Freyne and Barry Smyth. Communities, Collaboration and Cooperation in Personalized Web Search. In *Proceedings of the 3rd Workshop on Intelligent Techniques for Web Personalization (ITWP'05) in conjunction with the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, pages 73–80, Edinburgh, Scotland, 2005.

4. George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. The Vocabulary Problem in Human-System Communication. *Communications of the ACM*, 30(11):964–971, 1987.
5. Daniela Godoy and Anala Amandi. PersonalSearcher: An Intelligent Agent for Searching Web Pages. In *IBERAMIA-SBIA '00: Proceedings of the International Joint Conference, 7th Ibero-American Conference on AI*, volume 1952, pages 43–52. Springer-Verlag, 2000.
6. Nigel Hamilton. The mechanics of a Deep Net Metasearch Engine. In *Proceedings of the 12th International World Wide Web Conference (Posters)*, Budapest, Hungary, April 2003.
7. Rushed Kanawati, Michel Jaczynski, Brigitte Trousse, and Jean-Marc Andreoli. Applying the Broadway Recommendation Computation Approach for Implementing a Query Refinement Service in the CBKB Meta-search Engine. In *Coinference Fraicaise sur le Raisonnement a Partir de Cas (RaPC'99)*, 1999.
8. Steve Lawrence and C. Lee Giles. Context and Page Analysis for Improved Web Search. *IEEE Internet Computing*, July-August:38–46, 1998.
9. David B. Leake and Raja Sooriamurthi. When Two Case-Bases Are Better than One: Exploiting Multiple Case Bases. In David W. Aha and Ian Watson, editors, *Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR '01)*, volume 2080 of *Lecture Notes in Computer Science*, pages 321–335. Springer, 2001.
10. Anton Leuski. Evaluating document clustering for interactive information retrieval. In Ling Liu Henrique Paques and David Grossman, editors, *Proceedings of 10th International Conference on Information and Knowledge Management (CIKM'01)*, pages 41–48, Atlanta, Georgia, USA, 2001. ACM Press.
11. Lorraine McGinty and Barry Smyth. Collaborative Case-Based Reasoning: Applications in Personalised Route Planning. In David W. Aha and Ian Watson, editors, *Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR '01)*, volume 2080 of *Lecture Notes in Computer Science*, pages 362–376. Springer, 2001.
12. M.V. Nagendra Prasad, Victor R. Lesser, and Susan E. Lander. Retrieval and Reasoning in Distributed Case Bases. *Journal of Visual Communication and Image Representation, Special Issue on Digital Libraries*, 7(1):74–87, January 1996.
13. M.V. Nagendra Prasad and Enric Plaza. Corporate Memories as Distributed Case Libraries. In *Proceedings of the Corporate Memory and Enterprise Modeling Track in the 10th Knowledge Acquisition Workshop*, 1996.
14. Edwina L. Rissland and Jody J. Daniels. A Hybrid CBR-IR Approach to Legal Information Retrieval. In *ICAIL '95: Proceedings of the 5th International Conference on Artificial Intelligence and Law*, pages 52–61. ACM Press, 1995.
15. Barry Smyth, Evelyn Balfe, Oisín Boydell, Keith Bradley, Peter Briggs, Maurice Coyle, and Jill Freyne. A Live-User Evaluation of Collaborative Web Search. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, pages 1419–1424, Edinburgh, Scotland, 2005.
16. Barry Smyth, Evelyn Balfe, Jill Freyne, Peter Briggs, Maurice Coyle, and Oisín Boydell. Exploiting Query Repetition and Regularity in an Adaptive Community-Based Web Search Engine. *User Modeling and User-Adaptated Interaction*, 14(5):383–423, 2004.
17. Dell Zhang and Yisheng Dong. Semantic, Hierarchical, Online Clustering of Web Search Results. In *Proceedings of the 6th Asia Pacific Web Conference (APWEB)*, pages 69–78, Hangzhou, China, April 2004.