Understanding Query Aspects with applications to Interactive Query Expansion

Daniel Crabtree, Peter Andreae, Xiaoying Gao
School of Mathematics, Statistics and Computer Science
Victoria University of Wellington
New Zealand
daniel@danielcrabtree.com, pondy@mcs.vuw.ac.nz, xgao@mcs.vuw.ac.nz

Abstract

For many hard queries, users spend a lot of time refining their queries to find relevant documents. Many methods help by suggesting refinements, but it is hard for users to choose the best refinement, as the best refinements are often quite obscure. This paper presents Qasp, an approach that overcomes the limitations of other refinement approaches by using query aspects to find different refinements of ambiguous queries. Qasp clusters the refinements so that descriptive refinements occur together with more obscure and potentially better performing refinements, thereby explaining the effect of refinements to the user. Experiments are presented that show Qasp significantly increases the precision of hard queries. The experiments also show that Qasp's clustering method does find meaningful groups of refinements that help users choose good refinements, which would otherwise be overlooked.

1 Introduction

Finding relevant web pages with current search engines can be difficult. If the query terms are not sufficiently distinctive and discriminating (for instance, the query terms co-occur in different contexts), the most relevant web pages can become lost amongst irrelevant ones. When this occurs, users must refine their query to capture their search goal more effectively. However, query refinement is very time consuming as users must examine the result set, identify the irrelevant pages, and determine how to modify their query to exclude the irrelevant results.

Search systems can help users find refinements. The three main techniques for finding refinements are clustering, web log analysis, and relevance feedback. Web page clustering methods [4] find subgroups of related documents. Web log analysis [6] suggests similar queries based on the search sequences of previous users. Relevance feedback methods [13] assume some initial portion of the result set is relevant (pseudo relevance) or take relevance judgments from the user. Refinements are then suggested based on the most frequent terms in the relevant documents.

A new approach to query refinement is to analyze the query. The user's search goal is typically the combination of several aspects. For example, a search goal of finding information about black bears would have one aspect, but searching for instances where black bears have attacked humans involves three different aspects. When forming a query to solve a search goal, users normally select one term (of one or more keywords) for each aspect from their search goal. Query aspects are the aspects of the search goal that are represented in the query.

Although the search engine will only return documents that contain all the terms, many of the documents will contain some of the terms only in an incidental way. For example, the two aspect query "black bear attacks" may find many documents that discuss black bears in detail, but only mention in passing that they sometimes attack. Such a result set focuses on the "black bear" aspect, and underrepresents the "attacks" aspect. The problem of search engines focusing on just a subset of the query aspects is known as the aspect coverage problem [2, 3].

We have previously shown [5] how to address this problem by analyzing the query aspects and automatically modifying the query to increase the coverage of query aspects. An additional problem arises when queries are ambiguous, in which case there are often subtly different relationships between the query aspects, each leading to distinctly different ways of refining the original query.

When queries are ambiguous, users must provide further information about the intent of their search goal. Interactive query refinement may elicit their intent, but it is bothersome to ask the user for unnecessary information and the user is prone to making mistakes [12]. Therefore, the system should only request such information when it is essential. Qasp achieves this by using a novel aspect guided clustering to identify and address query ambiguity.

Following the overview of AbraQ in section 2, section 3 describes Qasp, a novel interactive query refinement method that helps users refine challenging queries. Section 4 evaluates Qasp by comparing its performance with other query refinement methods and finds that it performs significantly better than the alternatives.

2 Query Aspects in Unambiguous Queries

AbraQ [5] successfully used query aspects to perform automatic query expansion on queries with underrepresented aspects. AbraQ identified aspects and evaluated how well they were represented in result sets.

Query aspects can be identified using global document analysis and the semantic information in the order of words in a query: words relating to a single aspect typically occur in sequence, rather than scattered throughout the query. For example, users may search for "tennis courts Los Angeles" or "Los Angeles tennis courts", but probably not for "courts Los tennis Angeles". Building on this principle, global document analysis can be used to identify which query subsequences form aspects.

When a query contains an aspect, the result set for that query is more likely to contain certain vocabulary. For example, "Los Angeles" may invoke terms like "city", "urban", and "California", while "tennis courts" may invoke terms like "game", "clay courts", and "baseline". The degree of representation of an aspect in a result set can be determined by checking the degree to which that aspect's vocabulary is represented in the result set's documents. An aspect is underrepresented when the documents in a result set do not sufficiently represent the vocabulary of the aspect.

AbraQ used this idea to determine when a result set could be improved by modifying the query automatically, but AbraQ cannot help users when the query has no underrepresented aspects, and may perform poorly when the query is ambiguous. Qasp also uses query aspects, but works when the query is ambiguous, regardless of whether a query aspect is underrepresented and unlike traditional clustering methods that address ambiguity, works regardless of the distinctness of the document vocabulary for the different meanings of the query.

3 Qasp - Interactive Query Refinement

When the original result set underrepresents some aspects, then a good refinement is a query that produces a result set with no underrepresented aspects. When the query is ambiguous, two refinements will be more useful when they are very different refinements. Therefore, the system should present a diverse range of refinements of good quality, rather than just selecting the set of individually best refinements.

3.1 Finding Aspects

Like AbraQ [5], Qasp finds aspects by analyzing the word ordering of queries using Global document analysis and finds the vocabulary associated with those aspects by running sub-queries for each aspect and each aspect pair. The result is a set of aspects and a vocabulary model for each aspect, which consists of a set of weighted terms. Also, like AbraQ, Qasp has a refinement scoring function ($RS(r)$) that measures how well a refinement represents all aspects, and has a function that estimates the probability that an aspect is underrepresented in a query.

3.1.1 Selecting Refinements

Unlike AbraQ, Qasp finds a set of good refinements that may be presented to the user. Qasp first selects a set of candidate refinements and scores their performance. Terms that are strongly related to underrepresented aspects are more likely to increase representation of those aspects and therefore, are more likely to lead to good refinements. Qasp selects candidate refinements from the higher weighted terms in each aspect's vocabulary model. The number of terms taken from each aspect is proportional to the relative probability that the aspect is underrepresented.

Qasp uses a greedy approach to select the set of refinements to show the user. At each step, Qasp selects the refinement with the highest combined score ($Score(r)$).

$$Score(r) = \frac{RS(r)}{\operatorname{argmax}_{r' \in R} sim(r, r')}$$

where R is the set of previously selected refinements, and $sim(r, r')$ measures the similarity of the result sets of the refinements.

There are many ways to compute result set similarity. The simplest is the size of the intersection between the result sets. However, the documents may be different and yet still have very similar content. To take into account document content, Qasp computes the cosine similarity [1] ($cos(t_1, t_2)$) between the term vectors t_1 and t_2 for the documents in each result set (weighted by $tfidf$ and with stop words removed).

$$cos(t_1, t_2) = \frac{t_1 \bullet t_2}{|t_1||t_2|}$$

To focus on the differences due to the query ambiguity, Qasp improves on the Cosine similarity measure by computing the similarity by excluding all terms that occur in more than two thirds of the result sets, in addition to the stop words.

To measure the diversity of the set of refinements, Qasp clusters the candidate refinements. When there is only one cluster, the refinements are considered homogeneous and when there are multiple clusters, are considered diverse (suggesting an ambiguous query). Qasp clusters the refinements using an average-link agglomerative clustering algorithm [1] that terminates when the cluster cohesion of a newly merged cluster would be less than the product of the cluster cohesion of its component clusters. Qasp defines cluster cohesion as average similarity between the cluster's candidate refinements and defines the average similarity for singleton clusters as the maximum similarity between the refinement and any other candidate refinement.

3.2 Apply or Display Refinements

When the set of good refinements is homogeneous, there is no ambiguity, and no need to get user input. If all aspects are represented, Qasp does not suggest any refinements. When a query aspect is underrepresented, Qasp will automatically apply the best refinement and can expect to perform close to optimal. Automatically applying the expected best refinement is justified, as research [9] has found that users typically fail to reach optimal performance and typically perform worse than automatic query expansion techniques.

When the set of good refinements is diverse, Qasp assumes the query is ambiguous and that user input is necessary, and therefore presents the refinements to the user. When displayed, the refinements are organized into groups. The groups are based on the refinement clusters, but an additional threshold is applied to stop the clustering earlier, clustering terminates when the cluster cohesion is below 20%. This stopping criterion keeps the clusters from becoming too homogeneous for them to be useful to users. Within each group, the clusters are ordered by expected quality, as measured by the refinement score ($RS(r)$). Additionally, each refinement specifies the most closely associated aspect, which is the aspect from whose vocabulary model the refinement originated.

3.3 Explaining Query Refinements

Users may perform poorly when selecting refinements because they do not understand the effect of the refinements. Often good refinement terms are quite obscure and may require the user to possess substantial domain knowledge before understanding their effect. An advantage of Qasp is that it helps users interpret and choose the best refinements by grouping the refinements in clusters.

The grouping allows users to distinguish the different categories of refinement and identify the best refinements within each category. For example, a user may only recognize associations between their search and some of the refinement terms suggested. However, with the clustering, the user can feel confident in exploring the other refinements within the same cluster(s) as the known refinement terms, as the system has explained that these are very similar refinements. Even if the user did not previously know the best refinement term, they could select it before known terms when the system suggests it is similar to known terms.

Additionally, by associating refinements with aspects, users are guided to appropriate refinements when they observe that the documents are not focused on a particular aspect. Without this guidance, the user may struggle to understand how to address this underrepresentation, even when presented with useful refinements.

4 Evaluation

We evaluated Qasp by comparing its performance with a range of interactive query expansion methods. Algorithm performance was determined by the improvement to precision of 15 hard queries on Google.

4.1 Tests

Many queries are easy and current search engines solve them more than adequately, therefore, it is important to evaluate refinement methods on hard queries. For the test set, we used the topic titles of fifteen queries from the TREC 2005 hard track (topic numbers: 303, 307, 310, 314, 322, 325, 330, 336, 341, 344, 347, 353, 363, 397, and 416). There are seven kinds of queries in the TREC 2005 hard track [15], the test set uses the first two or three topics (by topic number) from each of the seven groups. The queries ranged in length between two and five words, and have between one and three aspects. The queries are mostly difficult ones: five had no relevant results and four had few relevant results in the initial Google search.

Algorithm performance was compared using two measurements: P@5 and P@10, which are the precision of the first 5 and first 10 documents in the search results respectively, where precision [7] is the number of relevant documents retrieved divided by the number of documents retrieved (5 or 10 in this case). The evaluation uses just the first 5 and 10 documents because frequently (over 70%) users only look at the first page of results [8].

4.2 Query Refinement Performance

There are many approaches to query refinement. Qasp was compared against an assortment from each approach. Qasp was compared against three of the best web page clustering algorithms (Suffix Tree Clustering (STC) [14], Lingo [11], and Query Directed Clustering (QDC) [4]), a query log analysis method (Mamma search engine [10]), and two relevance feedback methods [13], one based on document

Table 1. Qasp and interactive query refinement methods compared on 15 search tasks from TREC 2005 hard track against Google baseline

	5 results		10 results	
	Precision	Queries	Precision	Queries
	(Std Dev)	Improved	(Std Dev)	Improved
Method	%	%	%	%
GOOGLE	37 (34)	-	35 (28)	-
Qasp	**76 (19)**	**100**	**75 (20)**	**100**
STC	43 (34)	27	44 (34)	60
LINGO	56 (27)	53	53 (29)	73
QDC	59 (21)	60	54 (19)	80
MAMMA	39 (36)	7	39 (29)	33
RAQE	47 (34)	27	45 (32)	53
PRIQE	49 (37)	40	50 (34)	73
AbraQ	55 (28)	47	47 (26)	53
PRAQE	37 (28)	27	35 (22)	33

relevance feedback (RAQE), and one that lets users select from the top ranking $tfidf$ terms from the initial result set (PRIQE). Besides these interactive query expansion approaches, Qasp was also compared against AbraQ and PRAQE which are automatic query expansion approaches. PRAQE is a pseudo-relevance feedback method that selects the best ranking terms using $tfidf$ from the top N documents.

For each interactive approach, we report the method's optimal performance, when a perfect user either chooses to make no refinement or selects the optimal refinement from the fifteen suggestions made by each method. Note that automatic query expansion approaches can actually decrease performance relative to the search engine for some queries, as happened with PRAQE in 33% of the queries.

Table 1 shows that Qasp outperforms all other tested methods by a significant margin. Significance was tested by comparing algorithms in a pair-wise fashion using the Wilcoxon signed-rank test. Qasp was significantly better than all other algorithms at a 99% level of confidence. Many of the other algorithms improved on Google: AbraQ, PRIQE, Lingo, and QDC were significantly better than Google at a 95% level of confidence, and the remainder were not significantly better than Google.

Qasp was helpful with more queries than any other method. Of the fifteen queries, Qasp improves 80% within the first five results and is the only method to provide a useful refinement for all queries, improving 100% of the queries within the first ten results. AbraQ, which also analyzed query aspects, also does well: although it only improved 53% of queries within the first ten results, it made improvements to all queries that it chose to modify.

Table 2. Selecting different refinements compared with selecting the individually best refinements for 15 search tasks from TREC 2005 hard track against Google baseline

	5 results		10 results	
	Precision	Queries	Precision	Queries
	(Std Dev)	Improved	(Std Dev)	Improved
Method	%	%	%	%
GOOGLE	37 (34)	-	35 (28)	-
Qasp	76 (19)	100	75 (20)	100
Best	69 (23)	80	64 (22)	100

4.3 Selecting Different Refinements

Section 3 claimed that selecting a diverse range of refinements would lead to better results than simply selecting the individually best refinements. Table 2 substantiates this claim. Table 2 compares Qasp's method of selecting different refinements against simply selecting the individually best refinements ("Best"). Qasp's method improves on simply selecting the individually best refinements by a significant margin of 11% in the 10 result case at a 99% level of confidence, and by a smaller and insignificant margin of 7% in the 5 result case. As earlier, significance was tested using the Wilcoxon signed-rank test.

4.4 Clustering Refinements

There were two benefits to clustering refinements: reducing the number of times users need to refine queries, and making it easier for them to pick the appropriate refinement. When there is only one cluster, Qasp automatically applies the best refinement, saving the user effort. In two cases, Qasp found one cluster and automatically applied the best refinement. In both cases, Qasp selected good refinements that dramatically improved performance. For example, for the query "transportation tunnel disasters", Qasp automatically picked the refinement term "injury" and added it to the query: boosting precision to 80% over Google's original 20% in the first 10 results. These two cases are insufficient to generalize whether this automatic application is useful and more experiments would be required to verify the impact of clustering on reducing user effort.

As expected, most queries could be refined in different ways. We informally analyzed the effect of clustering on helping user's choose an appropriate refinement and found clustering was successful. For example, for the query "airport security", one cluster contained refinements such as "protection" and "threats" and were relevant for the search goal of TREC, while another cluster contained refinements such as "safari" which relate to a computer networking device named airport. We observed that clusters grouped related refinements together such that users could identify with one of the refinements and use that as a basis for exploring other related refinements that they would have otherwise felt were irrelevant. This is particularly useful here, as many of the best refinements initially seem irrelevant to users, as the best refinements may not be descriptive and may simply co-occur with other more descriptive terms. A user study would be required to fully determine the benefits for users, but our initial analysis suggests that our clustering approach is promising.

5 Conclusion

This paper presented Qasp, a new interactive query expansion method that helps users refine queries, with significant improvements to precision. Qasp analyzes query aspects and uses them to identify the different ways of refining the query. By clustering the refinements, Qasp determines if the query is ambiguous and therefore requires user input to select an appropriate refinement. The clustering of refinements also provides an explanation to users describing the effect of different refinements, which helps users select the best refinement.

The evaluation of Qasp found that it significantly improved the precision of many hard queries as compared to a representative range of query refinement methods. Furthermore, the evaluation found that Qasp improves on AbraQ which also uses query aspects, by working with ambiguous queries, and improves on clustering approaches, by working regardless of the document vocabulary.

While Qasp performed very well, there is plenty of scope for future work. Further investigation is needed into the impact of the clustering refinements on helping users choose appropriate refinements, and further research is needed on how to better explain refinements to users.

Acknowledgements

Daniel Crabtree is supported by a Top Achiever Doctoral Scholarship from the Tertiary Education Commission of New Zealand.

References

[1] P. Berkhin. Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, CA, 2002.

[2] C. Buckley. Why current ir engines fail. In *ACM SIGIR*, pages 584–585, New York, NY, USA, 2004. ACM Press.

[3] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg. What makes a query difficult? In *ACM SIGIR*, pages 390–397, New York, NY, USA, 2006. ACM Press.

[4] D. Crabtree, P. Andreae, and X. Gao. Query directed web page clustering. In *Web Intelligence*, pages 202–210, 2006.

[5] D. Crabtree, P. Andreae, and X. Gao. Exploiting underrepresented query aspects for automatic query expansion. In *The Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2007.

[6] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *WWW*, pages 325–332, 2002.

[7] C. de Loupy and P. Bellot. Evaluation of document retrieval systems and query difficulty. In *Using Evaluation within HLT Programs : Results and Trends*, pages 34–40, 2000.

[8] B. J. Jansen, A. Spink, and J. O. Pedersen. A temporal comparison of altavista web searching. *JASIST*, 56(6):559–570, 2005.

[9] M. Magennis and C. J. van Rijsbergen. The potential and actual effectiveness of interactive query expansion. In *ACM SIGIR*, pages 324–332, 1997.

[10] Mamma.com: www.mamma.com, 2007.

[11] S. Osiński, J. Stefanowski, and D. Weiss. Lingo: Search results clustering algorithm based on singular value decomposition. In *Intelligent Information Processing and Web Mining Conference*, Advances in Soft Computing, pages 359–368, Zakopane, Poland, 2004. Springer.

[12] M. Pechenizkiy, A. Tsymbal, and S. Puuronen. Pca-based feature transformation for classification: issues in medical diagnostics. In *17th IEEE Symposium on Computer-Based Medical Systems*, pages 535–540, 2004.

[13] I. Ruthven and M. Lalmas. A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, 19(2):95–145, June 2003.

[14] O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. In *Research and Development in Information Retrieval*, pages 46–54, 1998.

[15] J. Zhang, L. Sun, Y. Lv, and W. Zhang. Relevance feedback by exploring the different feedback source and collection structure. In *Text REtrieval Conference (TREC)*, 2005.