Inferring Query Aspects from Reformulations Using Clustering

Van Dang, Xiaobing Xue and W. Bruce Croft
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{vdang, xuexb, croft}@cs.umass.edu

ABSTRACT

When the information need is not clear from the user query, a good strategy would be to return documents that cover as many aspects of the query as possible. To do this, the possible aspects of the query need to be automatically identified. In this paper, we propose to do this by clustering reformulated queries generated from publicly availables resources and using each cluster to represent an aspect of the query. Our results show that the automatically generated reformulations for the TREC Web Track queries match up quite well with actual sub-topics of these queries identified by TREC experts. Moreover, agglomerative clustering using query-to-query similarity based on co-occurrence in text passages can provide clusters of high quality that potentially can be used to identify aspects.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query Formulation

General Terms

Algorithms, Measurement, Performance, Experimentation.

Keywords

Query diversity, query reformulation, clustering, anchor text.

1. INTRODUCTION

User queries do not always clearly represent the actual information need. They can be ambiguous or underspecified. Ambiguous queries are those that have different interpretations such as "TREC", which might refer to the home page of Texas Real Estate Commission or Text Retrieval Conference. Underspecified queries are queries with a known interpretation that have different aspects or sub-topics that may be relevant. For example, the user submitting the query "apple inc." might be looking for general information about the company or products of the company.

In order to deal with these queries, retrieval systems should retrieve documents that are relevant to different aspects of the query rather than a single, dominating aspect. Conventional information retrieval (IR) systems can do little for these types of queries since they rank documents without regard to possible meanings the query might have.

Dealing with these types of queries is the motivation for introducing diversity into search results, to make the set of returned documents cover as many aspects of the queries as possible. Existing work in this area generally relies on an intial retrieval and then selecting documents from the retrieved set according to some criteria. These methods can be categorized as *implicit* and *explicit*. The implicit approach [2, 13] chooses documents different to those that have been previously selected without modeling the actual sub-topics of the query. The explicit approach, on the other hand, explicitly models aspects of a query using a taxonomy [1], relevant documents [3] or reformulations [10, 12]. Many of the explicit models, however, assume the availablility of the optimal aspect representation for a query, leaving their effectiveness with automatically generated aspects unclear.

In this paper, we propose a simple method to generate reformulations that represent possible aspects of a query from anchor text and the Microsoft Web N-Gram Services [1], both of which are publicly available. We first generate reformulations by using these resources, and then cluster them using different clustering algorithms with different similarity measures. Our experiments show that many of the reformulations we generate for TREC queries in fact correspond well with their sub-topics as identified by TREC experts.

2. GENERATING REFORMULATIONS

Even though many techniques have been proposed for query reformulation, most of them aim to generate queries that are more effective than the original query [7, 5]. Their effectiveness for providing reformulations that cover different intents is thus unclear. Therefore, instead of using these models, we use a rather simple technique to generate reformulations from publicly available resources including anchor text extracted from a web collection and the Microsoft N-gram Services.

[1] http://research.microsoft.com/en-us/collaboration/focus/cs/web-ngram.aspx

2.1 Anchor Text

Anchor text is known to be an effective feature for web search [9]. Previous researchers have observed the similarity between anchor text and queries [6, 5]. Therefore, in this paper, we treat each anchor text as a reformulation that can potentially represent one aspect of a query.

The web collection from which we extract the anchor text is the English portion of the ClueWeb-09 category A [2]. It contains 500 million pages in English that were crawled from the web during early 2009. We extracted all pairs of anchor text and associated urls from the web pages in this collection.

Web pages are connected to one another via links, each of which is associated with some anchor text. A link is called *internal* if two connected pages are from the same domain and *external* if they come from different domains. Since most of the internal links are for navigation purposes, their associated anchor text is not very helpful. Typical examples of such anchor text are "home" and "index". As a result, we only consider *external* links.

In order to reduce noise, we discarded anchors that contain non-English words and those that contain navigation-triggered words such as "click", "download" and "subscribe". We also removed anchors that contain only numbers and stop words. Among the resulting anchors, we keep only those with frequency greater than 1 and are connected to at least *two* urls. The resulting anchor text collection contains $8,215,751$ unique anchors.

For any given query, we use the top-M most frequent anchor texts that contain all of its terms as its reformulations.

2.2 Microsoft Web N-gram Services

The Microsoft Web N-gram Services provide smoothed n-gram models built from document body, document title, anchor text and queries in the Bing query log separately. Each model gives the probability of seeing an unigram u coming after an n-gram n, or $P(u|n)$.

For each query q, we obtain the top-M unigrams u with largest $P(u|q)$. Each reformulation is formed by adding u to q.

We put all reformulations generated from the two sources above into a list L. Since we aim to use each reformulation as a query aspect, we keep only those with reasonably high frequency. Ideally, we can obtain this frequency from query logs. Because we rely only on publicly available resources, we approximate this frequency by the number of times the query appears in a web collection. Finally, we order the reformulations in L by their frequency and keep only the top-M, which will be the candidates for clustering.

3. CLUSTERING

The list of reformulated queries generated above are then clustered into groups, each of which is considered a coarse representation of a query aspect. The clustering is based on a measure of query similarity.

3.1 Similarity Measures

3.1.1 Relevance Models

Since the queries are short, computing their similarity based only on the query words is not likely to be effective. Instead, we expand a query with documents that are likely to be relevant to it. Specifically, we represent each query q by the relevance model $P_q(w|R)$ [8] estimated from the top-10 documents returned by the query likelihood retrieval model for q.

The similarity of two reformulations r_1 and r_2 is then the KL-divergence between their relevance models $P_{r_1}(w|R)$ and $P_{r_2}(w|R)$. We also try the cosine similarity measure as an alternative to KL-divergence.

3.1.2 Co-occurrence At Passage Level

Since estimating relevance models for every reformulation is computationally expensive, we also examine a more efficient method based on passage analysis. The idea is that two queries are more similar if they co-occur often in the same text passages. Therefore, for every pair of reformulations r_i and r_j, we compute N_i and N_j – the number of passages in which each of them occurs, and N – the number of passages in which they co-occur. The similarity between r_i and r_j is given by the Jaccard score:

$$sim(r_i, r_j) = \frac{N}{N_i + N_j - N}$$

3.2 Clustering Algorithms

We applied two standard clustering algorithms: K-Means and Agglomerative Clustering.

3.2.1 K-Means Clustering

The algorithm initializes each of the K clusters with a random reformulation. It then iteratively partitions all reformulations into K clusters in which each reformulation is assigned to the cluster that is most similar to it. The similarity between a query and a cluster is the average similarity between this query and all the other queries in the cluster. The algorithm terminates when the cluster assignment for reformulations does not change.

3.2.2 Agglomerative Clustering

Agglomerative clustering has an advantage over K-Means in that we do not have to specify the number of clusters beforehand. The standard algorithm treats each reformulation as a singleton cluster. It sucessively merges pairs of clusters that are most similar to each other until some creteria is achieved. In our experiments, the algorithm stops when the intra-cluster similarity drops below a certain threshold τ. We use complete-link to compute the similarity between two clusters, which is the minimum pair-wise similarity between the two clusters.

This procedure generally produces a deep binary tree, which is unnessary in our case since we are not interested in the tree structure. Therefore, we collapse all leave nodes, which correspond to our generated reformulations, starting from the third level (excluding the root node) to their parent and use these parent nodes as our resulting clusters.

4. EXPERIMENTS

In our experiments, we use queries from TREC Web Track 2009 and 2010. This query set contain 100 queries. Each query comes with associated sub-topics identified by TREC experts. On average, there are 4.6 subtopics per query.

For each query, we generate reformulations as described in Section 2. We evaluate the quality of these reformulations

[2]http://boston.lti.cs.cmu.edu/Data/clueweb09/

by judging how many of them correspond to the actual sub-topics identified by TREC experts. Then, we evaluate the clusters provided by different combinations of clustering algorithms and similarity measures.

4.1 Data Preparation and Parameter Settings

We used ClueWeb-09 category B as the web collection both for estimating the frequency of reformulations and estimating the co-occurrence statistics. For frequency estimation, we found it too strict to require the exact query to appear in the document. Therefore, we relaxed this by counting the number of times all of the query's terms co-occur within a windows of size 10 and used this as its frequency. For passage analysis, two reformulations are considered co-occurring in the same text passage if all of their terms co-occur within a window of size 20.

We set the number of reformulations $M = 100$ in all of our experiments. As for K-Means, we empirically set $K = 10$.

4.2 Quality of Reformulations

As mentioned in Section 2, we put all reformulations generated from different sources together for each query and keep only top-100 most frequent ones. Among these reformulations, 15% is exclusively from the anchor text, 76% is exclusively from the Web N-gram service and 9% is from both sources.

In this experiment, two graduate students independently judge each of those 100 reformulations to see if it corresponds to any actual sub-topics of the query. A reformulation is then labeled by the corresponding sub-topic, or "none" if it does not match with any of the sub-topics. The inter-agreement between our two judges is 94%.

An actual sub-topic of the query is considered covered if at least one of the reformulations corresponds to it. Fig. 1 shows the percentage of subtopics (averaged across all queries) covered by the top-N of the 100 reformulations with N varying from 10 to 100. In general, the reformulations covers on average about 60% of the actual sub-topics, which is promising considering these reformulations are acquired in a very simple way. This suggests that publicly available resources are very useful at identifying aspects of queries.

It is worth noting that the reformulations that do not correspond to any of the aspects are not necessarily bad. In fact, many of them represent valid intents that were not identified by TREC experts. We leave the evaluation of these reformulations for future work.

4.3 Quality of Clustering

In this section, we tried different combinations of clustering algorithms and similarity measures to cluster all reformulations we have generated. We expect the techniques to be able to put reformulations with the same label into the same cluster.

To evaluate the quality of the generated clusters, we use the Rand index (RI), a well-known cluster quality measure. It computes the percentage of decisions that are correct and is calculated as follows:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

where TP (true positive) is the number of pairs of reformulations with the same labels that are put into the same cluster, TN (true negative) is the number of pairs with different labels that are put into the same cluster, FN (false negative) is the number of pairs with the same labels that are put into different clusters, and FP (false positive) is the number of pairs with different labels that are put into the same clusters. Reformulations with the label "none" are ignored in this computation since "none" is not a topic. Table 1 shows the RI score that different combinations achieve.

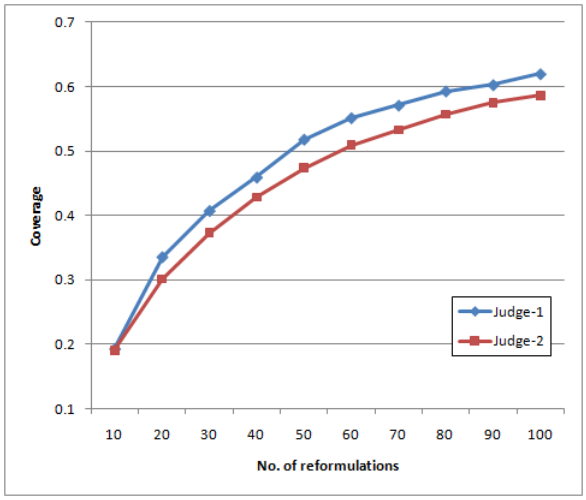

Figure 1: Quality of the generated reformulations in terms of how many of the actual sub-topics of the queries they cover.

Table 1: Quality of the automatically generated reformulations.

		RM+Cos.	RM+KL	PS+JAC
Judge-1	Agglo.	0.64	0.67	**0.76**
	K-Means	0.5	0.55	0.59
Judge-2	Agglo.	0.63	0.7	**0.73**
	K-Means	0.48	0.55	0.57

The first thing we observe from Table 1 is that agglomerative clustering consistently outperforms K-Means. The reason seems to be due to the fact that K-Means forces every reformulations to be in some cluster. This can result in unrelated reformulations being put into the same cluster. Once clusters are filled with unrelated reformulations, the centroids of those clusters are not very different from each other, making the cluster assignment in the next iteration unreliable. Agglomerative clustering only merges two clusters if they are very similar to each other, and has a lower chance of putting reformulations into unrelated clusters.

Secondly, Table 1 shows that the similarity measure based on co-occurrence is consistently better than those based on relevance models. It should be noted that most of the reformulations, especially those generated from the Microsoft N-Gram Services, are different to each other by only one word. The longer the original query, the less impact the augmented word has on the relevance model. As a result, the relevance models for these reformulations are more similar than they should be. The similarity measure based on co-occurence, on the other hand, is not affected as much by the length of the original query. Two reformulations are similar as long as their augmented words co-occur with each other and with

Table 2: Example of clusters generated by agglomerative clustering for the query "satellite"

{satellite tv; satellite tv vs cable; satellite network}
{satellite radio; sirius satellite radio; xm satellite radio}
{satellite image; google maps satellite}
{satellite internet}
{weather satellite; satellite climate}
{satellite technology; satellite development}
{satellite broadband}

the original query. This gives the co-occurrence-based measure superiority over the other two.

Tables 2 presents an example of clusters generated by agglomerative clustering with the co-occurrence similarity measure for the query "satellite".

5. RELATED WORK

Existing work in query diversity focuses mainly on selecting a diverse subset of documents returned by an initial retrieval. Maximal Marginal Relevance (MMR) [2] is a well-known example of this approach. It sequentially selects documents that are most different to those previously selected. Probabilistic versions of MMR have also been investigated [13, 4]. While these approaches only aim to select documents that cover different topics, others model the query's aspects explicitly [3, 1, 12].

Our approach is different to this existing research in that it works on the query side. It aims to generate clusters of reformulations, each of which represents one aspect of the query. In this preliminary study, even though we have not explicity diversified the set of reformulations, the clustering has this effect. Since clustering aims to put similar reformulations together, reformulations representing different aspects should be put in different clusters.

Query-side diversification has been investigated by Radlinski and Dumais [10]. However they use a proprietary query log whereas we examine the usefulness of publicly available resources. The most similar work to ours is that done by Radlinski et al. [11] in which they also infer queries' intents. However, they also rely on proprietary query logs.

Our approach does have a relationship to document-side diversification. The aspects our method generates for queries can be used by models such as [1, 12]. In addition, these models search for candidates from a set of documents retrieved for the original query. Our aspects are reformulations of the original query, and can be combined with the query to retrieve more potential documents for document-side models to work with.

6. CONCLUSIONS

In this paper, we have shown that reformulations for queries obtained from publicly available resources such as anchor text and Microsoft Web N-Gram Services match up well with the actual sub-topics of the queries. We then tested whether clusters of reformulations represent aspects, using different clustering algorithms and query similarity measures. We found that agglomerative clustering consistently outperforms K-Means and the similarity measure based on co-occurence is not only more effecient but also works better than similarity based on relevance models.

We observe that many of the clusters we obtained have very high intra-cluster consistency. We plan to build aspect representation from each cluster and use these in retrieval experiments.

Since clustering puts similar reformulations together, it has the effect of diversification because diverse reformulations should go to different clusters. Therefore, it would be interesting to see how clustering performs compared to applying existing techniques such as MMR on the set of reformulations.

As noted earlier, our method identified many interesting aspects that were not present in the TREC judgment. In the future, we will evaluate these reformulations, and identify how many of them represent valid aspects.

7. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by ARRA NSF IIS-9014442. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] R. Agrawal, S. Gollapudi, A. Halverson and S. Leong. Diversifying Search Results. In *Proceedings of WSDM*, 2009.

[2] J. Carbonell and J. Goldstein. The Use of MMR, Diversity-based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of SIGIR*, 1998.

[3] B. Carterette and P. Chandar. Probabilistic Models of Novel Document Rankings for Faceted Topic Retrieval In *Proceedings of CIKM*, 2009.

[4] H. Chen and D.R. Karger. Less is More: Probabilistic Models for Retrieving Fewer Relevant Documents In *Proceedings of SIGIR*, 2006.

[5] V. Dang and W.B. Croft. Query Reformulation Using Anchor Text. In *Proceedings of WSDM*, 2010.

[6] N. Eiron and K.S. McCurley. Analysis of Anchor Text for Web Search. In *Proceedings of SIGIR*, 2003.

[7] R. Jones, B. Rey and O. Madani. Generating Query Substitutions. In *Proceedings of WWW*, 2006.

[8] V. Lavrenko and W.B. Croft. Relevance-based Language Models. In *Proceedings of SIGIR*, 2001.

[9] D. Metzler, J. Novak, H. Cui, and S. Reddy. Building Enriched Document Representations Using Aggregated Anchor Text. In *Proceedings of SIGIR*, 2009.

[10] F. Radlinski and S. Dumais. Improving Personalized Web Search Using Result Diversification. In *Proceedings of SIGIR*, 2006.

[11] F. Radlinski, M. Szummer, and N. Craswell. Inferring Query Intent from Reformulations and Clicks. In *Proceedings of WWW*, 2010.

[12] R. Santos, C. Macdonald and I. Ounis. Exploiting Query Reformulation for Web Search Result Diversification In *Proceedings of WWW*, 2010.

[13] C. Zhai, W. Cohen and J. Lafferty. Beyond Independent Relevance: Methods and Evaluation Metrics for Subtopic Retrieval. In *Proceedings of SIGIR*, 2003.